# *NATIONAL AIRSPACE SYSTEM*

# *System Engineering Manual*

---

*Federal Aviation Administration*
*ASD-100*
*Architecture and System Engineering*

Approved by: _____ Date _____
Mike Harrison, Program Director,
ASD-100

FAA's Office of System Architecture and Investment Analysis

# SYSTEM ENGINEERING MANUAL
# TABLE OF CONTENTS

1. INTRODUCTION....1-1
2. OVERVIEW OF SYSTEM ENGINEERING....2-1
3. SYSTEM ENGINEERING IN THE ACQUISITION MANAGEMENT SYSTEM PROGRAM LIFE CYCLE....3-1
4. PERFORM SYSTEM ENGINEERING....4-1
   - 4.1 System Engineering....4.1-1
   - 4.2 Integrated Technical Planning....4.2-1
   - 4.3 Requirements Management....4.3-1
   - 4.4 Functional Analysis....4.4-1
   - 4.5 Synthesis....4.5-1
   - 4.6 Trade Studies....4.6-1
   - 4.7 Interface Management....4.7-1
   - 4.8 Specialty Engineering....4.8-1
   - 4.9 Integrity of Analyses....4.9-1
   - 4.10 Risk Management....4.10-1
   - 4.11 Configuration Management....TBD
   - 4.12 Validation and Verification....4.12-1
   - 4.13 Life Cycle Engineering....TBD
   - 4.14 Maintain System Engineering....4.14-1

APPENDIX A: ACRONYMS....A-1
APPENDIX B: GLOSSARY....B-1
APPENDIX C: INITIAL SYSTEM REQUIREMENTS REVIEW CHECKLIST....C-1
APPENDIX D: CONCERNS/ISSUES....D-1

# 1 INTRODUCTION

Since the end of World War II, development of larger, more complex systems has increased awareness of, and need for, System Engineering (SE). The principles of SE were especially significant in the development of ballistic missile weapon systems because of the need to apply new technology to both the system's existing and new parts while controlling the inherent risks. Difficulties experienced in the evolution of the integrated designs of new systems (especially in the software community) have led to the development of specific methods and techniques within the SE discipline in an attempt to provide better insight to and control of the development and management process. Commercial applications of SE have also expanded and are being used by utility companies, aviation, the automobile industry, the computer industry, communications, and health care.

There are many definitions of SE in books, professional journals, and classrooms. The definition arrived at by Simon Ramo, one of the early developers of formal SE and the "R" in TRW, is used by the System Engineering Manual (SEM):

> *A discipline that concentrates on the design and application of the whole (system) as distinct from the parts. It involves looking at a problem in its entirety, taking into account all the facets and all the variables and relating the social to the technical aspect.*

Because of lessons learned within the government and industry, the Federal Aviation Administration (FAA) is pursuing SE concepts. This manual reflects the application of SE within the FAA. The FAA's SE applies both technical and management principles in a manner that:

- Results in a responsive system design from an operational need through the use of a structured, iterative process
- Integrates the contributions of both traditional engineering and specialty disciplines to meet cost, technical, quality, and schedule objectives
- Provides a product that satisfies all stakeholders

The three primary purposes of this manual are to:

- Define the FAA's integrated practice of SE, which is compatible with all components of the agency and consistent with sound government and industry best practice, to be employed by any engineer or group performing a task requiring an SE approach
- Provide methods and tools that result in effective and consistent SE
- Supply detailed information on work products of SE activities that are needed to ensure uniform and consistent high-quality products

## 1.1 Scope

The SEM describes the proper application of SE elements within the FAA. These SE elements are specifically designed for the acquisition of systems for the National Airspace System (NAS) in the context of the FAA Acquisition Management System (AMS). However, SE is one of the key practices of the FAA, and the elements discussed here can be applied in a number of other FAA functions beyond acquisition. "Perform System Engineering" is specifically part of the AMS and is tightly linked with other AMS processes.

SE provides a comprehensive, structured, and disciplined approach for new system product and process developments, upgrades, modifications, and engineering efforts conducted to resolve problems in fielded systems in all development, production, and operations/support phases. SE is applicable to technical efforts that support advanced development of new technologies and their application. It applies to large- and small-scale systems (ranging in size and complexity from the NAS to individual parts such as bolts or wire bundles), single or multiple procurements, and replacement of current products and processes. The process is applicable to systems regardless of composition, including those that are integrated from diverse elements, either hardware-intensive or software-intensive. SE involves design and management of a total system, including hardware and software as well as other system elements. Each shall be considered in analyses, trade studies, and engineering methodology.

The SEM is designed as a how-to manual or guide to SE and defines the constituent SE elements to be performed throughout the program lifecycle. When the term “program” is used, it is intended to mean projects at all levels of size and complexity, ranging from the NAS to individual parts. While the SEM is primarily directed at NAS modernization, individual programs shall tailor the application of processes, tools, and techniques according to program requirements, with implementation of these processes directed by the appropriate SE management authority designated in the NAS System Engineering Management Plan (SEMP) or, on a given program, by the Program Manager or Chief System Engineer. This manual includes guidance on tailoring.

Figure 1.1-1 is a high-level overview of the organizational levels involved in SE along with several key SE work products. This figure provides only a contextual overview of SE within the FAA. Specific details on who is doing what regarding SE are explained in an organization or program SEMP.

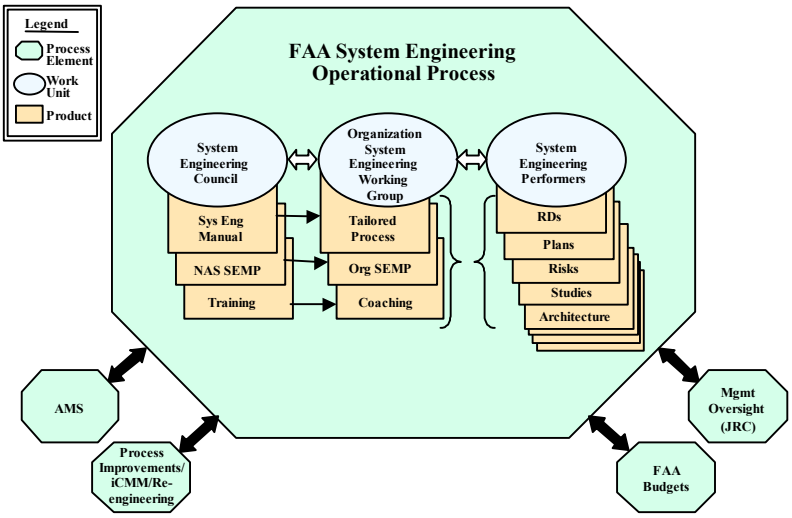



**Figure 1.1-1. Federal Aviation Administration System Engineering Operational Process**

## 1.2 Purpose

The SEM and SEMP are designed to work together. The SEM answers SE questions related to what and how, while the SEMP answers SE questions related to what, who, when, and why

(i.e., why a particular organization or program is implementing or not implementing a particular SE element versus the SEM's discussion on a SE element's purpose). They are directly connected by the "what" or products and activities of SE. This relationship between the SEM and SEMP appears in Figure 1.2-1.

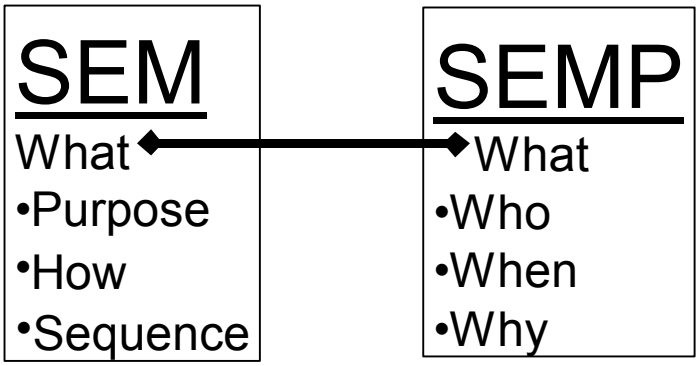



**Figure 1.2-1. Relationship Between the System Engineering Manual and the System Engineering Management Plan**

The SEM describes the key process activities and products that are necessary to effectively implement SE principles. The manual offers both what-to-do and how-to-do-it information for the elements defined for the FAA's SE model. These SE elements appear in Figure 1.2-2.

The SEM describes the purpose of specific SE activities, as well as guidance for determining the sequence in which each activity is to be performed relative to the AMS phases. The SE activities are applicable to most engineering programs. Program cost/benefit considerations shall be the basis for the allocation of appropriate resources, including manpower and schedule, to any process activity.

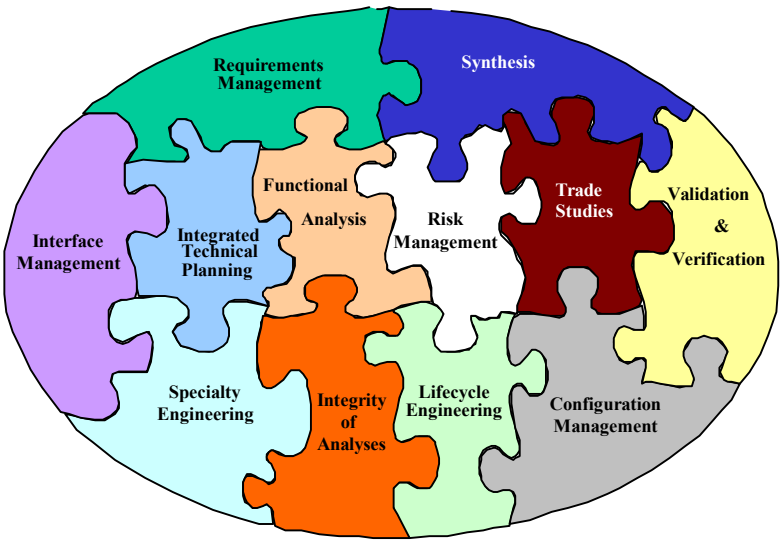



**Figure 1.2-2. Federal Aviation Administration System Engineering Elements**

The process descriptions consist of all aspects of each SE process, including the need to design for safety, as well as for affordability, performance, usability, operational suitability, and cost of ownership. On some programs, a given activity may be performed informally (e.g., in an engineer's notebook) or formally, with interim products under formal baseline control.

The SEM defines the SE elements, as conducted in the FAA, along with the work products generated from each SE elements. The SE elements are listed in Table 1.2-1:

**Table 1.2-1. System Engineering Elements**

| System Engineering Element | Purpose of Element |
|---|---|
| Integrated Technical Planning | Plans the SE efforts and products. |
| Requirements Management | Identifies and manages the requirements that describe the desired characteristics of the system. |
| Functional Analysis | Describes the functional characteristics (what the system needs to do) that are used to derive requirements. |
| Synthesis | Transforms requirements into physical solutions. |
| Trade Studies | Assists decisionmaking by analyzing and selecting the best-balanced solutions to requirements. |
| Interface Management | Identifies and manages the interactions between segments within a system or interactions with other peer systems. |
| Specialty Engineering | Analyzes the system, requirements, functions, solutions, and/or interfaces using specialized skills and tools. Assists in the derivation of requirements, synthesis of solutions, selection of alternatives, and validation and verification of requirements. |
| Integrity of Analyses | Ensures that the analyses provide the required level of fidelity and accuracy. |
| Risk Management | Identifies, analyzes, and manages the uncertainties of achieving program requirements by developing strategies to reduce the severity or likelihood of those uncertainties. |
| Configuration Management | Establishes and maintains consistency and manages change in the system performance, functional, and physical attributes. |
| Validation and Verification | Determines if system requirements are correct. Determines that the solution meets the validated requirements. |
| Lifecycle Engineering | Identifies and manages requirements for system lifecycle attributes including real estate management, deployment and transition, integrated logistics support, |

| System Engineering Element | Purpose of Element |
|---|---|
| | sustainment/technology evolution, and disposal. |
| System Engineering Process Management | Manages and maintains SE processes to meet FAA goals. Gains agency-wide skill and standardization by continuously improving the effectiveness and efficiency of SE processes and tools. |

Each program shall determine which SE elements and products are applicable to its success. A program shall justify its "tailoring out" of specific SE elements and products that are not deemed necessary. This tailoring and associated justification shall be captured in appropriate planning documents. The appropriate degree of tailoring for any SE process activity and its product(s) is determined by:

- Stakeholder needs
- Level of complexity of the program
- Need for communication of what activity is being performed (across members of a program team, across organizations, and/or over time to support future activities)
- Level of risk that is acceptable within the program

## 1.3 System Engineering Management Organization

SE addresses the translation of stakeholder needs into system requirements and facilitates the process by which the specification of systems and/or components satisfies those requirements. Although programs differ in underlying requirements, SE provides a logical sequence of steps toward the derivation of good requirements and the transformation into solutions regardless of the program's size or complexity. These steps generate a series of work products that specify characteristics of systems (at any level), demonstrate and document the traceability to stakeholder needs (expressed or implied), and define how the requirements are validated and the systems (and associated components) are verified. To maximize effectiveness, SE commences before any significant product development activities and continues throughout the program's lifecycle. If performed correctly, SE helps to ensure that program execution is right from the start and that if problems are encountered, they are detected and resolved early, which reduces program cost and risk as shown in Figure 1.3-1.

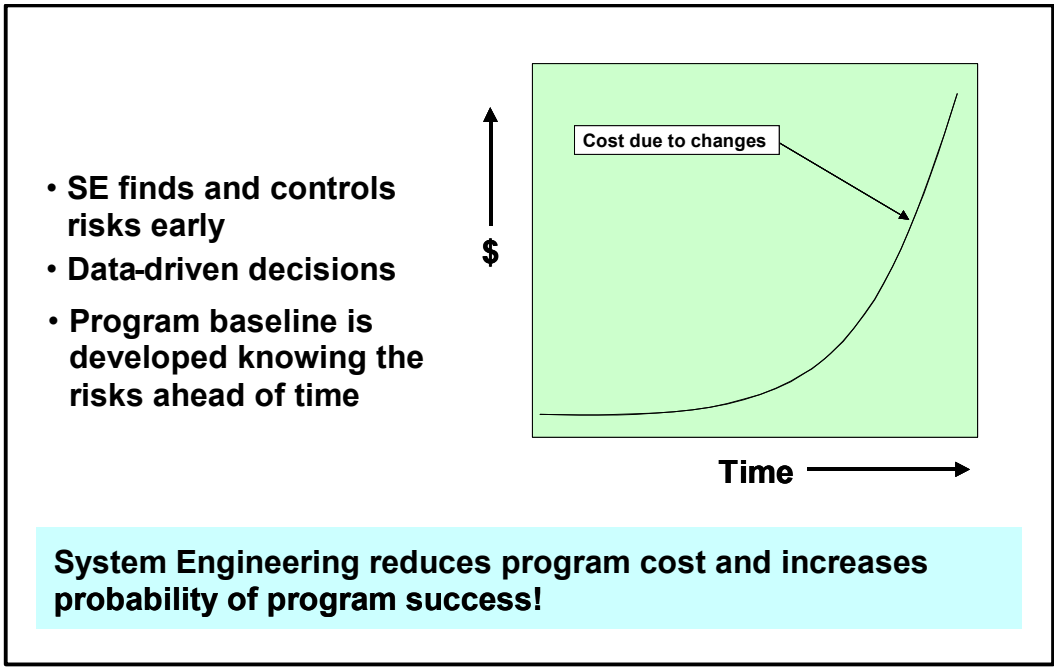



**Figure 1.3-1. Benefits of System Engineering**

### 1.3.1 Process-Based Management and System Engineering

The process flow for the "Perform System Engineering Process" appears in Figure 1.3-2. This model is a top-level process model that does not reflect all interactions among the SE processes. The details of these element interactions are discussed in Chapters 3 and 4. SE applies to all product development activities and is used throughout the program's lifecycle. This model shows the 12 lower-level processes and the 13th overarching process management element (the shaded blocks) that make up FAA SE. (The Validation and Verification process is depicted twice in the model but is considered one process.) Note that the figure does not show the iterative nature of SE by which each of the processes may be repeated multiple times throughout product development and program lifecycles at each level of system decomposition. It is the systematic iteration through these processes that leads to success. Throughout the manual, there are references to the FAA integrated Capability Maturity Model (iCMM) process areas and base processes. These references assist the user in complying with audits that are performed in support of iCMM.

SE provides key inputs to determining a solution as part of the Investment Analysis (IA) and Solution Implementation (SI) lifecycle phases. SE is initiated by the identification of a need (capability shortfall) in the existing system. This need drives the generation of a mission need statement (MNS). The SE outputs are used for subsequent iterations and provide the full set of requirements for the system being developed.

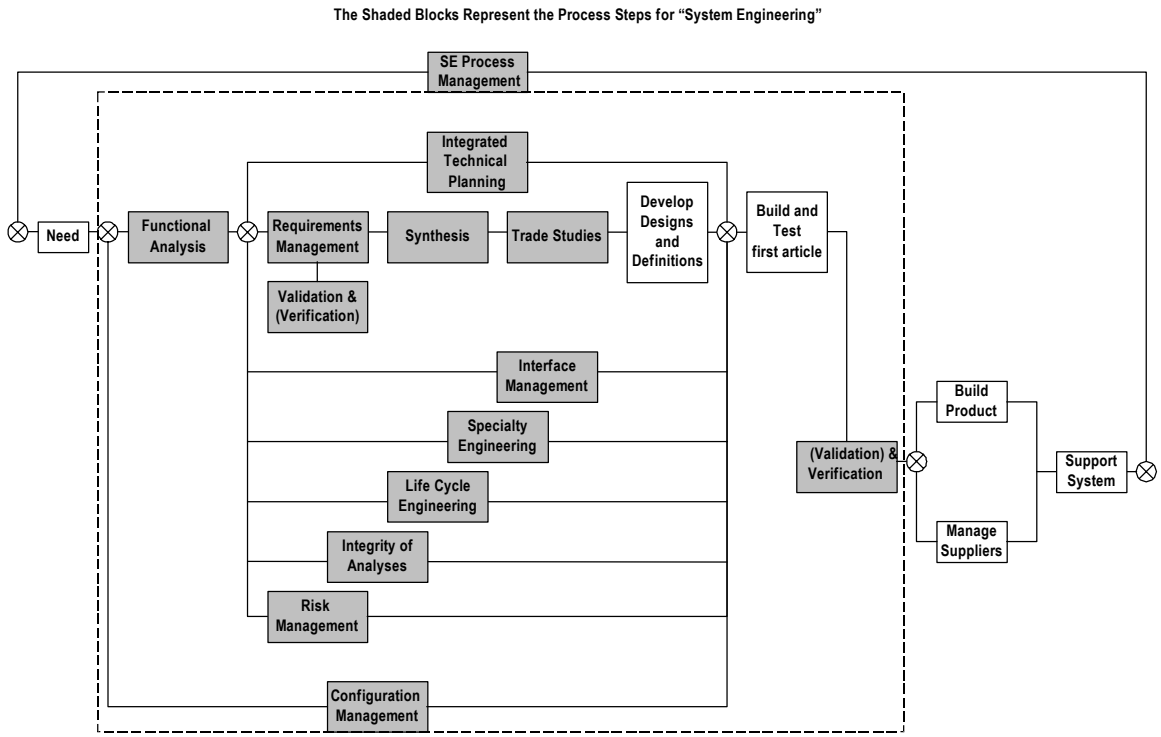



**Figure 1.3-2. Perform System Engineering Process Model**

### 1.3.2 System Engineering Process Descriptions

The SE process descriptions in Chapter 4 include the following information:

- **Process Definition.** Included are the purpose for carrying out the specific SE process and a narrative description of the specific SE process. This narrative discusses the function for the process (what to do). Program implementers may use this information to tailor specific activities to align with the development events of the program.
- **Process-Based Management (PBM) Charts.** Each SE element section in Chapter 4 contains a standard template that uses PBM charts to describe the SE element process. The templates indicate the major steps of the SE process, inputs to the process and associated providers, possible outputs generated, and associated product customers (from an SE view). The SEM also identifies the supplying (inputs) and using (outputs) processes that are used during process implementation to establish necessary program communication, documentation, and review activities.

Each SE process is broken down into its major workflow tasks, which are also shown in PBM chart form.

- **How To Do It.** The SEM discusses specific approaches or techniques that are used to implement each SE process and provides guidance for selecting the right approach for a given program phase. It summarizes the key points, focusing on the what and why, as well as the how.
- **Inputs.** This category includes information from external sources or from other processes that is received during the conduct of the process or that initiates the process.
- **Outputs.** This category includes information developed during and by the conduct of the process.
- **Entrance Criteria.** This category is what is required to start the process.

- **Exit Criteria.** This category includes activities that are performed when the process is complete.
- **Metrics.** This category includes examples of metrics that may be used to measure the level of performance for the process, as well as the work products generated by the process.
- **Methods/Tools.** This category includes specific tools or methods that are necessary (or desirable) to efficiently implement the process as described. They also guide the user as to what is available within the AMS FAA Acquisition System Toolset (FAST) (http://fast.faa.gov/).
- **Examples.** This category includes examples of SE work products as well as standard templates for producing the various SE work products. Examples may be contained either within a particular section of Chapter 4, an appendix to the SEM, or on the FAA's intranet, in which case a reference uniform resource locator (URL) will be provided.
- **References.** This category includes documents from the government, industry, and academia that cover relevant topics regarding that section.

# 2 OVERVIEW OF SYSTEM ENGINEERING

This section traces several key developments and lessons learned that led to today's championing of SE as a powerful approach to organizing and conducting complex programs, such as those found in the NAS. SE continues to evolve, with an emphasis on stronger commercial- and team-based engineering organizations, as well as organizations without technical products. Before World War II, architects and civil engineers were, in effect, system engineers who worked on large, primarily civil, engineering projects, including the Egyptian pyramids, Roman aqueducts, Hoover Dam, the Golden Gate Bridge, and the Empire State Building, while other architects worked on trains and large ships. However, "early" system engineers operated without any theory or science to support SE. Thus, they lacked defined and consistently applied processes or practices. During World War II, a program manager and chief engineer might oversee development of an aircraft program, while others managed key subsystems, such as propulsion, controls, structure, and support systems, leading to a lack of uniformity throughout the process.

Some additional SE elements, such as operations research and decision analysis, gained prominence during and after World War II. Today, with more complex requirements and systems, chief engineers use SE to develop requirements and to integrate the activities of the program teams.

SE began to evolve as a branch of engineering during the late 1950s. At this time—when both the race to space and the race to develop missiles equipped with nuclear warheads were considered absolutely essential for national survival—the military services and their civilian contractors were under extreme pressure to develop, test, and place in operation nuclear-tipped missiles and orbiting satellites. In this climate, the services and their contractors sought tools and techniques to improve system performance (mission success) and program management (technical performance, delivery schedule, and cost control). Engineering management evolved, standardizing the use of specifications, interface documents, design reviews, and formal configuration management. The advent of hybrid and digital computers permitted extensive simulation and evaluation of systems, subsystems, and components that facilitated accurate synthesis and tradeoff of system elements.

The lessons learned with development programs led to innovative practices in all phases of high-technology product development. A driving force for these innovations was attainment of high-system reliability. Some examples of changes introduced during the period are:

- Parts traceability
- Materials and process control
- Change control
- Product accountability
- Formal interface control
- Requirements traceability

## 2.1 What Is System Engineering?

Beyond the definition used in the “Introduction” (Chapter 1), SE is an overarching process that trades off and integrates elements within a system’s design to achieve the best overall product and/or capability known as a system. Although there are some important aspects of program management in SE, it is still much more of an engineering discipline than a management discipline. SE requires quantitative and qualitative decisionmaking involving tradeoffs, optimization, selection, and integration of the results from many engineering disciplines.

SE is iterative—it derives and defines requirements at each level of the system, beginning at the top (the NAS level) and propagating those requirements through a series of steps that eventually leads to a physical design at all levels (i.e., from the system to its parts). Iteration and design refinement lead successively to preliminary design, detail design, and final approved design. At each successive level, there are supporting lower-level design iterations that are necessary to gain confidence for decisions. During these iterations, many concept alternatives are postulated, analyzed, and evaluated in trade studies. These iterative activities result in a multi-tier set of requirements. These requirements form the basis for structured verification of performance. SE closely monitors all development activities and integrates the results to provide the best solution at all system levels.

## 2.2 What Is a System?

A system is an integrated set of constituent parts that are combined in an operational or support environment to accomplish a defined objective. These integrated parts include people, hardware, software, firmware, information, procedures, facilities, services, and other support facets. People from different disciplines and product areas have different perspectives on what makes up a system. For example, software engineers often refer to an integrated set of computer modules as a system. Electrical engineers might refer to a system as complex integrated circuits or an integrated set of electrical units. The FAA has an overarching system of systems called the NAS that includes, but is not limited to, all the airports, aircraft, people, procedures, airspace, CNS/ATM systems, and facilities.

At times, it is difficult to agree on what comprises a system, as it depends entirely on the focus of those who define the objective or function of the system. For example, if the objective is to print input data, a printer may be defined as the system. However, another might consider the electricity required for the printer. Expanding the objective to processing input data and displaying the results yields a computer as the system. Further expansion of the objective to include a capability for computing nationwide or worldwide and merging data/results to a database results in a computing network as the system, with the computer and printer(s) as subsystems of the system.

SE first defines the system at the top level, ensuring focus and optimization at that level, thus precluding narrow focus and suboptimization. It then proceeds to increasingly detailed lower levels until the system is completely decomposed to its basic elements. This hierarchy is described in the following paragraph.

### 2.2.1 System Hierarchy

A system may include hardware, software, firmware, people, information, techniques, facilities, services, and other support items. Figure 2.2-1 establishes a common reference for discussing the hierarchy of a system/subsystem within the NAS. Each system item may have its

associated hierarchy. For example, the various software programs/components that may reside in a system has a commonly accepted hierarchy as depicted in Figure 2.2-2. The depths of this common hierarchy may be adjusted to fit the complexity of the system. Simple systems may have fewer levels in the hierarchy than complex systems and vice versa. Because there may be varying hierarchal models referenced in the realm of SE, it is important for those who define the objective or function of a given system/subsystem to also lay out the hierarchal levels of the system in order to define the system's scope.

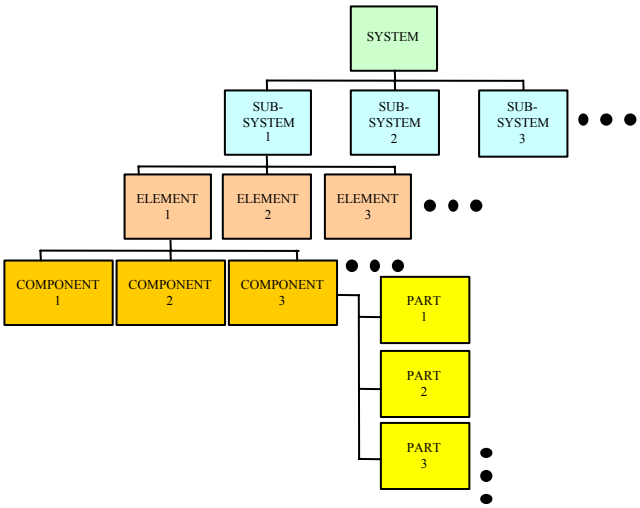



**Figure 2.2-1. System Hierarchy**

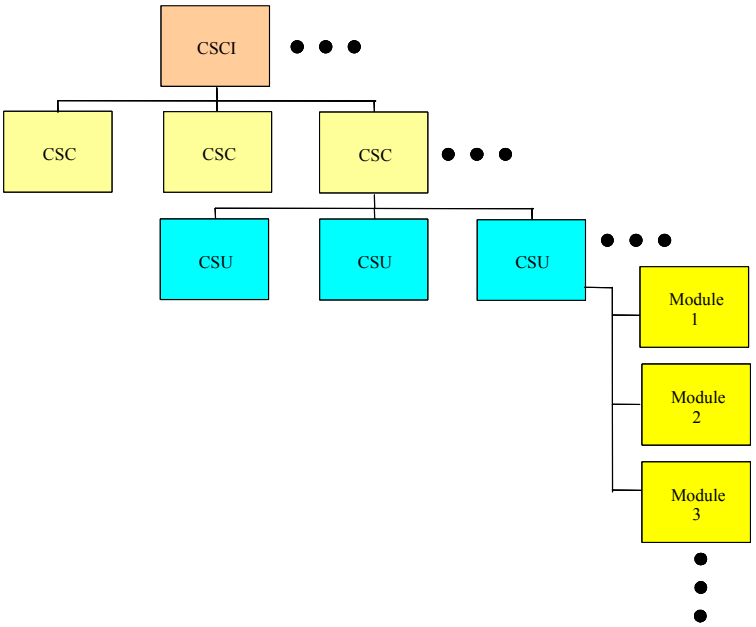



**Figure 2.2-2. Common Software Hierarchy**

Succeeding levels with the system/subsystem hierarchy are defined below:

- **System.** An integrated set of constituent parts that are combined in an operational or support environment to accomplish a defined objective. These parts include people, hardware, software, firmware, information, procedures, facilities, services, and other support facets.
- **Subsystem.** A system in and of itself (reference the system definition) contained within a higher-level system. The functionality of a subsystem contributes to the overall functionality of the higher-level system. The scope of a subsystem's functionality is less than the scope of functionality contained in the higher-level system.
- **Element.** An integrated set of components that comprise a defined part of a subsystem (e.g., the fuel injection element of the propulsion subsystem).
- **Component.** Composed of multiple parts; a clearly identified part of the product being designed or produced.
- **Part.** The lowest level of separately identifiable items within a system.
- **Software.** A combination of associated computer instructions and computer data definitions required to enable the computer hardware to perform computational or control functions.
- **Computer Software Configuration Item (CSCI).** An aggregation of software that is designed for configuration management and treated as a single entity in the Configuration Management process (Section 4.11).
- **Computer Software Component (CSC).** A functionally or logically distinct part of a CSCI, typically an aggregate of two or more software units.
- **Computer Software Unit (CSU).** An element specified in the design of a CSC that is separately testable or able to be compiled.
- **Module.** A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading.

## 2.3 Why Use System Engineering?

The need for effective SE is most apparent with large, complex system developments, such as weapons and transportation systems. However, SE is also important in developing, producing, deploying, and supporting much smaller systems, such as cameras and printers. The growing complexity in development areas has increased the need for effective SE. For example, about 35 years ago in the semiconductor industry, a single chip was no more complex than a series of a few gates or, at most, a four-stage register. Today, Intel's Pentium processor is far more complex, which immensely expands the application horizon but demands far more sophisticated analysis and discipline in design.

The movement to concurrent engineering as the technique for performing engineering development is actually performing good SE. SE provides the technical planning and control mechanisms to ensure that the activities/results of concurrent engineering meet overall system requirements.

A driving principle for SE is the teaming that often occurs during development programs. In this case, teaming is among several entities that may have different tools, analysis capabilities, and

so on. SE principles defined in this manual may provide an improved ability to plan and control activities that require interaction/interface across boundaries.

The strongest argument for using the SE processes is that they increase the likelihood that needs are fully and consistently met in the final product. SE delivers first-time quality and a satisfied stakeholder.

# 3 SYSTEM ENGINEERING IN THE ACQUISITION MANAGEMENT SYSTEM PROGRAM LIFECYCLE

## 3.1 Functional View of the System Engineering Process

To better focus on the SE processes as they relate to each phase of the AMS, Figure 3.1-1 provides a high-level view of the various SE processes and how they functionally interact. These functional interfaces only represent the predominant interaction between each process. The interaction between processes at a lower level is much more involved (i.e., Figure 3.1-1 is a simplified view and does not depict all the ways that processes interface). Figure 3.1-2 is a functional $N^2$ diagram of SE that shows the actual work products exchanged between the various SE processes shown in Figure 3.1-1.

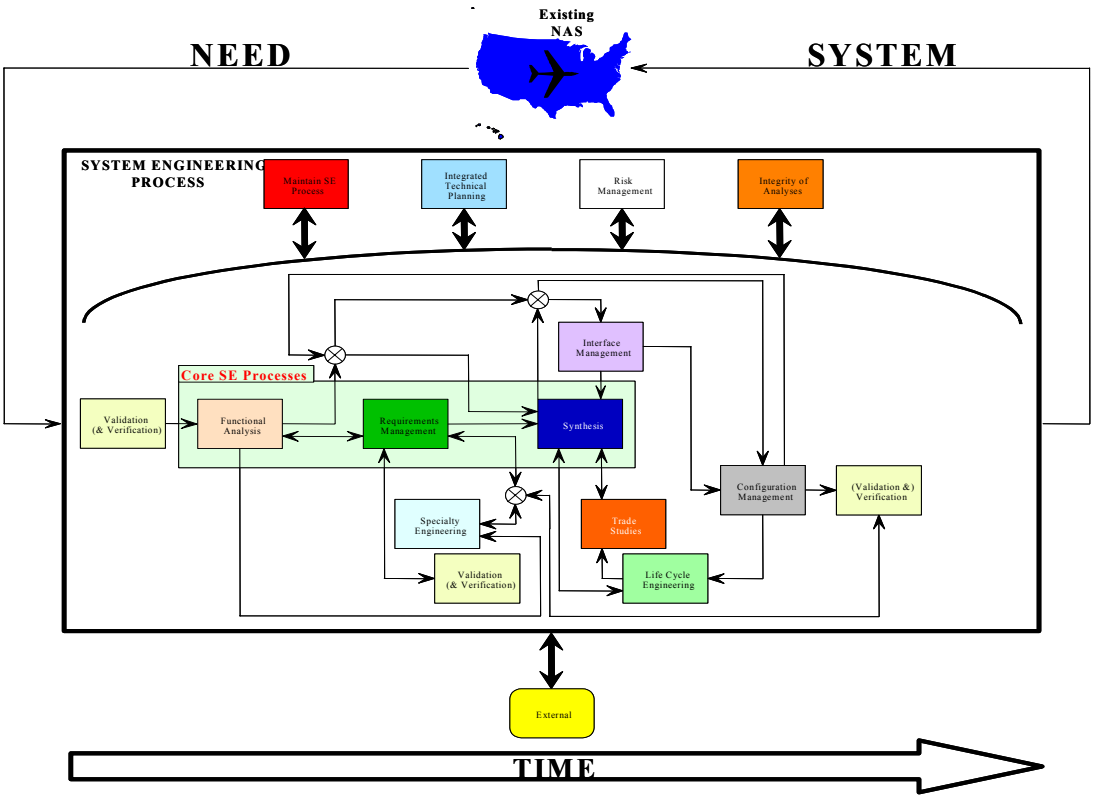



**Figure 3.1-1. Functional Flow Diagram of System Engineering**

In Figure 3.1-2, each SE process is laid out from left to right to notionally depict when in time each process is employed relative to another. The time arrow is *not* relative to the AMS lifecycle phases. It is recommended to note that overall SE, and many of the interactions at the lower levels, may be iterative in nature; thus, the left to right timeline is notional.

Figure 3.1-1 indicates that SE is initiated when there is a need; that is, a recognized shortfall in capability within the NAS. For example, the stakeholder need may arise as a result of a new service to be provided or with the advent of technological innovations to be leveraged to reap improvements in capacity, security, and/or safety. Once the need is validated, the Functional Analysis process (Section 4.4) is performed to develop a Concept of Operations (CONOPS). The Requirements Management process (Section 4.3) uses the CONOPS to develop an MNS,

which is then fed back to Functional Analysis as input to develop the highest level of functional architecture for the new or modified system. The Requirements Management process uses this high-level functional architecture, as well as inputs from Specialty Engineering analyses, to develop requirements. These requirements are validated via the Validation and Verification process (Section 4.12). The interaction between Functional Analysis and Requirements Management is iterative, as the functional architecture and resulting requirements are decomposed to a level necessary to the appropriate requirements that describe the needed system characteristics. Synthesis (Section 4.5) then develops the physical architecture or design solution to those requirements.

Along with these initial SE activities, three overarching processes that interact with all SE processes are employed. These overarching processes continue throughout the system's lifecycle and are as follows:

- Integrated Technical Planning (Section 4.2)
  - Provides the technical guidance tools required to track and manage program activity
- Risk Management (Section 4.10)
  - Provides an organized, systematic decisionmaking approach to identify risks that affect achievement of program goals
  - Analyzes identified risks
  - Mitigates risks effectively
  - Tracks the progress of the mitigation efforts
- Integrity of Analyses (Section 4.9)
  - Ensures the provision of credible, useful, and sufficient data/results for program management's decisionmaking process
  - Ensures the integrity and fidelity of the various analysis tools

Once a valid set of requirements is obtained, the Synthesis process (Section 4.5) is initiated to define system elements and to refine and integrate these elements into a physical architecture. In addition to the requirements input into the Synthesis process, the functional architecture is provided to clarify and bound the system. The Trade Studies process (Section 4.6) and the Lifecycle Engineering process (Section 4.13) supply cost estimates to support the Synthesis process, which ultimately determines the design alternative that best satisfies the identified stakeholder need.

Interface Management (Section 4.7) plays a key role in ensuring that the various internal system pieces are coordinated as well as integrated with external systems. As the total system is decomposed via iterative interaction of Functional Analysis, Requirements Management, and Synthesis, physical and functional interfaces are identified and managed.

The results of these SE activities are continually brought under Configuration Management (Section 4.11). The system is developed according to the baseline design and verified with the Validation and Verification process (Section 4.12). With the system verified as able to meet the identified stakeholder need, it is deployed into the NAS. Although the discussion of this simplified view and description of SE was sequential, SE is truly iterative and employed continuously throughout the lifecycle of the system.

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **EXTERNAL** | FAA Policy<br>Integrated Program Schedule<br>Corporate Strategy and Goals | Constraints<br>FAA Management Decisions<br>Govt & Intl Regulations & Statutes<br>Legacy System<br>Stakeholder Needs<br>Standards<br>Technology | FAA Management Decisions<br>Legacy System | Constraints<br>Legacy System<br>Market Research<br>Standards<br>Technology | Market Research<br>Technology<br>Constraints<br>Integrated Program Schedule | FAA Policy<br>Legacy System<br>Standards<br>Interface Change Request | FAA Policy<br>Standards | Technology<br>Constraints | External Environmental Forces<br>Constraints<br>Technology<br>Concerns/Issues<br>Integrated Program Schedule | FAA Policy | Need<br>Standards | Technology<br>Test & Assessment Articles | Constraints<br>External Environmental Forces<br>Govt & Intl Regulations & Statutes<br>Market Research<br>FAA Policy<br>Technology<br>Concerns/Issues<br>Integrated Program Schedule | |
| Integrated Program Plan | **INTEGRATED TECHNICAL PLANNING** | Integrated Lifecyle Plan<br>Integrated Program Plan<br>NAS Architecture<br>SEMP | Integrated Program Plan<br>SEMP | Integrated Program Plan<br>SEMP | Constraints<br>Integrated Logisitics Support Plan<br>Integrated Program Plan<br>SEMP | Integrated Program Plan<br>NAS Architecture<br>SEMP | Integrated Lifecyle Plan<br>Integrated Program Plan<br>NAS Architecture<br>SEMP | Integrated Program Plan<br>SEMP | Concerns/Issues<br>Integrated Program Plan<br>SEMP | Integrated Program Plan<br>NAS Architecture<br>SEMP | Integrated Program Plan<br>NAS Architecture<br>SEMP | Integrated Program Plan<br>Master Verification Plan<br>NAS Architecture<br>SEMP | Integrated Lifecyle Plan<br>Integrated Program Plan<br>NAS Architecture<br>SEMP | |
| Requirements<br>RVCD | Planning Criteria | **REQUIREMENTS MANAGEMENT** | Mission Need Statement<br>Requirements | Requirements<br>RVCD | Constraints<br>Requirements | Requirements<br>Mission Need Statement | Requirements<br>RVCD | Requirements<br>Tools/Analysis Requirements | Requirements<br>RVCD<br>Concerns/Issues | Requirements | Requirements | Requirements<br>VRTM | Mission Need Statement<br>Requirements | |
| | Concept of Operations<br>NAS CONOPS<br>Planning Criteria | Concept of Operations<br>Functional Architecture<br>OSED | **FUNCTIONAL ANALYSIS** | Functional Architecture<br>OSED | Constraints<br>Functional Architecture<br>OSED | Concept of Operations<br>Functional Architecture<br>OSED | Concept of Operations<br>Functional Architecture<br>OSED | Tools/Analysis Requirements | Concerns/Issues | Functional Architecture<br>OSED | Concept of Operations<br>Functional Architecture<br>OSED | Functional Architecture | OSED | |
| | Planning Criteria | Physical Architecture | Physical Architecture | **SYNTHESIS** | Physical Architecture<br>Design Constraint<br>Constraints<br>Description of Alternatives | Operational Concept Demo<br>Physical Architecture | Description of Alternatives<br>Physical Architecture | Tools/Analysis Requirements | Concerns/Issues | Configuration Description | Operational Concept Demo<br>Physical Architecture | Physical Architecture | Design Constraint<br>Operational Concept Demo | |
| Trade Study Reports | Planning Criteria | Trade Study Reports | | Trade Study Reports | **TRADE STUDIES** | Trade Study Reports | | Tools/Analysis Requirements | Concerns/Issues | | | | Trade Study Reports | |
| | Planning Criteria | Interface Control Documents<br>Interface Requirements Documents | Interface Control Documents | Interface Requirements Documents | Constraints | **INTERFACE MANAGEMENT** | Interface Control Documents | | Concerns/Issues | Interface Revision Proposal<br>Interface Requirements Documents<br>Interface Control Documents | Interface Requirements Documents | Interface Control Documents | | |
| Certification Package | Planning Criteria | Design Analysis Reports<br>Requirements | Design Analysis Reports | Design Analysis Reports | Design Analysis Report<br>Constraints | Design Analysis Reports | **SPECIALTY ENGINEERING** | Design Analysis Reports<br>Tools/Analysis Requirements | Concerns/Issues<br>Design Analysis Reports | | Demonstrations<br>Design Analysis Reports | Demonstrations<br>Verification Criteria<br>Design Analysis Reports | Design Analysis Reports | |
| Credible Analysis Results | Analysis Criteria<br>Planning Criteria | Analysis Criteria | Analysis Criteria | Analysis Criteria | Analysis Criteria<br>Constraints | | Analysis Criteria | **INTEGRITY OF ANALYSES** | Concerns/Issues<br>Analysis Criteria | Credible Analysis Results | | ~~Analysis Criteria~~ | Analysis Criteria | |
| Program Risk Register<br>Program Risk Summary<br>Risk Mitigation Plan Summary<br>Risk Mitigation Plans | Planning Criteria | Risk Mitigation Plans | | Risk Mitigation Plans | Constraints | | | Tools/Analysis Requirements | **RISK MANAGEMENT** | | | | | |
| | Planning Criteria | Approved Baseline Changes<br>Configuration Status Report<br>Updated Baselines | | Configuration Status Report<br>Updated Baselines<br>Approved Baseline Changes | Constraints | | Baselines | Baselines | Baselines<br>Concerns/Issues | **CONFIGURATION MANAGEMENT** | | Validation Reports<br>Updated Baselines<br>Configuration Status Report<br>Approved Baseline Changes | Updated Baselines<br>Configuration Status Report<br>Approved Baseline Changes | |
| | Planning Criteria | Validation Reports | Validated Need | | Constraints | | Validation Reports | | Concerns/Issues | | **VALIDATION** | | | |
| | Planning Criteria | RVCD<br>VRTM | | | Constraints | | | Tools/Analysis Requirements | Concerns/Issues | | | **VERIFICATION** | | |
| | Planning Criteria | | | Lifecycle Cost Estimate | Lifecycle Cost Estimate<br>Constraints | | | Tools/Analysis Requirements | Concerns/Issues<br>Constraints | | | | **LIFE CYCLE ENGINEERING** | |
| NAS SEMP | NAS SEMP | NAS SEMP | NAS SEMP | NAS SEMP | NAS SEMP | NAS SEMP | NAS SEMP | NAS SEMP | NAS SEMP | NAS SEMP | NAS SEMP | NAS SEMP | NAS SEMP | **Maintain SE Process** |

**Figure 3.1-2. System Engineering Functional $N^2$ Diagram**

## 3.2 Relationship of the System Engineering Processes to the Acquisition Management System Program Lifecycle

The program lifecycle includes all activities and products associated with a system, from initial inception to disposal and elimination, which falls in line with the global aspects of SE's definition. Definitions of the program lifecycle phases serve different purposes for different system processes. System sponsors shall use these phases and their associated milestones (e.g., Mission Need Decision (MND), Initial and Final Investment Decision, and In-Service Decision) to determine whether to continue or terminate the endeavor. Thus, the phases shall be used to measure a program's progress and develop input to the Joint Resources Council (JRC), which ultimately makes the noted decisions. Each program decision milestone is associated with a review, which are as follows:

- **JRC-1/MND milestone.** A briefing for review by the JRC is conducted before the MND.
- JRC-2a/Initial Investment Decision milestone. A briefing for review by the JRC is conducted before the Initial Investment Decision.
- **JRC-2b/Final Investment Decision milestone.** An Initial System Requirements review (ISRR) and a briefing for review by the JRC are conducted before the Final Investment Decision.
- **JRC-3/In-Service Decision milestone.** The in-service review (ISR) checklist is reviewed and briefed to the appointed decision authority before the In-Service Decision.

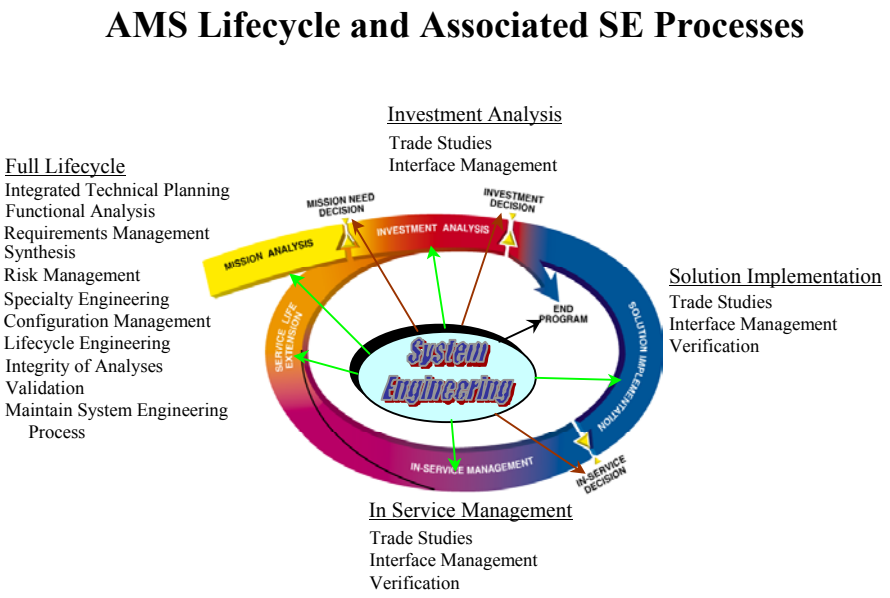



**Figure 3.2-1. Focus of System Engineering Effort in the Acquisition Management System Lifecycle Phases**

As shown in Figure 3.2-1, SE lays out and supports the technical and programmatic activities as the program moves through these phases. The SE processes support the lifecycle phases defined by the AMS and provide management visibility into the operation of the program, facilitating risk reduction through early identification of issues and, thus, allowing better management. Therefore, cost is reduced through earlier recognition and correction of problems (Figure 1.3-1). Support organizations are able to gauge and plan their work to support each phase.

Any or all of the SE processes may be applied throughout the entire lifecycle, regardless of the phase. However, Figure 3.2-1 shows that the predominance of a specific SE process depends on the phase of the lifecycle. For example, there are certain elements of SE employed during the Mission Analysis (MA) phase more so than during the SI phase. The Validation process (Section 4.12) is used more heavily during MA when compared to Verification (Section 4.12). Likewise, the Verification process is used more heavily during SI when compared to Validation.

Because SE establishes system requirements and overall system architecture, it is a dominating process during the MA and IA phases. During this time, mission needs, CONOPS, and requirements are analyzed and documented to support both functional and physical system decomposition to greater detail. Program plans are developed to guide and control subsequent activity consistent with the overall system requirements. Alternatives are developed and synthesized to identify the most viable approach. Trade Studies (Section 4.6) are conducted to provide input to the Synthesis process (Section 4.5) to facilitate selection of the most viable approach. The Trade Studies data supports the selection and approval to proceed with an Acquisition Program Baseline (APB).

During the SI phase, proposed system design alternatives are reviewed against requirements and selection criteria. When a system design is selected, iterations of SE are performed to completely decompose the system to a set of products (e.g., hardware, software, people) that may be built/programmed (or trained or otherwise provided) and verified. SE elements are used to maintain overall integration (cost, schedule, and technical) of the lower-level product development activities based on overall system requirements.

After the system design has been tested and accepted, SE activities shift to deployment and transition of the production systems to the field. As systems enter the In-Service Management lifecycle phase, SE is reiterated to ensure effective incorporation of system design changes due to problem fixes, new functionality, or product obsolescence. These activities continue through the remainder of the system's service life until disposal.

In addition to the functional view of SE (Figure 3.1-1), the SE model may be depicted as a cube (Figure 3.2-2) in order to emphasize the iterative nature of SE throughout the AMS lifecycle. The vertical dimension depicts the system hierarchy, which emerges from the iterative use of SE. The horizontal dimension indicates the SE elements, and the third dimension contains the phases of development. This representation shows that SE applies across all phases and to all levels of the system hierarchy.

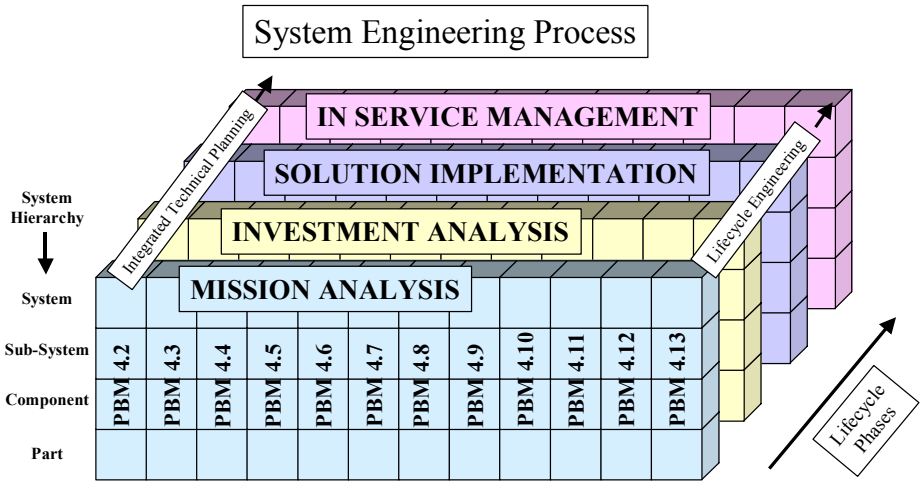



**Figure 3.2-2. The System Engineering Process Cube**

Each major program phase has a progressively more mature, detailed, and changing focus. SE responds with a different emphasis as the program progresses through its development and deployment. The application of SE, as defined in this manual, shall be continually tailored to the specific needs of each program and its current phase.

## 3.3 Program Lifecycle

This section addresses each phase of the AMS lifecycle and correlates the phases to SE activities. Data flow diagrams (DFD) highlight the SE processes and work products that are predominant during that time.

In addition, a table is included with each phase discussion that:

- Identifies the SE work products that are inputs and/or outputs to the addressed phase
- Identifies work products generated from processes external to SE that are necessary to initiate SE activities within the given phase

The following legend in Table 3.3-1 applies to each of these tables.

**Table 3.3-1. Legend for System Engineering Work Products Inputs/Outputs Tables**

| Abbreviation | | Meaning |
|---|---|---|
| C | = | Conceptual draft (precedes initial draft); the general notion and structure of the document has been created with minimal content |
| CM | = | Configuration Management |
| EXT | = | External to SE |
| F | = | Final draft; document is complete, accurate, and awaiting signature |
| FA | = | Functional Analysis |
| I | = | Initial draft; the document has been populated with the majority of required content, but it still requires review for accuracy of information |
| IA | = | Integrity of Analyses |
| IM | = | Interface Management |
| ISRR | = | Initial System Requirements Review |
| LC | = | Lifecycle Engineering |
| MSE | = | Maintain System Engineering |
| RM | = | Requirements Management |
| RSK | = | Risk Management |
| S | = | Synthesis |
| SD | = | Sustaining Document<br>For work products that are formal documents, the documents are sustained in the given phase<br>For work products that are not formal documents, the products are introduced, further developed, or sustained in the given phase |
| SpecEng | = | Specialty Engineering |
| ITP | = | Integrated Technical Planning |
| TS | = | Trade Studies |
| Val | = | Validation |

Table 3.3-2 gives a high-level view of the various SE work products and the timeframe during which they are developed.

**Table 3.3-2. System Engineering Work Product/ Acquisition Management System Milestone Matrix**

| | PROGRAM MILESTONES | | | | |
|---|---|---|---|---|---|
| **WORK PRODUCT** | **JRC1** | **JRC2A** | **ISRR** | **JRC2B** | **JRC3** |
| Acquisition Program Baseline[1] | | C | I | F | |
| Analysis Criteria | I | F | SD | SD | SD |
| Approved Baseline Changes | | | | | SD |
| Baselines | | | | SD | SD |
| Certification Package | | | | I | F |
| Concept of Operations | I | F | | | |
| Concerns/Issues | SD | SD | SD | SD | SD |
| Configuration Description | | | | I | F |
| Configuration Status Report | | | | | SD |
| Constraints | SD | SD | SD | SD | SD |
| Corporate Strategy and Goals | SD | SD | SD | SD | SD |
| Credible Analysis Results | SD | SD | SD | SD | SD |
| Demonstrations | | SD | SD | SD | SD |
| Description of Alternatives | I | F | | | |
| Design Analysis Reports | SD | SD | SD | SD | SD |
| Design Constraint | SD | SD | SD | SD | SD |
| External Environmental Forces | SD | SD | SD | SD | SD |
| FAA Management Decisions | SD | SD | SD | SD | SD |
| FAA Policy | SD | SD | SD | SD | SD |
| Functional Architecture | C | I | F[1] | SD | SD |
| Government and International Regulations and Statutes | SD | SD | SD | SD | SD |
| Integrated Lifecycle Plan | C | I | F | SD | SD |
| Integrated Program Plan | C | I | F | SD | SD |
| Integrated Program Schedule | C | I | F | SD | SD |
| Interface Change Request | | | | | SD |
| Interface Control Documents | | | C | I | F |
| Interface Requirements Documents | | I | F | SD | SD |
| Interface Revision Proposal | | | | | SD |
| Legacy System | SD | SD | SD | SD | SD |
| Lifecycle Cost Estimate | C | I | | F | |
| Market Research | SD | SD | SD | | |
| Master Verification Plan | | I | F | SD | SD |
| Mission Need Statement | F | | | | |
| NAS Architecture | SD | SD | SD | SD | SD |
| NAS Concept of Operations | SD | SD | SD | SD | SD |
| NAS System Engineering Management Plan | SD | SD | SD | SD | SD |
| Need | SD | | | | |
| Operational Concept | | SD | SD | | |

| | PROGRAM MILESTONES | | | | |
|---|---|---|---|---|---|
| WORK PRODUCT | JRC1 | JRC2A | ISRR | JRC2B | JRC3 |
| Demonstrations | | | | | |
| Operational Services and Environmental Description | C | I | F | | |
| Physical Architecture | | C | | I | F |
| Planning Criteria | SD | SD | SD | SD | SD |
| Program Risk Register | | | SD | SD | SD |
| Program Risk Summary | | SD | SD | SD | SD |
| Requirements | I | | F[2] | SD | SD |
| Requirements Verification Compliance Document | | | | | F |
| Risk Mitigation Plan Summary | | SD | SD | SD | SD |
| Risk Mitigation Plans | | SD | SD | SD | SD |
| Stakeholder Needs | I | F | SD | SD | SD |
| Standards | | | SD | SD | SD |
| Statement of Work[1] | | | I | F | |
| System Engineering Management Plan | | I | F | | |
| Technology | SD | SD | SD | SD | SD |
| Test and Assessment Articles | | | | | F |
| Tools/Analysis Requirements | | SD | SD | SD | SD |
| Trade Study Reports | | SD | SD | SD | SD |
| Updated Baselines | | | | | SD |
| Validated Need | F | | | | |
| Validation Reports | SD | SD | SD | SD | SD |
| Verification Criteria | | SD | SD | SD | SD |
| Verification Requirements Traceability Matrix | C | I | F | SD | SD |
| Work Breakdown Structure[1] | C | I | F | | |

1. This work product is not produced via the SE process. It is listed here as a point of reference since SE provides substantial input into the development of this work product.
2. This does not imply that there is no further decomposition (e.g., "Final" requirements at this point in time is with respect to the final Requirements Document (fRD), yet further decomposition takes place to generate a specification).

### 3.3.1 Mission Analysis Phase

#### 3.3.1.1 Mission Analysis Phase Objectives

The basic objective of the MA phase is to correctly identify and quantify a need so that the FAA may begin a program to resolve that need. The primary outputs of this phase are the MNS or the modification of an existing MNS and design constraints. The MA phase ends with an MND. In most cases, the MA consists of activities to validate high-level needs and to seek approval to proceed to the next phase. It has two dimensions: a technical dimension and a program-planning dimension. The technical dimension is to ensure that a complete understanding of the

demand for services has been identified and quantified, which is accompanied by identification and quantification of existing and projected supply of services. NAS shortfalls and potential technological opportunities shall be identified and quantified. The program-planning dimension is to identify potential project-scope and estimated resource requirements.

#### 3.3.1.2 System Engineering Activities

Figure 3.3-1 is an overview of the primary SE activities that occur during MA. SE is initiated when a stakeholder need is recognized. If the stakeholder need is valid, SE continues to better understand functionally what is required to best meet the stakeholder need. A CONOPS is developed via Functional Analysis (Section 4.4) and is used in Requirements Management (Section 4.3) to develop the MNS. The MNS is a primary SE output during the MA phase; it also drives the continued iterations of Functional Analysis and Requirements Management. The interaction of these two processes results in a high-level functional decomposition and, likewise, a high-level requirements decomposition. The resulting set of requirements is validated and is used, along with the high-level functional architecture, during the Synthesis process (Section 4.5) to develop a description of alternatives and associated design constraints. At this point in time, these alternatives and constraints are very high-level and are used as primary input into the IA phase to provide scope for the program.

In addition to the core Functional Analysis, Requirements Management, and Synthesis activities, other SE processes are initiated during the MA phase. These activities involve the technical planning necessary to provide proper guidance for SE activities throughout the system's lifecycle; identification of risks and plans to mitigate those risks; and establishment of analysis criteria for the various analyses that occur when designing the system. Any of the SE activities may surface concerns/issues to be processed by Risk Management (Section 4.10), as well as constraints to bound the activities of the Trade Studies process (Section 4.6) that occur during the follow-on phases.

Electronic Industries Association (EIA) 731-2 defines a constraint as (1) a restriction, limit, or regulation or (2) A type of requirement that is not tradeable against other requirements. Constraints describe what is not included in the program. These items define work that might be expected but will not be done. Often, these are defined in work scope statements given by project contributors during the cost definition process. This includes gathering stakeholder inputs on "needs" and "wants," system constraints (costs, technology limitations, and applicable specifications/legal requirements), and system "drivers" (such as competition capabilities, military threats, and critical environments). Tradeoffs should be done on the desirability of including a performance capability in the system versus a more affordable (or less risky) system approach. This tradeoff process often begins well before a firm set of needs is established and continues throughout the MA phase in which stakeholder interaction on specific items proposed may take place. Constraints may be further adjusted throughout later AMS phases. Like behavior deficiencies or shortfalls, these are excellent opportunities for preplanned product improvement. Funding, personnel, facilities, manufacturing capability, critical resources, or other reasons cause the existence of constraints. The reason for each constraint should be understood.

Risk always is present in the lifecycle of both commercial and military systems. The system may be intended for technical accomplishments near the limits of the state of the art, creating technical risk. System development may be rushed to deploy the system as soon as possible to meet an imminent threat, leading to schedule risk. All systems are funding-limited so that cost

risk is present. Risk can be introduced by external constraints or can develop from within the program, since technical risk can create schedule risk that in turn can create cost risk.

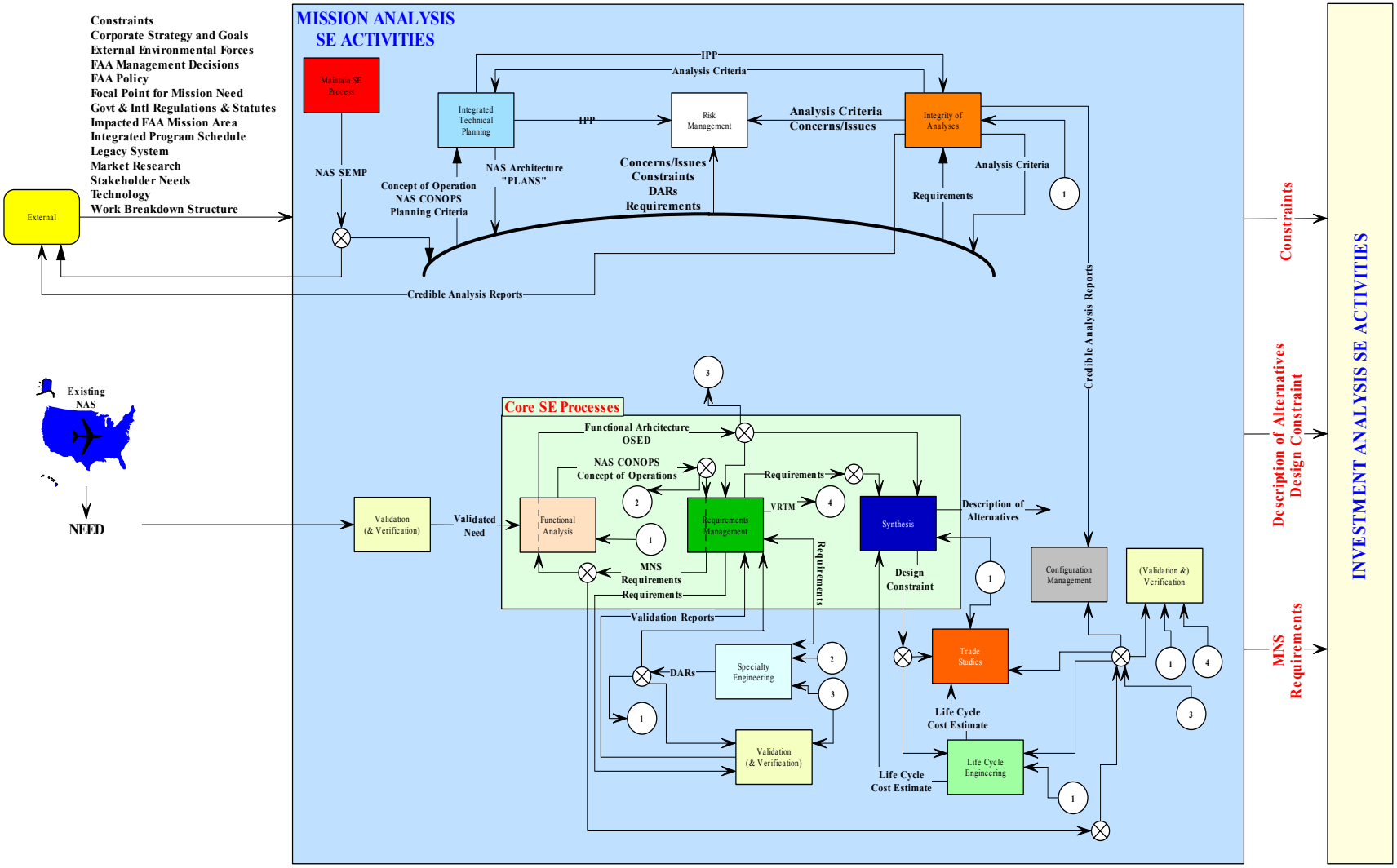



**Figure 3.3-1. Mission Analysis System Engineering Activities**

Table 3.3-3 summarizes the MA SE inputs and outputs.

**Table 3.3-3. Mission Analysis System Engineering Inputs and Outputs**

| WORK PRODUCT | PRODUCING SE ELEMENT | INPUT | OUTPUT |
|---|---|---|---|
| Analysis Criteria | IA | | I |
| Concept of Operations | FA | | I |
| Concerns/Issues | ALL | | SD |
| Constraints | ALL except TS | | SD |
| Corporate Strategy and Goals | EXT | SD | |
| Credible Analysis Results | IA | | SD |
| Description of Alternatives | S | | I |
| Design Analysis Reports | SpecEng | | SD |
| Design Constraint | S | | SD |
| External Environmental Forces | EXT | SD | |
| FAA Management Decisions | EXT | SD | |
| FAA Policy | EXT | SD | |
| Functional Architecture | FA | | C |
| Government and International Regulations and Statutes | EXT | SD | |
| Integrated Lifecycle Plan | ITP | | C |
| Integrated Program Plan | ITP | | C |
| Integrated Program Schedule | EXT | C | |
| Legacy System | EXT | SD | |
| Lifecycle Cost Estimate | LC | | C |
| Market Research | EXT | SD | |
| Mission Need Statement | RM | | F |
| NAS Architecture | ITP | SD | |
| NAS Concept of Operations | FA | SD | |
| NAS System Engineering Management Plan | MSE | SD | |
| Need | EXT | SD | |
| Operational Services and Environmental Description | FA | | C |
| Planning Criteria | ALL except ITP | | SD |
| Requirements* | RM | | I |
| Stakeholder Needs | EXT | SD | |
| Technology | EXT | SD | |
| Validated Need | Val | | F |
| Validation Reports | Val | | SD |
| Verification Requirements Traceability Matrix | RM | | C |
| Work Breakdown Structure | EXT | C | |

* Requirements expressed in the Initial Requirements Document (iRD).

NOTE: See Table 3.3-1 for legend.

#### 3.3.1.3 Mission Analysis Entrance Criteria

There are no other entrance criteria beyond the concept of a given “need” and approval to initiate SE efforts during the MA phase.

#### 3.3.1.4 Mission Analysis Exit Criteria

The following criteria shall be satisfied (to the satisfaction of program management and its review authority) before the program enters the IA phase:

- Completion of all work products identified as MA outputs to the version level specified
- Approval of the MND, thereby authorizing the program to proceed to the IA phase

#### 3.3.1.5 System Engineering Element Tasks in Mission Analysis

In addition to the tasks identified below, each SE Element active during this phase shall surface concerns/issues that present risk to the program, as well as any constraints that bound future Trade Studies (Section 4.6).

##### 3.3.1.5.1 Tasks for the Integrated Technical Planning Element

Although the major planning activities occur in the following IA phase( i.e., after the JRC1/MND) the following plans shall be completed by the end of MA to guide initial IA activities:

- Conceptual draft of the Integrated Program Plan (IPP) (with details for the mission needs analysis and IA items) to include the following planning information:
  - Configuration Management (Section 4.11) to control program documentation, including requirements that may change
  - Risk Management (Section 4.10) to lay out program policy for risk assessment and mitigation, including program preparation activities
  - Analysis Management (Section 4.9)
  - Requirements Management (Section 4.3)
- Conceptual draft of the Integrated Lifecycle Plan

##### 3.3.1.5.2 Tasks for the Requirements Management Element

The primary tasks of the Requirements Management process (Section 4.3) are:

- Develop the MNS (new or modified)
- Develop the Initial Requirements Document (iRD; new or modified) to include intended service life, the initial list of stakeholders, and a conceptual draft of the Verification Requirements Traceability Matrix (VRTM)

At this time, a means for tracking and tracing requirements (e.g., via a database) shall be established.

##### 3.3.1.5.3 Tasks for the Functional Analysis Element

Functional Analysis (Section 4.4) of the operational system begins in this phase by:

- Developing an initial draft of a CONOPS
- Developing a conceptual draft of the Operational Services and Environmental Description (OSED)
- Developing the conceptual version of the first-level functional architecture

Developing the functional architecture involves identifying at least the high-level functions to be performed to accomplish the required mission operations. Thus, high-level functional flow diagrams (FFD) are created. The decision to analyze the functions to lower levels depends on the presence of the right expertise and available information to conduct these analyses. Insofar as a conceptual product is defined, the primary rationale at this point is to provide a concise, clear description of the system for the MND without precluding any creativity in developing concepts to support the high-level functions.

#### 3.3.1.5.4 Tasks for the Synthesis Element

The role of Synthesis (Section 4.5) is to create sufficient alternative high-level concepts that might satisfy the service needs and perform the high-level functions to define the program space (i.e., to show the range of possibilities to explore). Focusing too soon on a feasible concept as *the* final solution—a typical difficulty—stifles creativity and reduces potential benefits and shall be avoided at all costs. Several wide-ranging, yet viable, concepts are valuable to inspire creativity in concept development. Synthesis activities during this phase include:

- Producing initial drafts of description of alternatives for at least two concepts in order to give the IA team a sufficient understanding of the program scope
- Identifying design constraints

#### 3.3.1.5.5 Tasks for the Specialty Engineering Element

During the MA phase, Specialty Engineering (Section 4.8) develops high-level design analysis reports (DAR) to support requirements development, Validation (Section 4.12) of existing requirements, Risk Management (Section 4.10), and Synthesis (Section 4.5) (if required).

#### 3.3.1.5.6 Tasks for the Integrity of Analyses Element

During the MA phase, the Integrity of Analyses process (Section 4.9) produces the analysis criteria for all immediately foreseen required analyses. Credible analysis results are generated for analyses initiated and completed during the MA phase.

#### 3.3.1.5.7 Tasks for the Validation Element

Validation (Section 4.12) is performed on the stakeholder need and initial high-level requirements.

#### 3.3.1.5.8 Tasks for the Lifecycle Engineering Element

Lifecycle Engineering (Section 4.13) develops a preliminary lifecycle cost estimate and a conceptual draft of the Integrated Lifecycle Plan.

### 3.3.2 Investment Analysis Phase

The IA phase of the AMS lifecycle has the following objectives:

- Further translate the mission needs into requirements and eventually into specifications
- Complete the CONOPS
- Complete the IPP and all additional program plans
- Complete the functional architecture to a level appropriate to requirements (i.e., those levels needed to support development of the final Requirements Document (fRD) or system specification)
- Develop potential solutions
- Analyze alternative solutions and determine the optimum solution from a NAS perspective
- List and analyze all programmatic risks
- Provide risk mitigation plans with associated costs
- Modify the architecture to the recommended solution
- Provide input for APB development

IA begins with the approval of a mission need and ends with an Investment Decision. There are two SE stages during the IA phase: the initial IA stage (or the JRC 2A stage) and the final IA stage (or the JRC 2B stage).

Figure 3.3-2 depicts these two stages as they relate to AMS milestones.

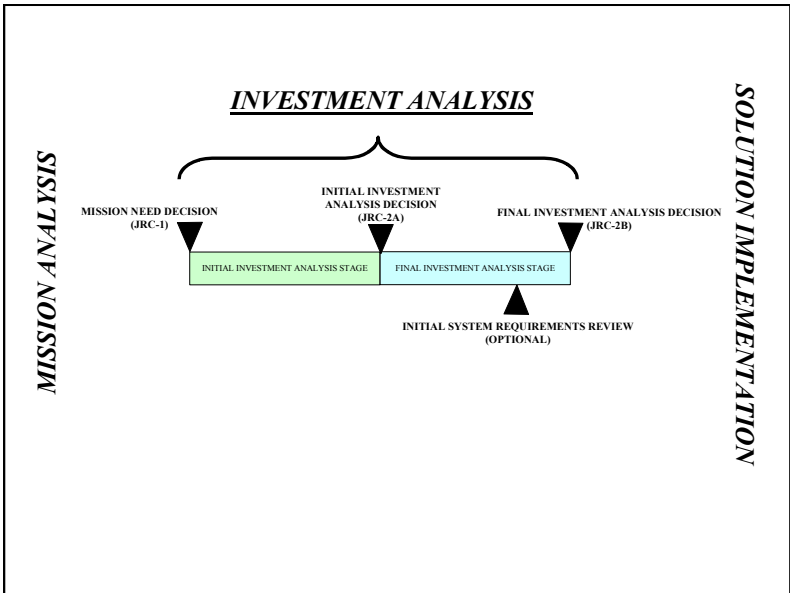



**Figure 3.3-2. Investment Analysis System Engineering Stages**

#### 3.3.2.1 System Engineering Activities

The SE activities occurring during the IA phase appear in Figure 3.3-3. The core SE processes continue, in an iterative fashion, to produce a design that meets the stakeholder need. Functional Analysis (Section 4.4) continues to decompose the functions to lower levels. These lower-level functions are used to develop more detailed requirements that, in turn, are used to bound the next level of functional decomposition. Specialty Engineering (Section 4.8) feeds this process by providing various DARs to further refine the requirements and manage various risk facets. Requirements generated from this interaction are then validated; and once validated, they are fed into the Synthesis process (Section 4.5), where alternative solutions to meet these requirements are developed. The Trade Studies process (Section 4.6) and the Lifecycle Engineering process (Section 4.13) are both heavily employed during this phase to provide Synthesis with the data required to make an informed decision concerning the best solution set. The resulting physical architecture, in conjunction with the functional architecture, is used in Interface Management (Section 4.7) to develop IRDs and eventually Interface Control Documents (ICD).

The primary outputs from the SE efforts in this phase are the functional and physical architectures and associated requirements in the form of IRDs and the fRD. The APB and statement of work (SOW) are developed from external processes to SE but are heavily based on SE input.

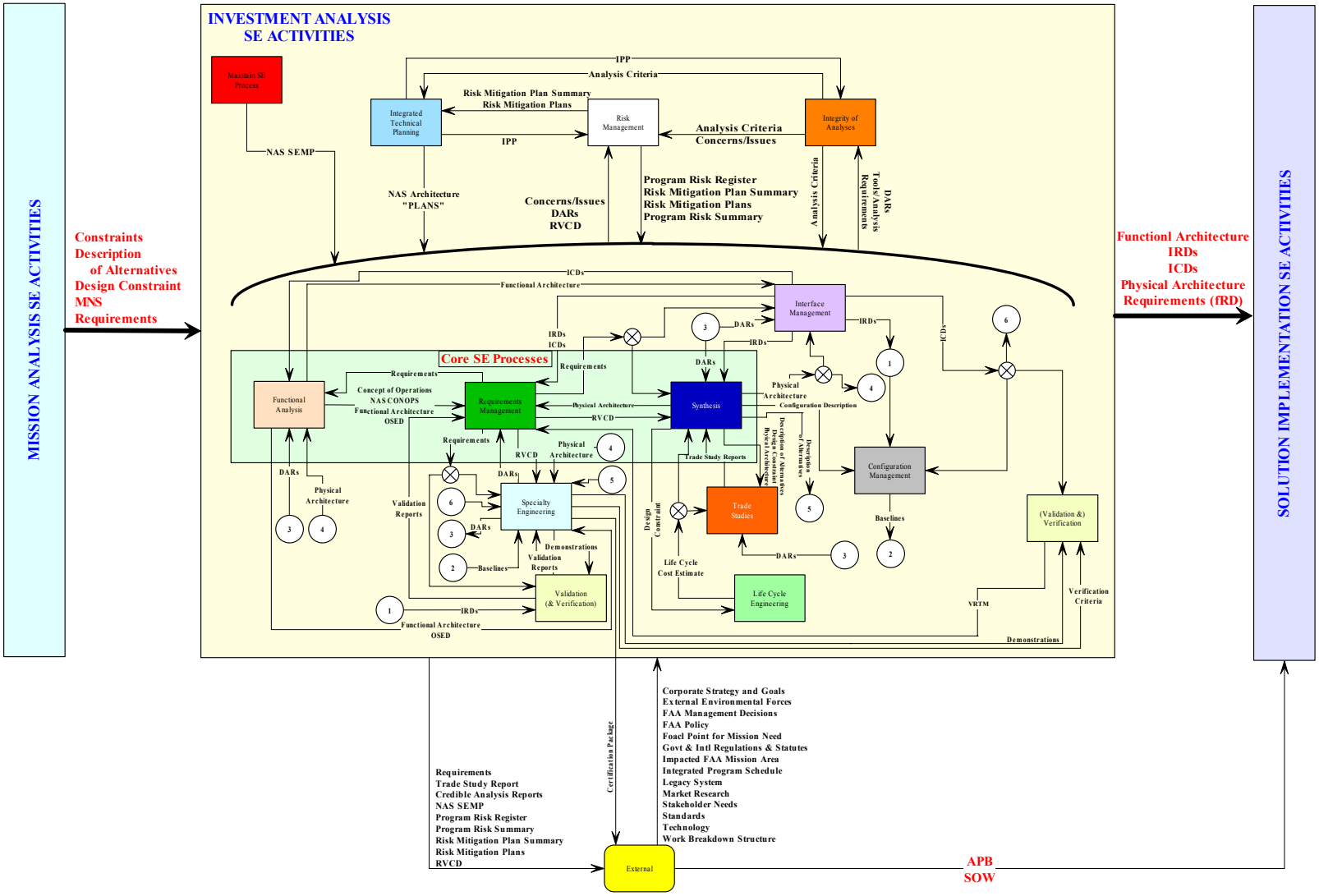



**Figure 3.3-3. Investment Analysis System Engineering Activities**

### 3.3.2.2 Initial Investment Analysis Stage

#### 3.3.2.2.1 Initial Investment Analysis Stage Objectives

Initial IA is the first of two stages in the IA phase. The main objective of this stage is to develop a set of alternative solutions to the requirements and then select the alternative that provides the most balanced solution to the need. To accomplish this objective, SE analyzes the high-level requirements so the needs, objectives, requirements, and operating scenarios are fully understood and integrated. Because these top-level requirements typically lack the details required to execute an end-item design, it is important that stakeholders adequately communicate to eliminate gaps in understanding requirements. To this end, the needs, mission(s), and utilization environments are analyzed, interpreted, and coordinated with stakeholders to determine system requirements. This stage also identifies the required disciplines needed to support the effort as well as a review indicating that all stakeholders have been identified.

In this stage, the system functional architecture is expanded. The functions are then transformed into more detailed system requirements that are resolved in the system physical architectures. Higher-level requirements constrain the next lower functional architecture. Also, the interfaces between the functions, subsystems, and elements that comprise the total system are documented. Functional and performance requirements are allocated to those subsystems and elements. Detailed subsystem and element requirements and constraints are developed, and subsystem and element concepts are traded and selected.

Further development and evaluation of alternative concepts pave the way for selection of the best concept. Each candidate concept is validated to ensure feasibility and that all requirements have been satisfied. Candidate concepts that fail to meet requirements are modified or discarded. More detailed concept development and analyses are then conducted to characterize each of the concepts to add maturity and facilitate selection of the best alternative. Trade Studies (Section 4.6) are conducted to select from alternative approaches to satisfy requirements; identify preferred technologies and processes; define support concepts; assess lifecycle cost elements; and quantify program risks. Down-selection criteria are established based on design sensitivities, cost/benefit ratios, schedules, programmatic constraints and requirements, risks, corporate strategies, and other considerations, as applicable. A single approach shall be selected before the close of this stage, and the details of this baseline are then placed under NAS Configuration Management. The cost/benefit analysis that results in the selection of the best concept is documented and made a part of the program documentation.

Table 3.3-4 summarizes initial IA stage inputs and outputs.

**Table 3.3-4. Initial Investment Analysis Stage System Engineering Inputs and Outputs**

| WORK PRODUCT | PRODUCING SE ELEMENT | INPUT | OUTPUT |
|---|---|---|---|
| Acquisition Program Baseline[1] | EXT | | C |
| Analysis Criteria | IA | I | F |
| Concept of Operations | FA | I | F |
| Concerns/Issues | ALL | SD | SD |
| Constraints | ALL except TS | SD | SD |

| WORK PRODUCT | PRODUCING SE ELEMENT | INPUT | OUTPUT |
|---|---|---|---|
| Corporate Strategy and Goals | EXT | SD | |
| Credible Analysis Results | IA | SD | SD |
| Demonstrations | SpecEng | | SD |
| Description of Alternatives | S | I | F |
| Design Analysis Reports | SpecEng | SD | SD |
| Design Constraint | S | SD | SD |
| External Environmental Forces | EXT | SD | |
| FAA Management Decisions | EXT | SD | |
| FAA Policy | EXT | SD | |
| Functional Architecture | FA | C | I |
| Government and International Regulations and Statutes | EXT | SD | |
| Integrated Lifecycle Plan | ITP | C | I |
| Integrated Program Plan | ITP | C | I |
| Integrated Program Schedule<br>Interface Requirements Document | EXT<br>IM | I | <br>I |
| Legacy System | EXT | SD | |
| Lifecycle Cost Estimate | LC | C | I |
| Market Research | EXT | SD | |
| Master Verification Plan | ITP | | I |
| Mission Need Statement | RM | F | |
| NAS Architecture | ITP | SD | |
| NAS Concept of Operations | FA | SD | |
| NAS System Engineering Management Plan | MSE | SD | |
| Operational Concept Demonstrations | S | | SD |
| Operational Services and Environmental Description | FA | C | I |
| Physical Architecture | S | | C |
| Planning Criteria | ALL except ITP | SD | SD |
| Program Risk Summary | RSK | | SD |
| Requirements[2] | RM | I | I |
| Risk Mitigation Plan Summary | RSK | | SD |
| Risk Mitigation Plans | RSK | | SD |
| Stakeholder Needs | EXT | I | |
| System Engineering Management Plan | ITP | | I |
| Technology | EXT | SD | |
| Tools/Analysis Requirements | ALL except EXT, ITP, IM, IA, CM, Val | | SD |
| Trade Study Report | TS | | SD |
| Validation Reports | Val | SD | SD |
| Verification Criteria | SpecEng | | SD |
| Verification Requirements Traceability Matrix | RM | C | I |

| WORK PRODUCT | PRODUCING SE ELEMENT | INPUT | OUTPUT |
|---|---|---|---|
| Work Breakdown Structure | EXT | C | |
| 1. This work product is not an output of SE per se, but it is identified in order to provide context for the level of SE input to its development.<br>2. The iRD is updated during Initial Investment Analysis.<br><br>NOTE: See Table 3.3-1 for legend. | | | |

#### 3.3.2.2.2 Initial Investment Analysis Stage Entrance Criteria

These criteria include the following:

- An MND approving continuation of the program to the IA phase
- MA output

#### 3.3.2.2.3 Initial Investment Analysis Stage Exit Criteria

These criteria include the following:

- All work products identified as initial IA outputs have been completed to the version level specified
- Required disciplines have been identified
- Trade Studies (Section 4.6) have been planned
- Initial baseline planning has been completed
- The initial IA decision has been made, authorizing the program to proceed to the final IA stage

#### 3.3.2.2.4 System Engineering Element Tasks

In addition to the tasks identified below, each SE Element active during this phase shall surface concerns/issues that present risk to the program, as well as any constraints that bound future Trade Studies (Section 4.6). Furthermore, those active SE Elements that involve analysis shall develop their requirements for analysis (e.g., necessary tools, required analyst levels of competencies).

##### 3.3.2.2.4.1 Tasks for the Integrated Technical Planning Element

In this stage of technical plans development, the following conceptual drafts, developed during MA, are updated to initial drafts:

- IPP
- Integrated Lifecycle Plan

In addition, the SEMP and Master Verification Plan (MVP) are created and developed to an initial draft state by the end of this stage.

#### 3.3.2.2.4.2 Tasks for the Requirements Management Element

The main task of Requirements Management (Section 4.3) is to update the Initial Requirements Document (iRD). Additional activities are focused on capturing the allocation of requirements to the conceptual version of the physical architecture, via a database, and updating the conceptual draft of the VRTM to initial draft.

#### 3.3.2.2.4.3 Tasks for the Functional Analysis Element

The Functional Analysis process (Section 4.4) focuses on finalizing the CONOPS, updating the OSED to an initial draft, and further decomposing the next level of functions into sequenced and traceable functional architectures (dependent upon the availability and detail of requirements documentation).

#### 3.3.2.2.4.4 Tasks for the Synthesis Element

During Synthesis (Section 4.5), conceptual versions of the physical architectures for the set of alternatives are produced and the description of alternatives are further refined. In conjunction with Functional Analysis (Section 4.4) and Trade Studies (Section 4.6), Synthesis (Section 4.5) is the heart of the IA process during this stage. It performs the design analysis of the benefits, strengths, and weaknesses of the alternative concepts against a common set of requirements and selection criteria to determine their relative merits. Down-selection to a preferred solution is the result of this design analysis. Design constraints are identified during this analysis and fed to Lifecycle Engineering (Section 4.13) and Trade Studies (Section 4.6) activities. Operational concept demonstrations are also conducted to support these activities.

#### 3.3.2.2.4.5 Tasks for the Trade Studies Element

Trade Studies (Section 4.6) are conducted at all system levels to ensure performance effectiveness, application of technology and processes, minimum cost, and supportability and affordability, and to generally provide input for the selection of the best concept. Identification of the best concept is based on the criteria developed specifically for the system, its mission, operations concept, performance requirements, programmatic constraints, and other applicable factors. The trade study report documents the findings and is used as data to support and justify the down-selection to one concept.

#### 3.3.2.2.4.6 Tasks for the Interface Management Element

This element begins to identify the external interfaces to the system. The conceptual and then initial draft(s) of the Interface Requirements Document(s) (IRD) are developed during this phase to capture these interfaces. Conceptual drafts of the ICDs are initiated. Functional and physical interfaces internal to the system are also identified and captured via interface control scope sheets.

#### 3.3.2.2.4.7 Tasks for the Specialty Engineering Element

During Specialty Engineering (Section 4.8), tailored analyses are initiated to derive requirements, support trades, and identify risks. The System Safety discipline produces the comparative safety analysis. The system baseline design is established, and the scope, ground rules, and assumptions of analysis are defined. Security shall initiate a risk analysis and vulnerability analysis, document program security level, and commence personnel security

actions. DARs are produced and verification criteria are established. In addition to these activities, demonstrations may be conducted.

#### 3.3.2.2.4.8 Tasks for the Integrity of Analyses Element

The main task for Integrity of Analyses (Section 4.9) is to finalize the analysis criteria, which is accomplished by:

- Identifying analysis needs
- Verifying that the right tools are available
- Verifying that input data is correct
- Verifying that analysts are competent
- Verifying that analytical compatibility rules are documented

Credible analysis results are generated for the latest completed analyses.

#### 3.3.2.2.4.9 Tasks for the Risk Management Element

The identification and analysis of risks begins during Risk Management (Section 4.10) as soon as the stakeholder need(s) is translated into functional requirements. Typical avenues of search include Specialty Engineering disciplines and capabilities (e.g., personnel, facilities, and tools) compared against the concepts, readiness of technology for the functions and concepts that evolve, and likelihood of cost restrictions. At this phase, identified risk areas shall be monitored, and only the nature of mitigation strategies shall be identified. As the program unfolds, the mitigation plans may be finalized. Thus, at this point, Risk Management activities involve:

- Creating a program risk summary
- Creating preliminary risk mitigation plans
- Developing a risk mitigation plan summary

#### 3.3.2.2.4.10 Tasks for the Validation Element

During the Validation process (Section 4.12), validation reports are generated to document the initial results in requirements validation.

#### 3.3.2.2.4.11 Tasks for the Lifecycle Engineering Element

An initial draft of the lifecycle cost estimates that are based on remaining concept alternatives are developed in Lifecycle Engineering (Section 4.13). The existing set of validated requirements, identified design constraints, DARs, and trade study reports are used to drive this estimate. The Integrated Lifecycle Plan is updated to reflect the maturing program planning and captures planning for real estate, deployment and transition, integrated logistics support, sustainment/technology evolution, and disposal.

### 3.3.2.3 Final Investment Analysis Stage

#### 3.3.2.3.1 Final Investment Analysis Stage Objectives

The main objective of this stage, which consists of two parts with an Initial System Requirements Review (ISRR) milestone near the end of the stage, is to establish validated requirements and document the complete functional baseline for the selected solution. This stage further refines the architecture. Selected subsystem and element concepts are expanded with details to verify that they meet high-level requirements and constraints. Also, the interfaces between the elements that comprise the subsystems are documented. Functional and performance requirements and constraints are allocated to those elements, and packages defining development of the elements are created.

A business case is developed that illustrates all stakeholder costs and obligations, providing details of both agency and nonagency resource demands. Program requirements are completed, corrected, and documented in the fRD. Also, the interfaces between the components that comprise the elements are documented, and functional and performance requirements are allocated to those components. The planned procurement specifications are listed, and the APB is completed. A successful IA leads to the JRC-2B decision for the program.

An ISRR is held toward the end of this stage, usually 1 to 2 months prior to JRC-2B, primarily as a means to review and agree upon the final set of system requirements. The fRD is reviewed at this time in preparation for the JRC-2B. Appendix C contains an ISRR checklist to use in preparing this review milestone.

The discussion of SE activities and associated inputs and outputs for the final IA phase is presented below in two parts to account for the ISRR milestone.

#### 3.3.2.3.2 Final Investment Analysis Pre-Initial System Requirements Review

Table 3.3-5 summarizes final IA pre-ISRR inputs and outputs.

**Table 3.3-5. Final Investment Analysis Pre-Initial System Requirements Review Inputs and Outputs**

| WORK PRODUCT | PRODUCING SE ELEMENT | INPUT | OUTPUT |
|---|---|---|---|
| Acquisition Program Baseline[1] | EXT | C | I |
| Analysis Criteria | IA | F | SD |
| Concept of Operations | FA | F | |
| Concerns/Issues | ALL | SD | SD |
| Constraints | ALL except TS | SD | SD |
| Corporate Strategy and Goals | EXT | SD | |
| Credible Analysis Results | IA | SD | SD |
| Demonstrations | SpecEng | SD | SD |
| Description of Alternatives | S | F | |
| Design Analysis Reports | SpecEng | SD | SD |
| Design Constraint | S | SD | SD |
| External Environmental Forces | EXT | SD | |

| WORK PRODUCT | PRODUCING SE ELEMENT | INPUT | OUTPUT |
|---|---|---|---|
| FAA Management Decisions | EXT | SD | |
| FAA Policy | EXT | SD | |
| Functional Architecture[2] | FA | I | F |
| Government and International Regulations and Statutes | EXT | SD | |
| Integrated Lifecycle Plan | ITP | I | F |
| Integrated Program Plan | TP | I | F |
| Integrated Program Schedule | EXT | F | |
| Interface Control Document | IM | | C |
| Interface Requirements Document | IM | I | F |
| Legacy System | EXT | SD | |
| Lifecycle Cost Estimate | LC | I | |
| Market Research | EXT | SD | |
| Master Verification Plan | ITP | I | F |
| NAS Architecture | ITP | SD | |
| NAS Concept of Operations | FA | SD | |
| NAS System Engineering Management Plan | MSE | SD | |
| Operational Concept Demonstrations | S | SD | SD |
| Operational Services and Environmental Description | FA | I | F |
| Physical Architecture | S | C | SD |
| Planning Criteria | ALL except ITP | | SD |
| Program Risk Register | RSK | | SD |
| Program Risk Summary | RSK | SD | SD |
| Requirements[2] | RM | I | F |
| Risk Mitigation Plan Summary | RSK | SD | SD |
| Risk Mitigation Plans | RSK | SD | SD |
| Stakeholder Needs | EXT | F | SD |
| Statement of Work[1] | EXT | | I |
| Standards | EXT | SD | |
| System Engineering Management Plan | ITP | I | F |
| Technology | EXT | SD | |
| Tools/Analysis Requirements | ALL except EXT, ITP, IM, IA, CM, & Val | SD | SD |
| Trade Study Report | TS | SD | SD |
| Validation Reports | Val | SD | SD |
| Verification Criteria | SpecEng | SD | SD |
| Verification Requirements Traceability Matrix[2] | RM | I | F |
| Work Breakdown Structure | EXT | F | |

1. This work product is not an output of SE per se, but it is identified in order to provide context for the level of SE input to its development.

| WORK PRODUCT | PRODUCING SE ELEMENT | INPUT | OUTPUT |
|---|---|---|---|
| 2. This does not imply that there is no further decomposition (e.g., "Final" requirements at this point in time is with respect to the Final Requirements Document (fRD), yet further decomposition takes place to generate a specification).<br><br>NOTE: See Table 3.3-1 for legend. | | | |

#### 3.3.2.3.3 Final Investment Analysis Pre-Initial System Requirements Review Entrance Criteria

These criteria include the following:

- Work products from the initial IA stage have been completed
- Required disciplines have been identified
- Trade Studies (Section 4.6) have been planned
- Initial baseline planning has been completed
- The initial IA decision (JRC 2A) has been made, authorizing the program to proceed to the final IA stage

#### 3.3.2.3.4 Final Investment Analysis Pre-Initial System Requirements Review Exit Criteria

These criteria include the following:

- All work products identified as final IA pre-ISRR outputs have been completed to the version level specified
- The ISSR has been successfully completed if conducted
- All remaining concepts performance requirements have been attained

#### 3.3.2.3.5 System Engineering Element Tasks

As in previous phases of SE efforts—in addition to the tasks identified below— each SE Element active during this phase shall surface concerns/issues that present risk to the program, as well as any constraints that bound future Trade Studies (Section 4.6). Furthermore, those active SE Elements that involve analysis shall develop their requirements for analysis (e.g., necessary tools, required analyst levels of competencies).

##### 3.3.2.3.5.1 Tasks for the Integrated Technical Planning Element

The level of Integrated Technical Planning (Section 4.2) increases during this stage. The following plans are finalized:

- IPP
- MVP
- SEMP

- Integrated Lifecycle Plan

#### 3.3.2.3.5.2 Tasks for the Requirements Management Element

At this stage during Requirements Management (Section 4.3), the fRD is created in preparation for the ISRR, and the final draft of the fRD VRTM is developed for further population during the Verification process (Section 4.12). Additional activities include continued requirements database development and initial specification work.

#### 3.3.2.3.5.3 Tasks for the Functional Analysis Element

Functional Analysis's (Section 4.4) primary task is to move to the next level of functional decomposition to identify the next lower-level functional requirements. Lower-level FFDs are created to document this decomposition. FFDs define the approved sequence of functions that the subsystems perform and lower-levels of the hierarchy to satisfy the overall needs. The functional architecture is a link to the Requirements Management process (Section 4.3) and the Synthesis process (Section 4.5). The functional architecture at this level is documented via FFDs, system behavior diagrams, system timelines, and/or via other Functional Analysis tools and techniques deemed appropriate by the program. Eventually, post ISRR, the completed Functional Analysis for the selected concept is provided as input to the APB document. In addition, the OSED is finalized.

#### 3.3.2.3.5.4 Tasks for the Synthesis Element

The physical architecture of the selected alternative is further decomposed. During this process, design constraints are identified and used to bound further Trade Studies and Lifecycle Engineering efforts.

#### 3.3.2.3.5.5 Tasks for the Trade Studies Element

Trade Studies (Section 4.6) is an organized decisionmaking process to resolve complex questions. As the selected system alternative is further decomposed, Trade Studies shall be conducted to identify the preferred concepts to perform lower-level functions. The results are documented in a trade study report.

#### 3.3.2.3.5.6 Tasks for the Interface Management Element

During Interface Management (Section 4.7), IRDs are finalized, and conceptual drafts of the ICDs are developed. Functional and physical interfaces internal to the system are continually captured via interface control scope sheets as the functional and physical architectures become more detailed.

#### 3.3.2.3.5.7 Tasks for the Specialty Engineering Element

As the concept of the system is further defined, the analyses and demonstrations performed during the Specialty Engineering process (Section 4.8) are used to provide data (i.e., verification criteria) for Synthesis (Section 4.5), Trade Studies (Section 4.6), and Verification (Section 4.12). Specialists examine the proposed solutions to determine which best satisfies the specialty requirements, after which they capture their findings in DARs.

#### 3.3.2.3.5.8 Tasks for the Integrity of Analyses Element

The Integrity of Analyses process (Section 4.9) continues to provide credible analysis results for the latest conducted analyses.

#### 3.3.2.3.5.9 Tasks for the Risk Management Element

Risk Management efforts (Section 4.10) transition to focus on the selected alternative. Risk mitigation plans are developed for all medium- and high-risk items identified for the selected concept. The program risk register is created and updates to the program risk summary, risk mitigation plans, and risk mitigation plan summary are made to filter out risks not associated with the final concept.

#### 3.3.2.3.5.10 Tasks for the Validation and Verification Element

The level of effort for the Validation and Verification process (Section 4.12) is to validate requirements and document validation reports. Verification criteria identified through Specialty Engineering analyses (Section 4.8), along with the existing set of validated requirements, are used to further identify any tools/analysis requirements for eventual Verification activities.

#### 3.3.2.3.5.11 Tasks for the Lifecycle Engineering Element

A final lifecycle cost estimate that is based on the selected alternative is developed in Lifecycle Engineering (Section 4.13). The Integrated Lifecycle Plan is finalized to reflect the maturing program planning and captures planning for real estate, deployment and transition, integrated logistics support, sustainment/technology evolution, and disposal.

### 3.3.2.3.6 Final Investment Analysis Post-Initial System Requirements Review

Table 3.3-6 summarizes final IA Post-ISRR inputs and outputs.

**Table 3.3-6. Final Investment Analysis Post-Initial System Requirements Review Inputs and Outputs**

| WORK PRODUCT | PRODUCING SE ELEMENT | INPUT | OUTPUT |
|---|---|---|---|
| Acquisition Program Baseline* | EXT | I | F |
| Baselines | CM | SD | SD |
| Certification Package | SpecEng | | I |
| Concerns/Issues | ALL | SD | SD |
| Configuration Description | S | | I |
| Constraints | ALL except TS | SD | SD |
| Corporate Strategy and Goals | EXT | SD | |
| Credible Analysis Results | IA | SD | SD |
| Demonstrations | SpecEng | SD | SD |
| Description of Alternatives | S | F | |
| Design Analysis Reports | SpecEng | SD | SD |
| Design Constraint | S | SD | SD |
| External Environmental Forces | EXT | SD | |
| FAA Management Decisions | EXT | SD | |

| WORK PRODUCT | PRODUCING SE ELEMENT | INPUT | OUTPUT |
|---|---|---|---|
| FAA Policy | EXT | SD | |
| Functional Architecture | FA | F | SD |
| Government and International Regulations and Statutes | EXT | SD | |
| Integrated Lifecycle Plan | ITP | F | SD |
| Integrated Program Plan | ITP | F | SD |
| Integrated Program Schedule | EXT | F | |
| Interface Control Document | IM | C | I |
| Interface Requirements Documents | IM | F | SD |
| Legacy System | EXT | SD | |
| Lifecycle Cost Estimate | LC | I | F |
| Market Research | EXT | SD | |
| Master Verification Plan | ITP | F | SD |
| NAS Architecture | ITP | SD | |
| NAS Concept of Operations | FA | SD | |
| NAS System Engineering Management Plan | MSE | SD | |
| Operational Concept Demonstrations | S | SD | |
| Operational Services and Environmental Description | FA | F | |
| Physical Architecture | S | C | I |
| Planning Criteria | ALL except ITP | | SD |
| Program Risk Register | RSK | SD | SD |
| Program Risk Summary | RSK | SD | SD |
| Requirements | RM | F | SD |
| Risk Mitigation Plan Summary | RSK | SD | SD |
| Risk Mitigation Plans | RSK | SD | SD |
| System Engineering Management Plan | ITP | F | |
| Stakeholder Needs | EXT | SD | |
| Standards | EXT | SD | |
| Statement of Work* | EXT | I | F |
| Technology | EXT | SD | |
| Tools/Analysis Requirements | ALL except EXT, ITP, IM, IA, CM, & Val | SD | SD |
| Trade Study Report | TS | SD | SD |
| Validation Reports | Val | SD | SD |
| Verification Criteria | SpecEng | SD | SD |
| Verification Requirements Traceability Matrix | RM | F | SD |
| Work Breakdown Structure | EXT | SD | |

* This work product is not an output of SE per se, but it is identified in order to provide context for the level of SE input to its development.

| WORK PRODUCT | PRODUCING SE ELEMENT | INPUT | OUTPUT |
|---|---|---|---|
| NOTE: See Table 3.3-1 for legend. | | | |

#### 3.3.2.3.7 Final Investment Analysis Post-Initial System Requirements Review Entrance Criteria

These criteria include the following:

- All work products identified as final IA pre-ISRR outputs have been completed to the version level specified
- The ISRR has been successfully completed
- Initial concepts performance requirements have been attained

#### 3.3.2.3.8 Final Investment Analysis Post-Initial System Requirements Review Exit Criteria

These criteria include the following:

- All work products identified as final IA post-ISRR outputs have been completed to the version level specified
- The solution selected at the Initial Investment Analysis Decision is defined via a physical architecture with assurance that it meets all system requirements
- The final IA decision has been made, authorizing the program to continue into the SI phase

#### 3.3.2.3.9 System Engineering Element Tasks

As in previous phases of SE efforts—in addition to the tasks identified below— each SE Element active during this phase shall surface concerns/issues that present risk to the program, as well as any constraints that bound future Trade Studies (Section 4.6). Furthermore, those active SE Elements that involve analysis shall develop their requirements for analysis (e.g., necessary tools, required analyst levels of competencies).

##### 3.3.2.3.9.1 Tasks for the Integrated Technical Planning Element

During Integrated Technical Planning (Section 4.2), plans are maintained and updated as necessary.

##### 3.3.2.3.9.2 Tasks for the Functional Analysis Element

The Functional Analysis process (Section 4.4) continues functional decomposition to the next level of functions and documents via sequenced traceable functional flow diagrams. This resulting functional architecture is the baseline starting point for Functional Analysis activities that will continue during SI.

#### 3.3.2.3.9.3 Tasks for the Requirements Management Element

The Requirements Management process (Section 4.3) is the primary stage during which to update the fRD and VRTM using feedback from the ISRR. Additional activities include continued requirements database development and specification work.

#### 3.3.2.3.9.4 Tasks for the Synthesis Element

The initial draft of the physical architecture for the selected alternative is completed by the end of this stage. Additional design constraints are identified and used to bound further Trade Studies and Lifecycle Engineering efforts, as well as the continued design efforts to take place in the next phase.

#### 3.3.2.3.9.5 Tasks for the Trade Studies Element

The Trade Study process (Section 4.6) continues, as needed, to derive the best solution for the lower-levels of the system.

#### 3.3.2.3.9.6 Tasks for the Interface Management Element

At this stage during Interface Management (Section 4.7), the ICDs are updated to the initial draft. Functional and physical interfaces internal to the system are continually captured via interface control scope sheets as the functional and physical architectures become more detailed.

#### 3.3.2.3.9.7 Tasks for the Specialty Engineering Element

During Specialty Engineering (Section 4.8), additional DARs are developed and used to support the refinement and validation of requirements. Demonstrations may continue to provide data to develop additional verification criteria. The initial draft(s) of certification package(s) for any required certifications are developed.

#### 3.3.2.3.9.8 Tasks for the Integrity of Analyses Element

Credible analysis results are developed and used during Integrity of Analyses (Section 4.9) to ensure that the analyses conducted to support the down-selection decision are viable.

#### 3.3.2.3.9.9 Tasks for the Risk Management Element

Risk Management's (Section 4.10) primary task in this stage is to determine new risks and provide mitigation plans for these risks. This effort supports Synthesis (Section 4.5) and Trade Studies (Section 4.6) efforts. Work products for this element include updates to the program risk register, program risk summary, risk mitigation plans, and risk mitigation plan summary.

#### 3.3.2.3.9.10 Tasks for the Configuration Management Element

Configuration Management's (Section 4.11) primary task during the final IA post ISRR is to bring the baselines, supporting documentation, requirements, and analyses under configuration control.

#### 3.3.2.3.9.11 Tasks for the Validation and Verification Element

The Validation and Verification effort (Section 4.12) is to complete the validation of requirements for the selected alternative and document via validation reports, which are then provided as input to the APB. Verification criteria identified through Specialty Engineering analyses, along with the existing set of validated requirements, are used to further identify any tools/analysis requirements for eventual Verification activities to begin creating the Requirements Verification Compliance Document (RVCD).

#### 3.3.2.3.9.12 Tasks for the Lifecycle Engineering Element

During Lifecycle Engineering (Section 4.13), the lifecycle cost estimate is finalized based on the selected concept. The final set of validated requirements, final DARs, and final trade study reports are used to support this estimate.

### 3.3.3 Solution Implementation Phase

#### 3.3.3.1 Objectives

The SI phase begins with the final IA decision at JRC 2B—an acquisition program is established for the solution selected at JRC 2A—and ends when the new capability goes into service. The SE activities conducted during SI vary widely, depending on the nature and scope of the acquisition program. For example, the activities associated with buying and deploying a commercial product typically are much less complex and time-consuming than those for a product requiring development. However, in each case, products shall be able to meet stakeholder requirements, be operationally suitable, and be compatible with other operational systems before the decision is made to place it in service. The main objective of this phase is to successfully complete the necessary actions and activities to obtain the solution and to accept a product or service for operational use.

#### 3.3.3.2 System Engineering Activities

SE activities required to accomplish the SI objectives appear in Figure 3.3-4. While the SE activities vary widely, depending on the program, the interactions of the SE processes remain essentially the same as in the IA phase. Up front, the activities involve finalizing and baselining the system, its requirements, and the program to support its development and operation. The SE effort then focuses on transforming the accepted concept into a product for deployment. Thus, toward the beginning of the phase, the emphasis remains on the core SE processes, which continue to refine the requirements and bring greater resolution to the design. In the latter portion of this phase, the emphasis shifts to Verification activities (Section 4.12) to verify that the system has been built and integrated according to the requirements.

The final set of SI activities consists of installing the product or service at each site and certifying it for operational use, as appropriate, which typically includes implementation planning, installation and checkout, integration and shakedown, dual operations, and removal and disposal of obsolete equipment.

Various reviews and audits are conducted throughout the SI phase to maintain proper oversight of system development. These reviews and audits, listed below, are defined in the glossary:

- System Requirements Review (SRR)

- System Design Review (SDR)
- Preliminary Design Review (PDR)
- Critical Design Review (CDR)
- Verification Readiness Review (VRR)
- Functional Configuration Audit (FCA)
- Physical Configuration Audit (PCA)

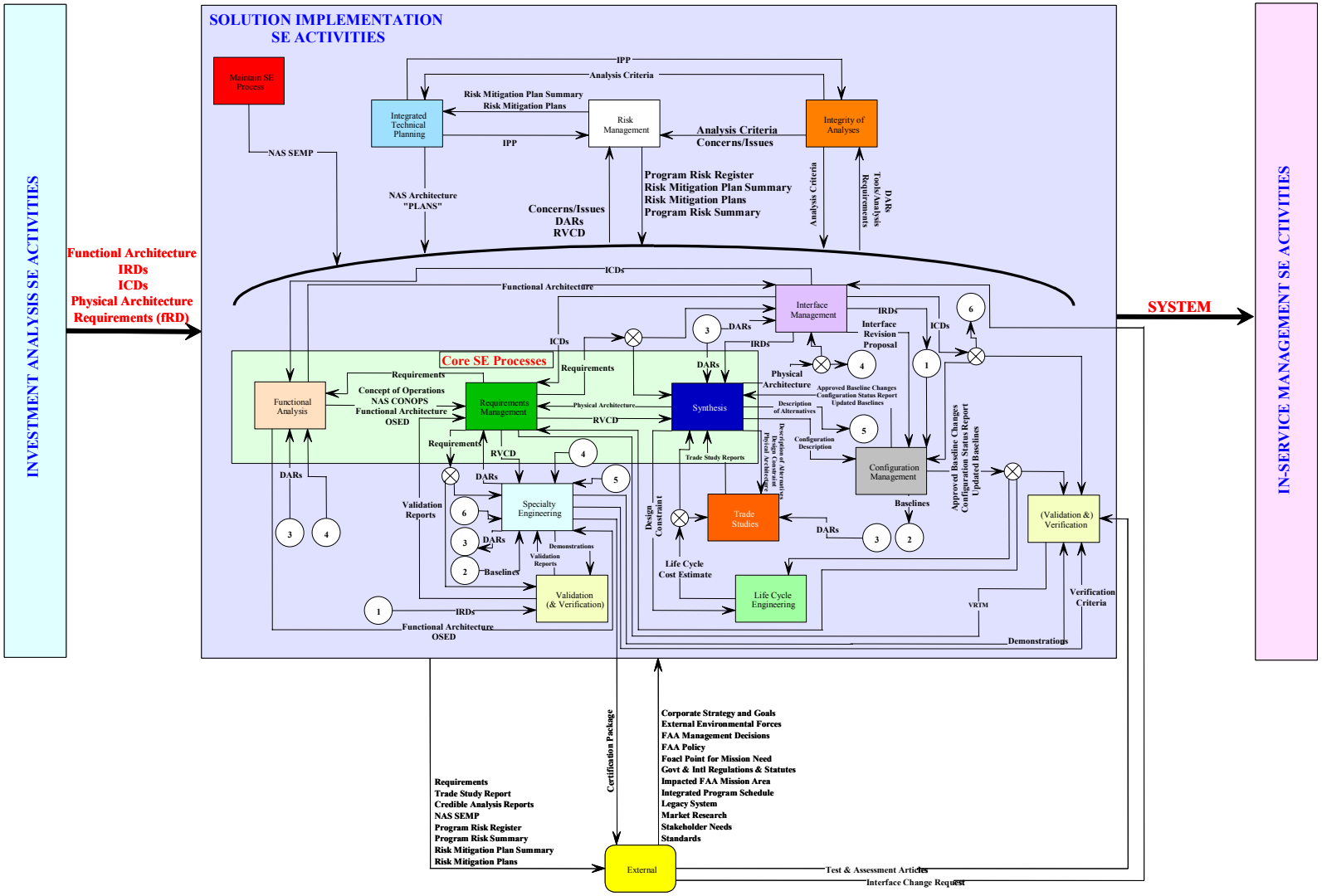



**Figure 3.3-4. Solution Implementation System Engineering Activities**

Table 3.3-7 summarizes the SI inputs and outputs. All products are completed and finalized before completion of the SI phase(i.e., they are not outputs at the end of the phase or a complete result of this phase's SE efforts; rather, they are produced at various points within the SI phase). Each program shall plan at what point in time during this phase that each product is required. For example, final ICDs shall be in place before development and well established for PDR and CDR.

**Table 3.3-7. Solution Implementation System Engineering Inputs and Outputs**

| WORK PRODUCT | PRODUCING SE ELEMENT | INPUT | OUTPUT |
|---|---|---|---|
| Acquisition Program Baseline | EXT | F | |
| Approved Baseline Changes | CM | | SD |
| Baselines | CM | | SD |
| Certification Package | SpecEng | I | F |
| Concerns/Issues | ALL | SD | SD |
| Configuration Description | S | I | F |
| Configuration Status Report | CM | | SD |
| Constraints | ALL except TS | SD | SD |
| Corporate Strategy and Goals | EXT | SD | |
| Credible Analysis Results | IA | SD | SD |
| Demonstrations | SpecEng | SD | SD |
| Design Analysis Reports | SpecEng | SD | SD |
| Design Constraint | S | SD | SD |
| External Environmental Forces | EXT | SD | |
| FAA Management Decisions | EXT | SD | |
| FAA Policy | EXT | SD | |
| Functional Architecture | FA | SD | SD |
| Government and International Regulations and Statutes | EXT | SD | |
| Integrated Lifecycle Plan | ITP | SD | SD |
| Integrated Program Plan | ITP | SD | SD |
| Integrated Program Schedule | EXT | SD | |
| Interface Change Request | EXT | SD | |
| Interface Control Documents | IM | I | F |
| Interface Requirements Document | IM | SD | SD |
| Interface Revision Proposal | IM | | SD |
| Legacy System | EXT | SD | |
| Master Verification Plan | ITP | SD | SD |
| NAS Architecture | ITP | SD | |
| NAS Concept of Operations | FA | SD | |
| NAS System Engineering Management Plan | MSE | SD | |
| Physical Architecture | S | I | F |
| Planning Criteria | ALL except ITP | SD | SD |
| Program Risk Register | RSK | SD | SD |
| Program Risk Summary | RSK | SD | SD |

| WORK PRODUCT | PRODUCING SE ELEMENT | INPUT | OUTPUT |
|---|---|---|---|
| Requirements | RM | SD | SD |
| Risk Mitigation Plan Summary | RSK | SD | SD |
| Risk Mitigation Plans | RSK | SD | SD |
| Requirements Verification Compliance Document | RM, Verification | | F |
| Stakeholder Needs | EXT | SD | |
| Standards | EXT | SD | |
| Statement of Work | EXT | F | |
| Test and Assessment Articles | EXT | F | |
| Tools/Analysis Requirements | ALL except IM, CM, & Val | SD | |
| Trade Study Report | TS | SD | SD |
| Updated Baselines | CM | | SD |
| Validation Reports | Val | SD | SD |
| Verification Criteria | SpecEng | SD | SD |
| Verification Requirements Traceability Matrix | RM, Verification | SD | SD |
| NOTE: See Table 3.3-1 for legend. | | | |

#### 3.3.3.3 Solution Implementation Phase Entrance Criteria

These criteria include the following:

- The final IA decision (JRC 2B) has been made, authorizing the program to continue into SI
- IA outputs have been completed
- Final APB has been completed (via a process external to SE)

#### 3.3.3.4 Solution Implementation Phase Exit Criterion

This criterion is as follows:

- The In-Service Decision has been made, authorizing the program to deploy and put the developed system into service.

#### 3.3.3.5 System Engineering Element Tasks

As in previous stages of SE efforts—in addition to the tasks identified below— each SE Element active during this phase shall surface concerns/issues that present risk to the program, as well as any constraints that bound future Trade Studies (Section 4.6).

##### 3.3.3.5.1 Tasks for the Integrated Technical Planning Element

During Integrated Technical Planning (Section 4.2), plans are maintained and updated as necessary.

#### 3.3.3.5.2 Tasks for Functional Analysis

Functional Analysis (Section 4.4) continues to decompose functions and develop the functional architecture, which continues as long as requirements are developed.

#### 3.3.3.5.3 Tasks for the Requirements Management Element

At the beginning of the SI phase, the Requirements Management process (Section 4.3) is focused on decomposing and finalizing the system specification. As the phase progresses, the requirements effort involves maintaining traceability from high-level requirements to their decomposed lower-level requirements down to actual verification cases and results, and tracking any deviations or waivers to the requirements set as development and verification progress. The final VRTM and RVCD are received as a result of the Verification process with the RVCD being used by Requirements Management to report out to stakeholders.

#### 3.3.3.5.4 Tasks for the Synthesis Element

The physical architecture of the selected alternative is decomposed and finalized during the Synthesis process (Section 4.5) in an iterative mode, as it is employed to design the system's lower-level components and parts being developed. A resulting configuration description is produced and baselined during this phase.

#### 3.3.3.5.5 Tasks for the Trade Studies Element

The Trade Studies process (Section 4.6) continues, as needed, to derive the best solution for the lower-level components and parts of the system.

#### 3.3.3.5.6 Tasks for the Interface Management Element

Interface Management (Section 4.7) is used to finalize the ICDs and process any Interface Change Requests. Updates to finalized ICDs are processed accordingly via interface revision proposals.

#### 3.3.3.5.7 Tasks for the Specialty Engineering Element

The focus of Specialty Engineering (Section 4.8) during this phase is to conduct the various analyses to support the definition of requirements at all levels; support Validation (Section 4.12) of requirements; support the Verification (Section 4.12) of requirements and/or the certification process (e.g., in the areas of safety and information security), as needed; support Trade Studies (Section 4.6); and assist Risk Management (Section 4.10).

#### 3.3.3.5.8 Tasks for the Integrity of Analyses Element

During Integrity of Analyses (Section 4.9), credible analysis results are generated for analyses completed during the SI phase.

#### 3.3.3.5.9 Tasks for the Risk Management Element

The Risk Management process (Section 4.10) continues with risk mitigation plans being put into effect and developing and documenting new plans as new risks are identified.

#### 3.3.3.5.10 Tasks for the Configuration Management Element

Configuration Management's (Section 4.11) primary task during this phase is to maintain configuration control. Changes to the configuration are considered and, if approved, documented via configuration status reports. Baselines are controlled with updates and approved engineering changes are documented and disseminated.

#### 3.3.3.5.11 Tasks for the Validation and Verification Element

Validation (Section 4.12) of requirements continues as the requirements are decomposed through the various system levels. Validation continues as long as requirements (at any level) are developed. Verification efforts (Section 4.12) increase, especially toward the end of the SI phase. As various levels of verification are conducted on test and assessment articles, the final RVCD and the VRTM are populated with results and fed back to the Requirements Management process (Section 4.3).

#### 3.3.3.5.12 Tasks for the Lifecycle Engineering Element

During the SI phase, Lifecycle Engineering elements are engaged in executing the planning that was finalized during the IA phase. This includes acquiring property, facilities, and the physical infrastructure to house the system; detailing the installation and checkout schedule of activities, including site preparation and onsite personnel support; deploying the system; supporting ISR checklist action resolution; training test, operations, and maintenance personnel; ensuring that documentation and spares are procured, verified, and delivered; performing continuous market research activities on any COTS products to project obsolescence situations and mitigate the risks in support of Risk Management (Section4.10); and conducting the disposal activities that are needed for assets that are to be replaced.

### 3.3.4 In-Service Management

In-Service Management involves two distinct sets of work activities. The first set monitors and assesses the real-world performance of the system against its requirements and expected benefits in the APB and takes action to optimize performance throughout its operational life. The second set of activities deals with operating and maintaining the system throughout its service life, as well as maintaining the physical and support infrastructure. The various SE elements are employed within both sets of these activities. Regarding this latter set of activities, the results of SE efforts are used to support the decisionmaking process regarding when a new capability or improvement needs to be in place.

In addition to the timing decision, a decision is made regarding whether modifications or improvements are feasible within approved sustainment funding in the APB. If an engineering change to the system within the sustainment funding cannot be supported, then the shortfall is addressed via the standard AMS lifecycle phases; thus, the SE efforts for this route are as noted in "Mission Analysis Phase" (Paragraph 3.3.1), "Investment Analysis Phase" (Paragraph 3.3.2), and "Solution Implementation Phase" (Paragraph 3.3.3).

If the effort to modify and/or optimize system performance is within the scope of sustaining funds, then the various SE elements are employed much as in the SI phase but on a lesser

scale. The specific SE process and associated level of effort depend on the scope of the upgrade.

If a modification is made to sustain system operations beyond its planned service life, a new investment decision for a service life extension must be requested. Again, the SE efforts during this phase are essentially the same as noted in "Solution Implementation Phase" with respect to the pieces of the system that are being modified to extend the life of the system as a whole.

### 3.3.5 Disposal

SE efforts to support disposal of a system being replaced occur during the new system's SI phase. Lifecycle Engineering (Section 4.12) defines the process for planning and executing disposal activities. The Integrated Technical Planning process (Section 4.2) is used to develop a Disposal Plan in accordance with FAA Order 4800.2, Utilization and Disposal of Excess and Surplus Personal Property.

## 3.4 Reserved

## 3.5 Reserved

## 3.6 Guidance for Tailoring of System Engineering

This section provides guidance regarding the conduct of SE on programs that do not include all engineering disciplines or have unique programmatic demands. The following principles are not intended to be comprehensive, but rather to give general guidelines that may be applied to any part of SE and to programs of any size if rationalization is provided.

### 3.6.1 Basic Principle of Tailoring of System Engineering

The basic principle in program tailoring of SE is that for programs that alter the NAS, reduction in size of an individual element is permissible, but deletion is not. This principle does not mean that large, complex programs may be de-scoped, except under the ground rules listed in this section. The following paragraphs give examples of specific aspects of SE and how they shall be treated in a tailoring effort.

### 3.6.2 Tailoring of Acquisition Management System Process Phase Aspects of System Engineering

"Relationship of the System Engineering Processes to the Acquisition Management System Program Lifecycle" (Section 3.2) describes the AMS phases employed on all programs. These phases shall not be eliminated or combined on any program; however, they may be shorter in duration. Furthermore, the entrance and exit criteria for any phase shall not be ignored. In addition, the exit reviews associated with the phases shall not be eliminated. "Tailoring of Review Aspects of System Engineering" (Paragraph 3.6.5) discusses the reviews.

### 3.6.3 Tailoring of Planning Aspects of System Engineering

All plans pertinent to the program shall be written; however, some plans may be shortened to a single page. The most important plan is the IPP (Integrated Technical Planning (Section 4.2)). The IPP may be reduced to its essential elements, and individual entries may be as short as a single line. The aspects that shall be retained are:

- AMS Phases (Section 3.2)
- SE elements (Sections 4.2 through 4.14)
- SE specialties to be employed on the program

The IPP shall capture information that shows the phase of each SE process, where each engineering specialty is employed, and what work product each process produces. This information is required to produce a forecast of program effort and meaningful schedule.

### 3.6.4 Tailoring of System Engineering Element Aspects of System Engineering

Sections 4.2 through 4.14 describe the 13 SE elements. Each SE process shall be performed on all programs that change the NAS, regardless of the scope of the program, while the nature of the program determines the depth to which each process is performed.

### 3.6.5 Tailoring of Review Aspects of System Engineering

Two rules prevail regarding this topic: (1) all major reviews shall be performed at the end of the phases defined in “Relationship of the System Engineering Processes to the Acquisition Management System Program Lifecycle” (Section 3.2); and (2) reviews shall not be combined. However, a review may be shortened to an hour for a simple project. The moderator of the review shall confirm the basic purpose and ground rules of the review to ensure that they have not been compromised. Software reviews are only required if software is selected as a solution to the system requirements (“Tailoring of Software Aspects of System Engineering” (Paragraph 3.6.10)).

### 3.6.6 Tailoring of Functional Analysis Aspects of System Engineering

The Functional Analysis process (Section 4.4) is an example of a fundamental process whose basic principles shall be maintained on programs of any size. On all programs, Functional Analysis shall be used to derive requirements in a structured and systematic method. The depth, scope, and tools used in the development of the functional architecture may be tailored according to program complexity.

### 3.6.7 Tailoring of Requirements Management Aspects of System Engineering

The Requirements Management process (Section 4.3) is an example of a fundamental process whose basic principles shall be maintained on programs of any size. On all programs, a Requirements Management tool shall be used and the results loaded into the master requirements database.

### 3.6.8 Tailoring of Programmatic Risk Management Aspects of System Engineering

The Risk Management process (Section 4.10) shall be performed on programs of any size and throughout the lifecycle. The example forms provided in Risk Management show that risk to the process is not paper intensive. On the contrary, the Risk Management process presented is extremely practical and adaptable to programs of any size.

### 3.6.9 Tailoring of Verification Aspects of System Engineering

The Verification process (Section 4.12) is one of the SE basic principles—all requirements shall be verified, which is not to say that extensive testing is required, but simply that steps shall be taken to ensure that the solution satisfies the requirements. A simple analysis often provides that assurance. This principle shall not be compromised on small programs. Failure to verify requirements may cause small programs to turn unintentionally into large programs.

### 3.6.10 Tailoring of Software Aspects of System Engineering

Software is a solution to system (i.e., hardware and software) requirements. Hence, if software is not selected as a solution, software reviews and other documentation are not required. If software is required, standard software reviews and documentation are required. However, it shall not be assumed that, if a program is designated as a software program, then the total system aspects of SE may be ignored.

### 3.6.11 Tailoring of Lifecycle Engineering Aspects of System Engineering

Reserved

# 4 PERFORM SYSTEM ENGINEERING

## 4.1 System Engineering

The Federal Aviation Administration (FAA) System Engineering (SE) method is robust, iterative, and has extensive interdependencies among the SE elements listed in Table 1.2-1. The process workflow (see Figure 3.1-1) captures the essence of these linkages. When used properly, SE creates an infrastructure that ensures customer requirements and expectations are effectively and efficiently identified, integrated, and managed. Each SE element is designed to maximize the thoroughness and quality of interaction and cooperation between individuals, teams, suppliers, and stakeholders as each SE element is performed. In addition, each SE element plays various roles throughout the lifecycle phases as shown in Figure 3.2-1. The summary provided below gives an overview of each SE element in terms of objective, definition, and value. Each SE element is extensively documented in the subsequent sections (Sections 4.2 through 4.14), which contain the following details:

- Process-Based Management (PBM) chart (objectives, inputs and associated providing process (providers), outputs and associated receiving process (customers), process tasks, and applicable lifecycle phases)
- Process workflow
- Methods, tools, and detailed descriptions of how each SE element's tasks are accomplished
- Steps to tailor the SE element
- Appendices for terms, acronyms, and work product examples

### 4.1.1 Summary of System Engineering Areas

The following paragraphs briefly summarize FAA SE and its 13 elements. The subsequent sections of the System Engineering Manual (SEM) further detail each element. The brackets following each subsection heading provide a cross-reference to the applicable section number and the relevant integrated Capability Maturity Model (iCMM) process areas.

#### 4.1.1.1 System Engineering

[SEM 4.1; iCMM PA 01 through 04, 07, 08, 11, 13, 16, 20, and 21]

##### 4.1.1.1.1 Objective

The objective of SE within the FAA is to consistently provide balanced solutions to complex FAA system needs.

##### 4.1.1.1.2 Definition

SE defines how the organization discerns a problem, how it approaches solution development to a problem, and how it implements the plan enabling resolution of the problem. It is a discipline that concentrates on the design and application of the whole (system) as distinct from the parts. It involves looking at a problem in its entirety, taking into account all facets and variables and relating the social aspects to the technical aspects.

##### 4.1.1.1.3 Utilization and Value

While SE process elements support the cycle defined by the Acquisition Management System (AMS), they also provide more granularity. This finer, more detailed breakdown provides better management visibility into the operation of the program. Risk is reduced through earlier identification of issues and better identification of requirements. Cost is reduced through earlier recognition and correction of problems. Support organizations are able to gauge and plan their work to support each phase.

#### 4.1.1.2 Integrated Technical Planning

[SEM 4.2; iCMM PA 11]

##### 4.1.1.2.1 Objective

The objective of the Integrated Technical Planning element (Section 4.2) is to provide program management with a sound, repeatable method for the execution of a requirements-based and structurally managed program.

##### 4.1.1.2.2 Definition

The Integrated Technical Planning element provides program management with specific guidance and direction on how to plan a program's execution. The technical plans provide stakeholder- and contract-driven tailoring of SE to optimally satisfy program needs. These plans are living documents that are kept current throughout the program's lifecycle.

##### 4.1.1.2.3 Utilization and Value

Various levels of technical and program management use the technical plans that result from Integrated Technical Planning. Expending upfront effort to generate clear, complete, and correct technical plans results in consistent performance across the program. Optimally, miscommunication and misinterpretation of stakeholder and executive expectations by individuals are eliminated. Developing and following properly prepared plans assists in eliminating miscommunication and helps the program to adapt to changes in program environment.

#### 4.1.1.3 Requirements Management

[SEM 4.3; iCMM PA 01 and 02]

##### 4.1.1.3.1 Objective

The objective of the Requirements Management element (Section 4.3) is to identify and develop all requirements and ensure that they are met throughout the product's lifecycle.

##### 4.1.1.3.2 Definition

The Requirements Management element is a series of iterative tasks performed by a multifunction team throughout all AMS phases. The team's focus is to elicit, develop, manage, and control requirements and associated documentation. Once requirements are defined, the team uses a disciplined Requirements Management methodology to manage the requirements

through verification, helping to ensure compliance with stakeholder needs and expectations, communication of allocations, and adaptation to/control of changes.

#### 4.1.1.3.3 Utilization and Value

Requirements are the fuel for the design process. They define the needed characteristics of a system at all levels of complexity. They are derived from multiple inputs from internal and external sources that need to be logically and efficiently collected and synthesized in a centralized, accessible decision database(s). The information collected, managed, and controlled is accessed by various teams within the stakeholder and program organizations, associated internal interfaces (e.g., management or operations), and contractors/suppliers. When Requirements Management is performed well, rework and poorly communicated information typically is minimal, if not eliminated entirely. Furthermore, this process is used to surface gaps, redundancies, biases, and/or inconsistencies and resolve, revise, and/or refine them in a consistent, integrated method to the satisfaction and agreement of all the stakeholders. The solid foundation built through Requirements Management provides an ongoing resource for all program stages.

### 4.1.1.4 Functional Analysis

[SEM 4.4; iCMM PA 03]

#### 4.1.1.4.1 Objective

The objective of the Functional Analysis element (Section 4.4) is to provide a framework for requirements that significantly improves innovation, synthesis, and product integration.

#### 4.1.1.4.2 Definition

The Functional Analysis element takes the stakeholders' needs and translates them into a sequenced and traceable functional architecture. The system is represented as a set of functions defined as tasks, actions, or activities that are performed to achieve specified sequenced and time-based behaviors. Functions are described as what needs to be done, not how. Therefore, each function is written in the verb-noun form (e.g., “read book” and “cook food”). The functions are accomplished by one or more elements, including equipment/hardware, software, firmware, facilities, personnel, and/or procedural data. Each function is hierarchically decomposed until the basic subfunction is reached, and the requirements are fully developed. The functional architecture defines what the system does, including interfaces (both within the system and to the external world).

#### 4.1.1.4.3 Utilization and Value

A logically sequenced and thoroughly functional architecture is critical to the definition of requirements. It surfaces innovative design solutions and sheds light on vague interfaces. It also provides the basis for logical and realistic product integration and synthesis. As the analyses are performed, additional requirements often are flushed out/derived, thereby providing the program with a more detailed list of requirements and an increased understanding of the system. The functional architecture and functional interfaces enable the stakeholders and program management to logically develop requirements down to the lowest level of a system hierarchy.

#### 4.1.1.5 Synthesis

[SEM 4.5; iCMM PA 04]

##### 4.1.1.5.1 Objective

The objective of the Synthesis element (Section 4.5) is to develop (synthesize) balanced solutions to requirements.

##### 4.1.1.5.2 Definition

The Synthesis element develops solutions to problems (as defined by the requirements). This SE element uses scientific/engineering knowledge and methods to derive and document the hows used to solve the whats that are reflected in the requirements. The synthesized design generated is a balanced (i.e., cost, quality, schedule, risk, performance, producible/supportable) solution. The synthesized design is created through the analysis of candidate elements. The candidate elements are preliminarily defined and then iterated until the refinement of the system concept is complete. The final outputs, which also show relationships between candidate elements, are distributed to the groups responsible for building various system elements.

##### 4.1.1.5.3 Utilization and Value

A series of benchmarks for various design performance parameters (e.g., power, data storage, testability, reliability) are generated and used to measure the viability and worth of a candidate design solution. Design performance parameters, ranked by importance, are refined during the design evolution of an affordable, responsive system design. Throughout the evolutionary analyses, credibility and acceptability by the stakeholders shall be ensured. The iterative nature of the candidate element task provides the mechanism for continuous correction of design inadequacies and refinement of the physical allocation process. It also surfaces opportunities for new technologies and innovative ideas to be considered, justified, and integrated. These efforts are used to validate the synthesized design in terms of balance, completeness, understandability, and reflection of the stakeholders' requirements.

#### 4.1.1.6 Trade Studies

[SEM 4.6; iCMM PA 04]

##### 4.1.1.6.1 Objective

The objective of the Trade Studies element (Section 4.6) is to select balanced (i.e., cost, schedule, quality, and risk) solutions from a set of available alternatives based on defined criteria.

##### 4.1.1.6.2 Definition

The Trade Studies element is used by multidisciplinary teams to confirm that the most balanced technical solutions have been identified. The team methodically evaluates a series of design alternatives and recommends the preferred feasible solutions that enhance the value and performance of the overall system and/or functions. The primary assessment methods are the +/- method, the weighted value method, and the cost assessment method. Each assessment is taken to an appropriate level of detail that allows differentiation between alternatives.

Recommendations are assembled in a trade study report and forwarded to the appropriate decisionmaker(s) (e.g., program management or stakeholders) for action.

#### 4.1.1.6.3 Utilization and Value

The tasks within the Trade Studies element are designed to assist decisionmakers. The thorough identification and assessment of multiple facets of a problem aids the decisionmaker to relate the whole problem to optimal, feasible solutions by comparing technical, cost, and schedule interactions. The appropriate authority uses this information to make a final decision. The Trade Studies process provides the traceability needed to substantiate design and configuration changes to the baseline product design; it also documents why one alternative was chosen over another during the decisionmaking process.

### 4.1.1.7 Interface Management

[SEM 4.7; iCMM PA 07]

#### 4.1.1.7.1 Objective

The objective of the Interface Management element (Section 4.7) is to achieve functional and physical compatibility between all interrelated system elements.

#### 4.1.1.7.2 Definition

An interface is any boundary between one area and another. It may be external, internal, functional, or physical. Interfaces occur within the system (internal) as well as between the instant system and another system (external) and may be functional or physical (e.g., mechanical, electrical) in nature. Interface requirements are documented in an Interface Requirements Document (IRD). The Interface Control Document (ICD) contains the final details of how the contractor implements the requirements. An Interface Control Plan describes the management process for IRDs and ICDs. This plan provides the means to identify and resolve interface incompatibilities and to determine the impact of interface design changes.

#### 4.1.1.7.3 Utilization and Value

During the program's life, compatibility and accessibility shall be maintained for the many diverse elements. Compatibility analysis of the interface definition demonstrates completeness of the interface and traceability records (or lack thereof). As changes are made, an authoritative means of controlling the design of interfaces shall be managed with appropriate documentation, thereby avoiding the situation in which hardware/software, when integrated into the system, fails to function as part of the system, as intended. Ensuring that all system pieces work together is a complex task that involves teams, stakeholders, contractors, and program management from the end of the initial concept definition stage through the operations and support stage.

### 4.1.1.8 Specialty Engineering

[SEM 4.8; iCMM PA N/A]

#### 4.1.1.8.1 Objective

The objective of the Specialty Engineering element (Section 4.8) is twofold: (1) to integrate specific system attributes and disciplines into the acquisition process; and (2) to assess and confirm various system attributes (Specialty Engineering).

#### 4.1.1.8.2 Definition

The Specialty Engineering element includes System Safety Engineering (SSE); Reliability, Maintainability, and Availability (RMA); Human Engineering (human factors); Electromagnetic Environmental Effects ($E^3$); Quality Engineering; Information Security Engineering; and Hazardous Materials Management/Environmental Engineering. Specialty Engineering analyses describe technical details of the design from a particular perspective and often require specialized skills. These analyses help the program to define requirements and design features and/or describe characteristics of the design and related operations in support of Validation and Verification (Section 4.12), requirements, Trade Studies (Section 4.6), Synthesis (Section 4.5), and Functional Analysis (Section 4.4). These analyses are performed throughout the product's lifecycle. At minimum, analysis results shall be available at standard design milestones, including the preliminary and critical design reviews. Table 4.1-1 provides a general description of the specialty engineering disciplines.

**Table 4.1-1. Specialty Engineering Disciplines**

| Specialty Engineering Discipline | Description |
|---|---|
| SSE | Evaluation and management of the safety risk associated with a system using measures of safety risk identified in various hazard analyses, fault tree analyses, safety risk assessments, and hazard tracking and control. |
| RMA | Quantitative and qualitative analyses of the attributes of the system to perform reliably. Quantitative assessments are in the form of probabilistic, mean, and/or distribution assessments. Qualitative analyses are in the form of failure mode assessments. Evaluation of the design's ability to meet operational readiness requirements through preventive and corrective maintenance. |
| Human Factors Engineering | Human factors is a multidisciplinary effort to generate and compile information about human capabilities and limitations and apply that information to:<br>– equipment, systems, facilities<br>– procedures, jobs, environments<br>– staffing<br>– training<br>– personnel and organizational management<br><br>for safe, comfortable, and effective human performance. |

| Specialty Engineering Discipline | Description |
|---|---|
| $E^3$ | Analysis of the system for susceptibility and/or vulnerability to electromagnetic fields or capability to generate such fields that might interfere with other systems, identify sources of interference, and means for correction within the levels prescribed by law, program requirements, spectrum management, or recognized standards.<br>$E^3$ is composed of Electromagnetic Interference (EMI) and Electromagnetic Compatibility (EMC) |
| Quality Engineering | Evaluation of a system's ability to meet its requirements and to mitigate product defects. |
| Information Security Engineering | Evaluation of the vulnerability of the system to unauthorized access and use, or susceptibility to sabotage. Assessment of the ability of the system to survive a security threat in the expected operational environment. |
| Hazardous Materials Management/Environmental Engineering | Determination of environmental impacts at deployment sites and during operations, including both environmental impacts on the system and system impacts on the environment during all phases of the product life. |

#### 4.1.1.8.3 Utilization and Value

These analyses are used to support functional analysis, define and allocate requirements, contribute to the design, and to evaluate design progress, technical soundness, and risk. They are also needed by the stakeholders to ensure that the product performs as intended, as well as by engineering, operations, and product support personnel to accomplish their responsibilities in product development and operation.

### 4.1.1.9 Integrity of Analyses

[SEM 4.9; iCMM PA N/A]

#### 4.1.1.9.1 Objective

The objective of the Integrity of Analyses element (Section 4.9) is to ensure that analyses provide the required level of fidelity and accuracy in a timely manner.

#### 4.1.1.9.2 Definition

Throughout SE and the program's lifecycle, analyses are constantly being performed. These analyses range from simple to complex, quantitative to qualitative, top-down to bottom-up, and basic formulas to sophisticated simulations. In order to ensure credible, useful, and sufficient data/results for program management's decisionmaking process, the integrity and fidelity of the various analysis tools shall be understood and validated. This validation takes several forms: the attributes of the tool suite, validity of the input data, and proficiency and workmanship of the

analyst. An Analysis Management Plan is generated that outlines the details of the various analysis methods and tools. It is recommended that this plan also reflect the program's constraints in terms of technical capabilities, schedule requirements, and cost requirements.

#### 4.1.1.9.3 Utilization and Value

The initial selection of the method, tools, or model to be used in an analysis focuses on determining a practical tool that provides the most visibility into the problem with the least complexity. Because this process is iterative, there is an ongoing need to use the best approach to select the right method, tool, or model, considering the preferences of the stakeholders, other teams' previous experience with different tools, and the limitations of budgets, technology, and schedule. The bottom line is to have analyses in place that guard against mistakes and embed a consistent level of confidence in the integrity of the analysis. The analysis, in turn, contributes significantly to the success of the decisionmaking processes of program management, teams, stakeholders, and contract managers.

### 4.1.1.10 Risk Management

[SEM 4.10; iCMM PA 13]

#### 4.1.1.10.1 Objective

The objective of the Risk Management element (Section 4.10) is to identify and analyze the uncertainties of achieving program objectives and develop plans to reduce the likelihood and/or consequences of those uncertainties.

#### 4.1.1.10.2 Definition

The Risk Management element is an organized, systematic decisionmaking SE process element used by all disciplines and program teams to identify risks regarding achieving program goals, analyze these risks, and effectively mitigate these risks. Risk is defined as an event or situation with a realistic uncertainty of occurrence and an unfavorable consequence if the risk occurs. Risk Management is applied at all levels, from small projects to large programs. Risks are identified (what might go wrong), impacts are analyzed (how big is the risk), mitigation plans are defined (how to reduce the risk), and risk status is continuously tracked and monitored (how the mitigation efforts are progressing). Identifying risks begins when a program is initiated and continues throughout the program's life. A risk watchlist, which compiles the most significant risk items into a single composite document, is generated. The watchlists are used to continuously monitor and track the overall risk status within team meetings and program management reviews.

#### 4.1.1.10.3 Utilization and Value

Understanding the levels of likelihood and consequences of risk occurring increases the program manager's and program team's ability to anticipate and control the impacts of internal and/or external events on their programs. These impacts include, but are not limited to, cost, quality, schedule, and stakeholder satisfaction trends. The comprehensiveness of the analysis drives the thoroughness of what resources are required to mitigate the risk (e.g., budgets, requirements changes, stakeholder interfaces). Risk identification worksheets, tools, and terminology ensure a consistent approach that generates an analysis in which subjectivity is minimized, and confidence in the analysis is maximized.

#### 4.1.1.11 Configuration Management

[SEM 4.11; iCMM PA 16]

##### 4.1.1.11.1 Objective

The objective of the Configuration Management element (Section 4.11) is to establish and maintain consistency of a product's performance, functional, and physical attributes with its requirements, design, and operational information throughout its life.

##### 4.1.1.11.2 Definition

The Configuration Management element is an orderly identification, documentation, and maintenance of a product's functional performance and physical attributes. The tasks are focused on consistency of requirements, design, and operational information throughout the product's life. Once baselined as defined by stakeholder requirements, changes are systematically approved and managed to ensure that traceability/accountability is maintained throughout myriad levels of documentation. The scope of this process element begins with planning the Configuration Management process for the context and environment in which it is to be performed. It ends when the configuration and its associated changes are verified and audited for accuracy and completeness. Throughout the entire Configuration Management effort, a status check provides accurate and timely information concerning the product and its associated data. Support tasks within Configuration Management include developing training plans, defining performance-based management measurements, and assessing methods and trends to effect process improvements.

##### 4.1.1.11.3 Utilization and Value

Configuration Management benefits the program, stakeholders, and contractors/suppliers. As product attributes are defined, measurable performance parameters may be established for the product's acquisition and use. As changes are made, Configuration Management provides correct and current information to the decisionmaking process. When configurations are managed, product repeatability is enhanced, guesswork and downstream surprises are avoided, cost and schedule savings are realized, erratic changes are minimized, proper replacement and repairs are ensured, and maintenance costs are reduced. The overall effect is the establishment of a high level of confidence in the product information.

#### 4.1.1.12 Validation and Verification

[SEM 4.12; iCMM PA 08]

##### 4.1.1.12.1 Objective

The objective of the Validation and Verification element (Section 4.12) is to determine that the system and process requirements are correct and have been met.

##### 4.1.1.12.2 Definition

The Validation and Verification element ensures that all system requirements are correct and have been met. The Validation process proves requirements are correct. The Verification process proves that requirements are met. Requirements may not be verified by test alone.

The majority of requirements are verified by a combination of test and assessment, which comprise the two categories of verification. Test is the disciplined and controlled subjection of the system to conditions that replicate operations in a real or simulated environment as defined by the requirements. It involves examination, observation, and evaluation of measurable parameters of a system element. Assessment, the second category of verification, includes analysis, demonstration, inspection, verification by similarity, validation of records, simulation, and review of design documentation. It is a basic principle of SE that all requirements shall be verified.

**4.1.1.12.3 Utilization and Value**

The Validation process eliminates poor or unnecessary requirements, ensuring that the FAA obtains a set of requirements that are necessary and sufficient to meet the needs of the stakeholders. The Verification process ensures that the product satisfies the validated requirements, and thus meets the stakeholders' needs.

**4.1.1.13 Lifecycle Engineering**

[SEM 4.13; iCMM PA N/A]

**4.1.1.13.1 Objective**

The objective of the Lifecycle Engineering element (Section 4.13) is to assess and confirm system attributes (Lifecycle Engineering).

**4.1.1.13.2 Definition**

The Lifecycle Engineering analyses supplement the program to define requirements and design features or describe characteristics of the design and related operations. These analyses provide technical details of the design from a particular perspective and are performed throughout the product's lifecycle. At minimum, analysis results shall be available at standard design milestones, including the preliminary and critical design reviews.

**4.1.1.13.3 Utilization and Value**

These analyses are used to evaluate design progress, technical soundness, and risk. They are also needed by the stakeholders to ensure that the product performs as intended, as well as by engineering, operations, and product support personnel to accomplish their responsibilities in product development and operation.

**4.1.1.14 System Engineering Process Management**

[SEM 4.14; iCMM PA 20 & 21]

**4.1.1.14.1 Objective**

The System Engineering Process Management element (Section 4.14) has two objectives. The first is to manage and maintain the SE processes in order to satisfy the FAA's goals. This objective is accomplished by maintaining technical awareness, inserting new technology into SE, maintaining the SE support environment, and monitoring the SE support environment for improvement opportunities. The second is to gain agencywide skill and process consistency by

continuously improving the effectiveness and efficiency of the SE processes. This objective is accomplished by analyzing the processes and explicitly planning and deploying improvements to those processes.

#### 4.1.1.14.2 Definition

System Engineering Process Management provides support and balance for the 12 other SE process elements. It also covers activities to measure and improve the SE process elements, which involves designing, developing, improving, and maintaining definitions of SE activities, work, products, methods, techniques, practices, and tools. It additionally provides the technology environment needed to develop systems and perform SE.

#### 4.1.1.14.3 Utilization and Value

This process provides the details and data required to ensure and improve the efficiency and effectiveness of overall SE. In turn, the purpose of improved SE is to reduce cost and schedule, while improving the efficiency and safety of the National Airspace System.

## 4.2 Integrated Technical Planning (Satisfies Criteria of EIA/IS731 FA 2.1 & iCMM PA 11)

### 4.2.1 Introduction to Integrated Technical Planning

Integrated Technical Planning is the tactical and strategic means of defining problems, forecasting conditions, and coordinating program elements to maximize program focus on providing superior products and services. The Integrated Technical Planning process provides the technical guidance tools required to track and manage program activity, as well as the program-specific process tailoring required to optimally satisfy program needs. This System Engineering (SE) element has been subdivided into two primary areas: plans and reviews. The plans include the Integrated Program Plan (IPP) and supporting technical plans such as the Master Verification Plan (MVP), the System Engineering Management Plan (SEMP), the System Safety Management Plan etc. The review section contains both design reviews and audits. This section includes all planning documents; specific development details are in Appendix E. Tailoring may be performed by deleting planning requirements, with rationale provided for each deletion. The only allowable additions are those unique to the program and formally required by the stakeholders. The size, complexity and visibility of a program will determine which SE elements need to be supported by more detailed planning documents. Integrated Technical Planning applies to all programs/projects regardless of size, whether or not they are new programs or changed or derivative projects. The size and scope of planning may change to meet program needs. A change to a program with an existing IPP, SEMP, or other plans only requires documentation that existing plans still apply. On any existing program, the current plans should be referenced in all new plans developed.

#### 4.2.1.1 Integrated Technical Planning Objective

The objective of the Integrated Technical Planning process is to provide program management with a sound, repeatable method for executing requirements-based and structurally managed programs.

#### 4.2.1.2 Process-Based Management

The Process-Based Management (PBM) chart appears in Figure 4.2-1.

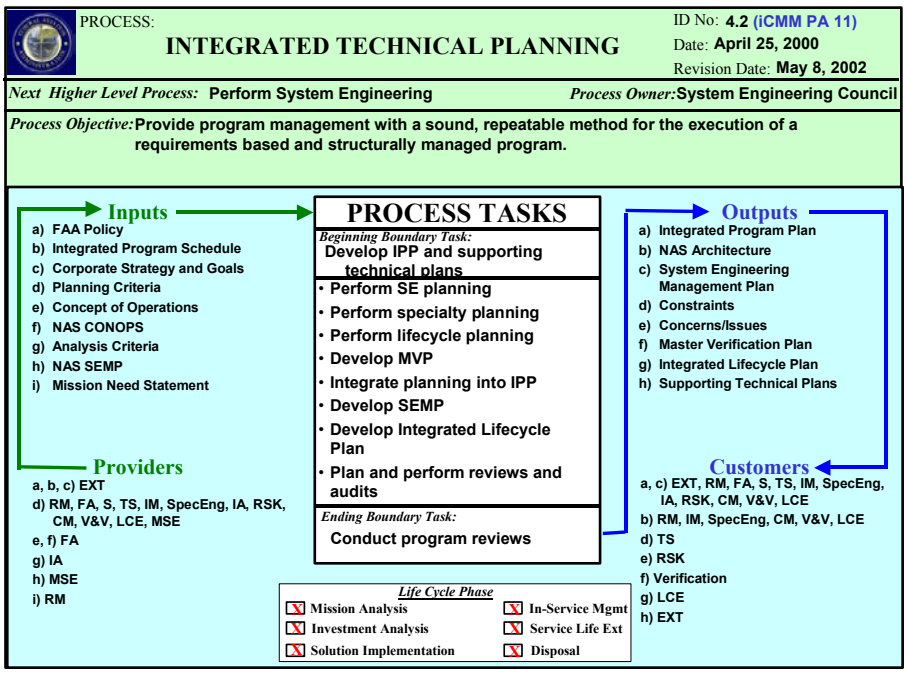



**Figure 4.2-1. Integrated Technical Planning Process-Based Management Chart**

### 4.2.1.3 Inputs to Integrated Technical Planning

The inputs to the process at this level appear in the PBM chart. Some of these inputs provide requirements, while others impose constraints.

### 4.2.1.4 Integrated Technical Planning Process Tasks

The process tasks are shown on the PBM.

### 4.2.1.5 Outputs of Integrated Technical Planning

The output from this process constitutes most of the "Manage To" package, as well as a part of the "Design To" package. A summary of the output for this process is shown on the PBM chart above. Details of the outputs are documented later in this chapter.

### 4.2.1.6 Integrated Technical Planning Process Metrics

The metrics for performance of the Integrated Technical Planning process are the sum of the metrics on lower-level processes.

### 4.2.1.7 Integrated Technical Planning Tools

Integrated Technical Planning requires word processing, display, and scheduling tools.

#### 4.2.1.8 Key Decisions

Key decisions required for this process are:

- Request by the stakeholder and/or program manager for Integrated Technical Planning (usually included in the IPP and SEMP)
- Identification of necessary planning elements by the program system engineer and the project team
- Program manager acceptance that the identified planning elements are necessary
- Baseline plan accepted by the program manager, stakeholders, and the Joint Resources Council (JRC)
- Program manager's approval of the IPP, MVP, SEMP and any other supporting technical plans

#### 4.2.1.9 Key Process Interfaces

Integrated Technical Planning interfaces with all other SE processes, either receiving inputs from them or providing outputs to them.

#### 4.2.1.10 Acquisition Management System Process Interface

#### 4.2.1.11 The Acquisition Management System (AMS) process interface is described in Chapter 3. AMS process activities that most strongly interact with the SE must be taken into account in the Integrated Technical Planning process. Plan Review

All plans are living documents and are subject to continuous review and update to satisfy program needs and changes. All available plans should be reviewed at each AMS milestone and as part of subsequent system baseline modifications throughout the program lifecycle.

### 4.2.2 Integrated Program Plan

#### 4.2.2.1 Introduction to the Integrated Program Plan

The IPP is the primary document within the AMS for planning the actions and activities needed to execute the program within the cost schedule, benefits, and performance baselines in the approved Acquisition Program Baseline (APB). An approved IPP is required for the Final Investment Decision (JRC-2B). IPP development is initiated when the Mission Needs Review (MNR) has been completed and when the Concept of Operations (CONOPS) has been developed. The IPP is reviewed and updated at all subsequent phase exit reviews and reflects changes throughout the program's lifecycle. The IPP reflects contractual requirements and unique programmatic requirements.

The IPP is the recognized plan used to manage a project and contains the Integrated Program Schedule, which encompasses milestones (events), accomplishments, and criteria. The IPP relates accomplishments to program events and demonstrates a logical, event-driven sequence of effort. It is directly traceable to the Work Breakdown Structure (WBS) and Statement of Work (SOW). The IPP provides vertical and horizontal integration traceability through its task statements and numbering system and identifies task relationships. It facilitates resource planning and provides time-phased tasks, a tool for

measuring progress against planned efforts, problem identification, and a framework to develop recovery and workaround plans. Table 4.2-1 lists the sections of an IPP.

**Table 4.2-1. Integrated Program Plan Table of Contents**

| Integrated Program Plan Table of Contents | |
|---|---|
| 1 | BACKGROUND |
| 1.1 | Mission Need |
| 1.2 | Status |
| 2 | OVERVIEW |
| 2.1 | Program Scope |
| 2.2 | Products |
| 3 | INTEGRATED PROGRAM FUNDING |
| 4 | INTEGRATED PROGRAM SCHEDULE |
| 5 | PERFORMANCE |
| 5.1 | Core Work Activities |
| 5.2 | Program Management Work Activities |
| 5.3 | Procurement Work Activities |
| 6 | BENEFITS |
| 7 | PHYSICAL INTEGRATION |
| 8 | FUNCTIONAL INTEGRATION |
| 9 | HUMAN INTEGRATION - RESERVED |
| 10 | SECURITY - RESERVED |
| 11 | IN-SERVICE SUPPORT |
| 12 | VERIFICATION (INCLUDES TEST AND EVALUATION) |
| 13 | IMPLEMENTATION AND TRANSITION |
| 14 | QUALITY ASSURANCE |
| 15 | CONFIGURATION MANAGEMENT |
| 16 | IN-SERVICE MANAGEMENT |

#### 4.2.2.2 Inputs to the Integrated Program Plan

The following inputs are necessary to develop the IPP:

- Program objective as reflected in the stakeholder-provided, top-level Mission Need Statement (MNS) and requirements documents, which detail the operational environments in which the system is expected to operate
- Program-specific guidelines
- Top-level program constraints and assumptions, including program-specific organizational constraints and assumptions to be used on the program

- Program-specific schedule constraints and events
- Concept approach, including top-level conceptual alternatives, functional analyses, design support alternatives, and initial system evaluations
- Any specified government or external standards to be employed on the program
- Any other supporting technical plans (e.g., MVP, SEMP) to be presented at the JRC-2B, either in a draft or baseline stage

#### 4.2.2.3 Integrated Program Plan Steps

An IPP is normally coordinated and written by SE, using the following steps.

##### 4.2.2.3.1 Step 1: Collect Inputs

All program elements, both technical and nontechnical, are responsible for providing IPP inputs. The stakeholders provide the inputs identified in Paragraph 4.2.2.1.1 for every technical and nontechnical discipline involved. Inputs are also gathered from the RFP, SOW, WBS, organizational charts, Contract Data Requirements List (CDRL and schedule information.

##### 4.2.2.3.2 Step 2: Prepare Integrated Program Plan

The IPP is prepared in accordance with the format described within the AMS. IPP tools must be selected and a timetable for implementation prepared. The draft IPP includes accomplishments and criteria for each event, responsibility for each accomplishment, entrance and exit criteria, milestone linkages, and supporting narratives. “System Engineering in the Acquisition Management System Program Lifecycle” (Chapter 3) provides some guidelines on the timing for developing various IPP drafts, with the final approved IPP required for the Final Investment Decision (JRC-2B). The AMS FAST Toolset contains the IPP template.

##### 4.2.2.3.3 Step 3: Coordinate and Baseline

The internal and external IPP stakeholders are provided drafts of the IPP for review. Once concurrence is obtained from the stakeholders, the IPP is approved at the JRC-2B and becomes the baseline IPP. SE coordinates IPP impacts and develops workaround strategies.

##### 4.2.2.3.4 Step 4: Maintain Plan

The program progress is monitored continually throughout the life of the program. Changes in the program are reflected in the IPP, which is then coordinated for approval of the modifications.

##### 4.2.2.3.5 Step 5: Provide Current Plan

The IPP is provided to all stakeholders.

#### 4.2.2.4 Outputs of the Integrated Program Plan

There are five basic types of data in the IPP:

- **Data Type 1: Event.** This may be major program reviews, especially the AMS phase exit reviews; or they are sub-events.

- **Data Type 2: Accomplishment.** An accomplishment is the end goal of any program task tied to the event. The accomplishment may be the development of a deliverable or conduct of an analysis or test.
- **Data Type 3: Success Criteria.** A success criterion is the measure of whether the accomplishment was met or not. The criterion may be completion of the task, delivery of a report, or completion of the test. Success criteria may also include quality measures, such as the success of a test or the approval of a report.
- **Data Type 4: Task.** A task is the activity required to accomplish the objectives tied to the event. The task statement should reference the applicable WBS and SOW elements.
- **Data Type 5: Subtask.** A subtask is a subdivision of the task described in the major task.

#### 4.2.2.5 Integrated Program Plan Metrics

The primary IPP metric is publication and approval of the IPP at each AMS milestone. The IPP itself is a metric to evaluate the conduct of the program. The performance and conduct of the events, accomplishments, success criteria, tasks, and subtasks are program metrics.

#### 4.2.2.6 Integrated Program Plan Tools

The primary IPP tool is a generic template for any project using the SE elements and is contained in the FAST Toolset under “Required Planning Documents”. Specific projects may tailor this template to provide information pertaining to specific deliverables, tasks, and tools.

#### 4.2.2.7 Integrated Technical Planning Inputs to the Integrated Program Plan

The Integrated Program Plan and System Engineering. SE planning directly relates to elements of the SE process and is included as sections of the IPP. It describes how the SE process is applied to the given program or project at a summary level with detailed SE implementation activities discussed in supporting technical plans e.g., SEMP, MVP, SSMP etc. These planning sections become the tailored process. All IPP sections apply to every program; however, stakeholder direction or the nature of the program may dictate elimination of a planning section. As an example, a program without any avionics interfaces does not require a certification planning section. The program system engineer documents the rationale for eliminating any IPP sections or tailoring any process, and the program manager approves these actions. Table 4.2-2 lists the sections of an IPP and the SE elements from the SEMP that provide summary level inputs to the applicable IPP sections.

**Table 4.2-2. SE Inputs To The Integrated Program Plan**

| | Integrated Program Plan | |
|---|---|---|
| 1 | BACKGROUND | |
| 1.1 | Mission Need | Integrated Technical Planning (ITP) |
| 1.2 | Status | ITP |
| 2 | OVERVIEW | |
| 2.1 | Program Scope | ITP |
| 2.2 | Products | ITP |

| | Integrated Program Plan | |
|---|---|---|
| 3 | INTEGRATED PROGRAM FUNDING | ITP |
| | INTEGRATED PROGRAM SCHEDULE | ITP |
| 5 | PERFORMANCE | |
| 5.1 | Core Work Activities | ITP; Functional Analysis (FA); Synthesis (Syn); Trade Studies (TS); Interface Management (IM); Integrity of Analyses (IA); Specialty Engineering (SpecEng – Reliability, Maintainability and Availability – RM&A) |
| 5.2 | Program Management Work Activities | Requirements Management (RM); SpecEng (System Safety); Risk Management (RSK) |
| 5.3 | Procurement Work Activities | ITP |
| 6 | BENEFITS | RM |
| 7 | PHYSICAL INTEGRATION | Lifecycle Engineering (LCE – real property; deployment and transition); SpecEng (Hazardous Materials Management/Environmental Engineering; Electromagnetic Environmental Effects $E^3$ ) |
| 8 | FUNCTIONAL INTEGRATION | IM |
| 9 | HUMAN INTEGRATION | SpecEng (Human Factors Engineering) |
| 10 | SECURITY | SpecEng (Information Security Engineering) |
| 11 | IN-SERVICE SUPPORT | LCE (Integrated Logistics Support; Sustainment/Technology Evolution) |
| 12 | VALIDATION (INCLUDES TEST AND EVALUATION) | Validation and Verification (VV) |
| 13 | IMPLEMENTATION AND TRANSITION | LCE (Deployment and Transition; Disposal) |
| 14 | QUALITY ASSURANCE | SpecEng (Quality Engineering) |
| 15 | CONFIGURATION MANAGEMENT | Configuration Management (CM) |
| 16 | IN-SERVICE MANAGEMENT | LCE (ILS; Sustainment/Technology Evolution) |

The following describes which SE element is the source of information for each section of the IPP. The IPP summarizes the SE activities while the SEMP and other supporting technical plans describe the implementation detail.

#### 4.2.2.7.1 Background

Integrated Technical Planning is the source of information for summarizing the mission need and status of the program.

#### 4.2.2.7.2 Overview

Integrated Technical Planning is the source of information about the scope of the program and the primary deliverables.

#### 4.2.2.7.3 Integrated Program Funding

Integrated Technical Planning is the source for WBS, level of effort and schedule/duration information in sufficient detail to allow cost estimators to identify funding requirements.

#### 4.2.2.7.4 Integrated Program Schedule

Integrated Technical Planning is the source for WBS, milestone and SE activity information to allow for a logical networking of program activities to achieve program objectives.

#### 4.2.2.7.5 Performance

Within the “Core Work Activities” section, SE elements that are not specifically broken out as separate work activities are described here. SE elements such as Integrated Technical Planning, Functional Analysis, Synthesis, Trade Studies, Interface Management, Integrity of Analyses and Specialty Engineering sub-elements such as Electromagnetic Environmental Effects ($E^3$) and Reliability, Maintainability and Availability (RM&A) can be addressed to the extent that they apply.

Within the “Program Management Work Activities” section, specific SE elements such as Requirements Management, Specialty Engineering (System Safety) and Risk Management are identified as work activities requiring discussion. Program metrics are also described in this section with Integrated Technical Planning as the source.

Within the “Procurement Work Activity” section, those SE resources required to support Screening Information Request (SIR) release, Request for Proposal (RFP) development, proposal evaluations, and contractor requirements definition are identified.

#### 4.2.2.7.6 Benefits

Requirements Management is the source for technical or performance benefits.

#### 4.2.2.7.7 Physical Integration

SE inputs to this section of the IPP to identify space, facility, environment, hazardous materials and power work activities come from several sources including

#### 4.2.2.7.8 Functional Integration – Reserved

#### 4.2.2.7.9 Human Integration – Reserved

#### 4.2.2.7.10 Security - Reserved

#### 4.2.2.7.11 In-Service Support - Reserved

#### 4.2.2.7.12 Verification—See SEMP

#### 4.2.2.7.13 Implementation and Transition – Reserved

#### 4.2.2.7.14 Quality Assurance – Reserved

#### 4.2.2.7.15 Configuration Management—See SEMP

#### 4.2.2.7.16 In-Service Management - Reserved

It is recommended that, as part of the IPP, these planning sections be reviewed and changed whenever dictated by a change in the program or discovery of a discrepancy in the IPP. It is also recommended that changes to any these planning sections be coordinated with the SEMP, MVP, and other associated plans. All plans shall be reviewed at each phase exit review through the review following the last action required by the plan. After any plan is created following this manual, it is recommended that the plan be provided as reference material for future plan developers. It is recommended this be done through SE. It is also recommended that, along with the plan to be achieved, comments are provided to continue improvement of the plan development process.

### 4.2.3 System Engineering Management Plan

#### 4.2.3.1 Introduction to the System Engineering Management Plan

The System Engineering Management Plan (SEMP) is the only implementing document that integrates all SE activities. The SEMP unambiguously ties together the organization, direction and control mechanisms, and personnel to be used to attain program/project cost, performance, and schedule objectives. Prepared by the SE manager, this tool identifies and ensures control of the overall SE process and provides greater SE implementation detail than does the higher level IPP. The preliminary issue of the SEMP typically occurs in Mission Analysis; a final version is released in the first phase of Investment Analysis for JRC-2A, and a scheduled update occurs in the later phase of Investment Analysis, with additional updates as necessary to reflect changing input conditions throughout the program/project.

#### 4.2.3.2 Inputs to System Engineering Management Plan

The SEMP relates the technical requirements to program requirements, providing the structure to guide and control the integration of engineering activities needed to achieve the SE objectives consistent with a top-level management plan for the program. The SEMP includes more detailed planning for all SE elements to be executed as part of the program. Organizing to execute the system development involves defining the entire organizational structure (such as teams, work groups, and programs); establishing the responsibilities, authority, and

accountability of each; and clearly defining structural interfaces. It is recommended that this be an iterative process.

Information and data necessary to begin creation of a SEMP include the following:

- Knowledge of corporate strategy and goals
- Description and understanding of the overall program/project, usually found in an IPP or draft IPP
- Identification of top-level program/project requirements, usually from the MNS, final Requirements Document (fRD), change request, or one of the outputs developed during Mission Analysis Structure of engineering and other organizations, both internal (e.g., stakeholder) and external (e.g., supplier)
- Contract documents
- Any restrictions or constraints

#### 4.2.3.3 System Engineering Management Plan Steps

The following steps shall be employed to write a SEMP.

##### 4.2.3.3.1 Step 1: Collect Inputs

##### 4.2.3.3.2 Step 2: Analyze Inputs

To determine the SE effort required and committed to by program management, review the IPP, which is based on the nature and magnitude of the program/project.

- Large and complex system developments demand full System Engineering application to insure success
- Small-scale projects may be run under a subset process
- SE shall coordinate with IPT teams and program management, as their concurrence ensures the project team shall refer to and comply with the SEMP

##### 4.2.3.3.3 Step 3: Define Activities and Efforts

After evaluating all inputs, establish how the SE manager integrates them. Decisions that should be made involve the following:

- Tailoring the SE process
- Selecting an approach to ensure integration of engineering specialties
- How program team members will interact and communicate to execute technical program planning and control
- Identifying the explicit SE responsibilities to be assigned to each individual and organization, which, in total, will account for all such tasks planned
- The structure of the comprehensive SE Master Schedule (integrated with the IPP) for scheduled tasks

- Explicit guidance regarding development of each task for optimal inclusion, as program team members employs the SEMP as a handbook and reference source for essential information

#### 4.2.3.3.4 Step 4: Baseline

Prepare a draft SEMP for review and comment, using input from all affected engineering, engineering specialty, and program/project management organizations and, when appropriate, the stakeholders. The draft may also include contractual SEMP requirements, such as a CDRL Item and/or Data Item Description, with which all affected parties shall comply.

#### 4.2.3.3.5 Step 5: Interface With Other Processes/Plans

In addition to employing the IPP as an input during development, the SEMP interfaces with and forms a roadmap to other SE and engineering specialty plans (e.g., Master Verification Plan). The SEMP addresses all of the SE elements:

- Integrated Technical Planning (Section 4.2)
- Requirements Management (Section 4.3)
- Functional Analysis (Section 4.4)
- Synthesis (Section 4.5)
- Trade Studies (Section 4.6)
- Interface Management (Section 4.7)
- Specialty Engineering (Section 4.8)
- Integrity of Analyses (Section 4.9)
- Risk Management (Section 4.10)
- Configuration Management (Section 4.11)
- Validation and Verification (Section 4.12)
- Lifecycle Engineering (Section 4.13)
- Maintain System Engineering (Section 4.14)

#### 4.2.3.3.6 Step 6: Update and Maintain the Plan

It is recommended that throughout the program/project, the SE manager monitor inputs (especially the IPP). When there is a significant change in one or more inputs, the SEMP should be revised (by repeating the creation steps above).

### 4.2.3.4 Outputs of System Engineering Management Plan

Table 4.2-3 is a SEMP outline.

**Table 4.2-3. System Engineering Management Plan Outline**

| System Engineering Management Plan Outline | |
|---|---|
| SECTION 1 | INTRODUCTION |
| 1.1 | Scope |
| 1.2 | Purpose of the System Engineering Management Plan |
| 1.3 | Organization of the System Engineering Management Plan |
| 1.4 | SEMP Overview |
| 1.5 | *Program/Project name* System Description |
| 1.6 | Program Organization |
| 1.7 | System Engineering Responsibility Assignments |
| 1.8 | System Engineering Environment and Tools |
| 1.9 | System Engineering Metrics |
| SECTION 2 | SYSTEM ENGINEERING |
| 2.1 | System Engineering Process |
| 2.2 | Integrated Technical Planning |
| 2.3 | Requirements Management |
| 2.4 | Functional Analysis |
| 2.5 | Synthesis |
| 2.6 | Trade Studies |
| 2.7 | Interface Management |
| 2.8 | Specialty Engineering |
| 2.8.1 | System Safety Engineering |
| 2.8.2 | Human Factors Engineering |
| 2.8.3 | Quality Engineering |
| 2.8.4 | Reliability, Maintainability and Availability |
| 2.8.5 | Electromagnetic Environmental Effects ($E^3$) |
| 2.8.6 | Hazardous Materials Management/Environmental Engineering |
| 2.9 | Integrity of Analysis |
| 2.10 | Risk Management |
| 2.11 | Configuration Management |
| 2.12 | Validation and Verification |
| 2.13 | Lifecycle Engineering |
| 2.13.1 | Real Property Management |
| 2.13.2 | Deployment and Transition |
| 2.13.3 | Integrated Logistics Support |

| System Engineering Management Plan Outline | |
|---|---|
| 2.13.3.1 | Maintenance Planning |
| 2.13.3.2 | Maintenance Support Facility |
| 2.13.3.3 | Direct-Work Maintenance Staffing |
| 2.13.3.4 | Supply Support |
| 2.13.3.5 | Support Equipment |
| 2.13.3.6 | Training, Training Support, and Personnel Skills |
| 2.13.3.7 | Technical Data |
| 2.13.3.8 | Packaging, Handling, Storage, and Transportation (P,H,S&T) |
| 2.13.3.9 | Computer Resources Support |
| 2.13.4 | Sustainment/Technology Evolution |
| 2.13.5 | Disposal |
| 2.14 | Maintain System Engineering |
| SECTION 3 | |
| 3.1 | System Engineering Master Schedule |

Appendix E contains more detailed input and format information for the planning associated with all of the SE elements discussed in Section 2 of the SEMP.

### 4.2.4 Requirements Management Planning—See Appendix E for Details

This planning specifies the tasks, products, responsibilities, and schedule for managing requirements throughout product development. The planning is baselined at the JRC-2B in the IPP and is updated as necessary at subsequent exit reviews.

The planning section details the total effort in managing requirements. The work includes identifying and capturing requirements (Paragraph 4.3.2.1), analyzing and decomposing requirements (Paragraph 4.3.2.2), and allocating requirements (Paragraph 4.3.2.3).

### 4.2.5 Functional Analysis Planning—See Appendix E for Details

The Functional Analysis planning section of the SEMP specifies the tasks, products, responsibilities, and schedule for functional analysis throughout the development of the product. Because there is no program level SEMP in the early phases of the program (i.e., Mission Analysis and Investment Analysis), Functional Analysis in these phases is guided by the NAS-level SEMP. When the IPP is developed, the Functional Analyses is guided by the program's tailored SEMP. The planning section is baselined at the JRC-2B and is updated as necessary at subsequent exit reviews. This planning section details the total effort for managing functional analysis. This work includes analysis of the concept of operations and environment, the decomposition of functions into subfunctions, decomposing and allocating requirements to functions, evaluating alternative decompositions, defining functional sequences and timelines, defining functional interfaces, and documenting the functional baseline. These tasks are described in Functional Analysis (Section 4.4).

### 4.2.6 Synthesis Planning - Reserved

### 4.2.7 Trade Study Planning—See Appendix E for Details

The Trade Study planning documents the formal management planning regarding how alternative solutions to a problem or design issue associated with a program/project product development is to be assessed in a fair and impartial manner.

Trade study planning:

- Provides the formats for how trade study results and information are to be presented to management at design reviews
- Identifies the organization or person designated to be the trade study leader
- Identifies any tools that are to be used in performing of the trade study (i.e., cost models, computer simulations, test articles and fixtures, analytical tools)
- Provides the criteria (including constraints) under which the trade study is to be conducted
- Provides instructions on where trade study results and data are to be stored for future reference and which organization is responsible for maintaining the data

### 4.2.8 Interface Management Planning—See Appendix E for Details

Interface management (IM) planning ensures establishment of the formal management system of interface (I/F) controls that enable physical and functional compatibility between interfacing hardware, software, personnel, and facilities. This planning:

- Provides the means for identifying, defining, documenting, and controlling the interfaces at all levels of the system
- Provides the means for changing the interfaces as required by the evolution of the design and for resolving interface incompatibilities
- Guides management, control, and documentation of all system functional and physical interfaces
- Establishes the Interface Control Working Group (ICWG) and its policies and procedures
- Contains requirements and templates for preparing, revising, and processing the interface documentation; identifies products
- Establishes the participants of the I/F control process and their responsibilities
- Establishes the interface management schedule

The ICWG Chairman drafts the IM planning policies and procedures in the early phase of Investment Analysis concurrent with the SEMP and the SE Schedule. The ICWG Chairman updates and reviews the ICIM planning section of the IPP to reflect the system functional and physical architectures developed in later phase of Investment Analysis.

### 4.2.9 Specialty Engineering Planning - Reserved

#### 4.2.9.1 System Safety Management Planning

System safety is the application of engineering and management principles, criteria, and techniques to optimize safety within constraints of operational effectiveness, time, and cost throughout all program lifecycle stages. The NAS Modernization System Safety Management Plan (SSMP) governs system safety efforts conducted in the AMS. The SSMP requires each program to develop, as part of the IPP, an Integrated System Safety Program (ISSP) tailored to the program's safety needs. The ISSP calls for contractors or vendors to develop and maintain a System Safety Program Plan (SSPP) that details the planned safety activities. The SSPP describes safety assessments, tasks, and activities of system safety management and system safety engineering required to support the design process and to identify, evaluate, and eliminate or control hazards throughout the system lifecycle.

Government System Safety engineers in the program are responsible for generating the ISSP, and, typically, the System Engineering Council (SEC) approves it as the first step in the system safety program. Contractor System Safety engineers in the program are responsible for generating the SSPP; the Program Manager approves the document internally, and the SEC approves it externally. System safety is an integral element of system engineering applicable to all design stages. Consequently, the stakeholder typically requires the SSPP as early as possible in the program lifecycle, usually within 60 to 90 days after contract award. Updates to the SSPP are necessary from stage to stage. Significant program changes may also warrant an update.

A comprehensive, approved SSPP provides value to the overall program. Misunderstandings are avoided regarding the safety definitions, scope of safety analysis, and risk-resolution procedures. The SSPP serves to increase safety awareness within the integrated team, building system safety into the product. The SSPP is tailored guidance for the System Safety Manager or engineer. Finally, the SSPP serves as an important audit trail, justifying the safety work performed and the methodology for safety decisions made.

#### 4.2.9.2 RMA Planning - Reserved

#### 4.2.9.3 Human Integration Planning—See AMS section

#### 4.2.9.4 Security Planning - Reserved

#### 4.2.9.5 Quality Assurance Planning - Reserved

#### 4.2.9.6 Hazardous Material Management Planning - Reserved

#### 4.2.9.7 Electromagnetic Interference/Electromagnetic Compatibility and Environmental Planning - Reserved

### 4.2.10 Analysis Management Planning—See Appendix E for Details

The Analysis Management planning section of the IPP is compiled following JRC-1 approval. It supports the objective of that process: "to create high likelihood that the program's analyses are credible, useful, and sufficient." Analysis Management planning defines the analyses to be performed throughout the program and the operational criteria for the analytic tools to be used,

as well as the users and the requirements for verifying that the results are correct and sufficient. As a part of the IPP, this section is reviewed with any other plans at the JRC-2B.

### 4.2.11 Risk Management Planning—See Appendix E for Details

Risk is inherent in every program. Stakeholders know this and expect contractors to address risks in program plans. SE addresses three facets of risk: technical, schedule, and cost. Technical risks include all events that may prevent the program from satisfying contractual requirements, including performance, supportability, maintainability, and regulatory requirements. Schedule risks are events that may prevent timely execution of tasks identified in the IPP. Cost risks are events that may cause actual expenditures to exceed estimated costs.

Risk management is a key process within SE. The program and functional managers implement it by ensuring appropriate resources are applied to reduce risk to acceptable levels. Risk management consists of five essential components: identify risks, analyze risks, identify mitigation options, implement risk-reduction plan, and monitor risks.

The risk management planning section describes the approach, methods, procedures, and criteria for risk management and its integration into the program decision process. It is continually updated throughout the program life with the IPP.

### 4.2.12 Configuration Management Planning—See Appendix E for Details

Configuration Management planning documents the formal management system of CM to ensure that the integrity and continuity of the design, engineering, and cost tradeoff decisions made between technical performance, producibility, operability, testability, and supportability are recorded, communicated, and controlled by program and functional managers. CM planning provides the means for the:

- Configuration Identification process that identifies the functional and physical characteristics of selected system components, designated as configuration items (CI), during the system's acquisition lifecycle
- Configuration Control process that controls the changes to CIs during the system's acquisition lifecycle
- Configuration Status Accounting process that records/reports change processing and implementation status
- Configuration Audits process that supplies current descriptions of developing hardware configuration items (HWCI), computer software configuration items (CSCI), and the system itself

The Configuration Management Organization typically owns this planning section. The planning section may be initiated by inputs from the SE process as early as the Investment Analysis, phase one, but formally starts at Investment Analysis, phase two, and continues throughout the program lifecycle as the system develops and is modified.

### 4.2.13 Master Verification Plan (Includes Test and Evaluation Planning)—See Appendix E for Details

The MVP contains both validation and verification planning. Validation is the process of proving that the right system is being built (i.e., that the system requirements are unambiguous, correct,

complete, consistent, traceable to needs, operationally and technically feasible, and verifiable). The validation planning process is conducted to demonstrate that the requirements for a system are clearly understood and that it is possible to satisfy them through design work using available state-of-the-art technology, funding, and schedule. Verification is the process (tasks, actions and activities) of confirming that evolving system solutions comply with functional, performance, and design requirements that spell out stakeholder (internal and external) expectations of capabilities, as well as performance and characteristics of the developed system. Product verification may occur during any phase of a product development cycle, but is more likely to occur after the product Preliminary Design Review (PDR). Verification is the process that ensures that system requirements have been met by the design solution and that the system is ready for use in its operational environment. This means that a verified system may demonstrate that it complies with mission need and meets functional, performance, allocated, derived, and interface requirements, as well as design and allocated constraints that achieve customer needs.

The MVP objective is to define all verification activities that demonstrate the system's capability to meet the specification requirements.

### 4.2.14 Integrated Lifecycle Planning – Reserved

#### 4.2.14.1 Real Property Management – Reserved

#### 4.2.14.2 Deployment and Transition – Reserved

#### 4.2.14.3 Integrated Logistics Support – Reserved

#### 4.2.14.4 Sustainment/Technology Evolution – Reserved

#### 4.2.14.5 Disposal – Reserved

### 4.2.15 Reviews and Audits

#### 4.2.15.1 Technical Reviews

##### 4.2.15.1.1 Joint Resources Council 1 Review

##### 4.2.15.1.2 Joint Resources Council 2A Review

##### 4.2.15.1.3 Initial System Requirements Review

##### 4.2.15.1.4 Joint Resources Council 2B Review

##### 4.2.15.1.5 Preliminary Design Review

##### 4.2.15.1.6 Critical Design Review

##### 4.2.15.1.7 Joint Resources Council 3 Review

#### 4.2.15.2 Audits

##### 4.2.15.2.1 Functional Configuration Audit

##### 4.2.15.2.2 Physical Configuration Audit

# 4.3 Requirements Management

## 4.3.1 Introduction to Requirements Management

The Requirements Management process, an element of System Engineering (SE), is an activity that spans the program's entire lifecycle. It is associated with iterative identification and refinement, to successively lower levels, of the top-level requirements, functional baselines and architectures, and synthesis of solutions established for the preferred system concept. For the purposes of Requirements Management, a system or a product shall mean any physical product being designed, developed, and/or produced, or any intangible product such as the development of a process or service-based product.

The Requirements Management process defines, collects, documents, and manages all requirements, including the complete requirements set consisting of the Mission Need Statement (MNS), the initial Requirements Document (iRD) and final Requirements Document (fRD), and the system and procurement specifications. A requirement is defined as a condition or capability that shall be met or exceeded by a system or a component to satisfy a contract, standard, specification, or other formally imposed document. Executing this process results in the authorized, organized, and baselined set of requirements for the product. These requirements are presented as requirements sets, usually in the form of requirements documents, to all other applicable SE and Federal Aviation Administration (FAA) processes. To effectively develop and manage system requirements, all requirements shall be developed through this process.

### 4.3.1.1 Process Description

#### 4.3.1.1.1 Purpose

Requirements Management's purpose is to establish a layered approach that defines the necessary and sufficient attributes of the lower-level system components required for the product's successful development, production, deployment, operation, and disposal. Successful completion of this process is measured by the acceptable transformation of stakeholder needs into discrete, verifiable, low-level requirements. The process identifies, clarifies, balances, and manages the entire requirements set through interactive dialogue with all stakeholders. The top-level process appears in Figure 4.3-1.

#### 4.3.1.1.2 Requirements Management Objectives

Requirements Management is an iterative process that:

- Identifies and captures the requirements applicable to the system
- Analyzes and decomposes the requirements into clear, unambiguous, traceable, and verifiable requirements
- Allocates the requirements to the appropriate component within the system hierarchy and/or to the appropriate organizational entities
- Derives lower-level requirements from higher-level requirements in the system hierarchy
- Establishes the method of verification for each requirement
- Ensures that the product complies with the requirements

- Manages, documents, and controls the requirements and changes to them in a traceable manner

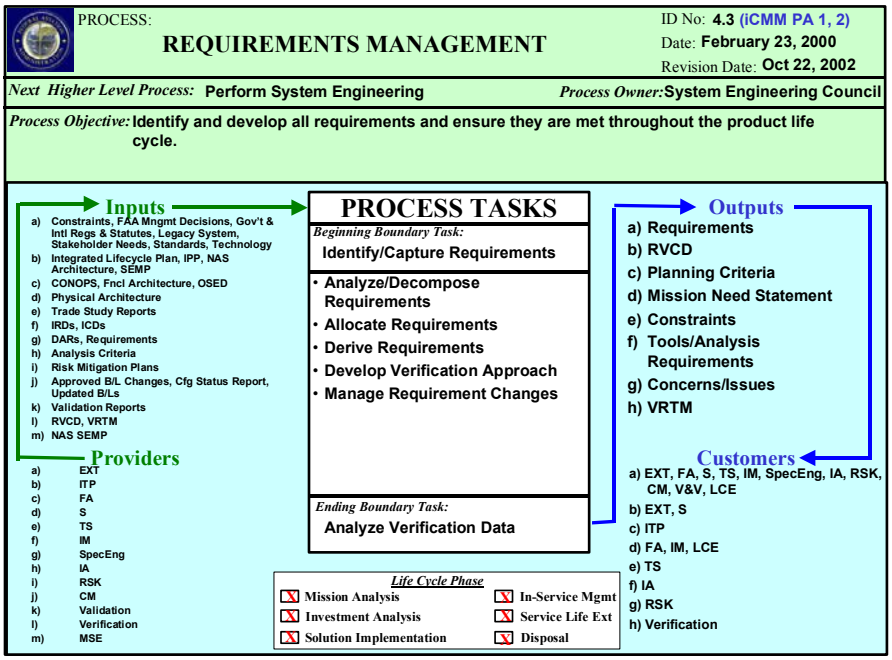



**Figure 4.3-1. Requirements Management Process-Based Management Chart**

#### 4.3.1.2 Management

The Requirements Management process bridges integrated product development system stages. The products of this process are baselined in accordance with the milestones established in the Integrated Program Plan (IPP) for the applicable project. Prerequisites for successful performance of the process are:

- Empowering a requirements analysis team with the authority and mission to execute the process
- Assigning an experienced team leader knowledgeable in SE principles and committed to the standard SE methods documented herein
- Assigning team members that are experienced and knowledgeable in relevant engineering, manufacturing, operational, specialty engineering, and support disciplines
- Establishing the criteria for decisionmaking and any supporting tools
- Completing the relevant training of team members in using this process and relevant tools
- Defining the formats of the output deliverables from this activity

### 4.3.2 Inputs to Requirements Management

An input to the Requirements Management process is defined as information received during the process. Inputs are classified according to their source (i.e., external or internal). External

inputs come from sources outside SE. Internal inputs come from other SE processes as described in this manual. Typical inputs include Stakeholder Needs and objectives, missions, measures of effectiveness (MOE) and measures of suitability (MOS), environments, key performance parameters, technology base, output requirements from prior application of SE, and program decision requirements. Input requirements shall be comprehensive and defined for both system products and system processes, including the eight lifecycle functions of development, manufacturing, verification, deployment, operations, support, training, and disposal.

Requirements Management is an iterative process that flows from a high level to a low level of requirements. Therefore, some of the inputs described in the following paragraphs may be inputs to one stage of the requirements development process and outputs of other stages. All requirements sources described were, at one point in the process, inputs and shall be captured. The inputs to the Requirements Management process are as follows.

#### 4.3.2.1 External Inputs

External inputs come from outside SE's boundaries.

##### 4.3.2.1.1 Constraints

A Constraint is a boundary condition within which the system remains while satisfying the aggregate system requirements.

###### 4.3.2.1.1.1 External Constraints

External constraints, including guidelines and assumptions, shall be identified. External constraints are imposed from outside the project or system boundaries. External conditions under which the mission is to be performed and systems developed are described. The conditions may include cost, schedule, performance, technology, use of infrastructure, labor/management agreements, and programmatic constraints. Additional assumptions concerning programmatics, technology, and environments that may be required are captured.

###### 4.3.2.1.1.2 Internal Constraints

Internal constraints, including assumptions, guidelines, and program-specific constraints, shall be identified. Internal constraints are imposed from within the project or system boundaries but outside of SE. Program-specific conditions under which the mission is to be performed and systems developed are described. The conditions may include cost, schedule, performance, technology, use of infrastructure, and programmatic constraints. Additional assumptions concerning programmatics, technology, and environments that may be required are captured.

###### 4.3.2.1.1.3 Program-Specific Constraints

Program-specific organizational constraints and assumptions are captured, as well as program-specific needs, schedule constraints, and events.

###### 4.3.2.1.1.4 Technology Constraints

Technology availability or technology constraints are captured. Technology necessary to satisfy requirements and the resulting derived requirements are described. Constraints identify the

envelope of the technology operation. These inputs may include identifying key technologies, performance, maturity, cost, and risks; they may also include technology breakthroughs and forecasts.

#### 4.3.2.1.2 Standards, Specifications, and Handbooks

Specified government standards, external standards, and general specifications or handbooks to be employed on the program are identified. The most common standards, specifications, and handbooks used in FAA requirements documents appear in Appendix A.

##### 4.3.2.1.2.1 Standards

A standard is a document that establishes engineering and technical requirements for processes, procedures, practices, and methods that have been adopted as standard. Standards may also establish requirements for selection, application, and design criteria for material. The FAA, Department of Defense (DoD), other U.S. Government agencies, the RTCA, international organizations, and commercial standards organizations publish standards.

###### 4.3.2.1.2.1.1 RTCA Standards

The RTCA publishes standards as Minimum Operational Performance Standards (MOPS) and Minimum Aviation System Performance Standards (MASPS).

###### 4.3.2.1.2.1.1.1 Minimum Operational Performance Standards

The MOPS contain performance requirements for avionics. The standards describe typical equipment applications and operational goals and establish the basis for required performance and test procedures for verification under a common set of standards. Definitions and assumptions essential to proper understanding are provided, as well as installed equipment tests and operational performance characteristics for equipment installations. The MOPS also provide information that explains the rationale for equipment characteristics and stated requirements.

###### 4.3.2.1.2.1.1.2 Minimum Aviation System Performance Standards

The MASPS address the user-level service requirements used to qualify the system for operational acceptance and to allocate requirements for the subsystems (including avionics). The standards provide information that explains the rationale for system characteristics, operational goals, requirements, and typical applications.

##### 4.3.2.1.2.2 Specifications

A specification is a document prepared specifically to support an acquisition that clearly and accurately describes the essential technical requirements for purchased material or products and the criteria for determining whether the requirements are satisfied. The FAA, DoD, other U.S. Government agencies, international organizations, and commercial standards organizations publish specifications.

#### 4.3.2.1.2.3 Handbooks

A handbook is a guidance document that contains information or guidelines for use in design, engineering, production, acquisition, and/or supply management operations. These documents present information, procedural and technical use data, or design information related to processes, practices, services, or commodities. Handbooks provide industry with reference materials that help to standardize FAA assets. Use of handbooks is optional unless required by a specification or contract document. The FAA, DoD, other U.S. Government agencies, international organizations, and commercial standards organizations publish handbooks.

#### 4.3.2.1.2.4 Federal Aviation Administration Orders

An FAA order is a permanent directive on individual subjects or programs that apply to the FAA. It directs action or conduct using action verbs. Orders also prescribe policy, delegate authority, and empower and/or assign responsibility for compliance with stated requirements or direction. Orders empower or direct only FAA personnel and carry no weight with contractors. Thus, orders shall not be used in contract documents. They are not referenced in requirements documents but are used as inputs with the potential to generate requirements.

#### 4.3.2.1.2.5 National Airspace System Management Directives

NAS-MD-001, “National Airspace System Master Configuration Index,” lists all baselined equipment and software currently operational or under procurement for the National Airspace System (NAS) with current approved baseline documentation. FAA and contractor personnel use NAS-MD-001 to identify configuration items and documentation requiring NAS Change Proposals (NCP).

### 4.3.2.1.3 Federal Aviation Administration Management Decisions

Management decisions that are imposed on the system from the national, department, or agency level are captured.

### 4.3.2.1.4 Government Policy

#### 4.3.2.1.4.1 Government Regulations and Statues

Government statutes and military and civilian regulations impacting the system are identified, including requirements incorporated into legislation (e.g., safety or security requirements). These requirements also include government standards that have been mandated as part of a contract.

#### 4.3.2.1.4.2 International Policy

The International Civil Aviation Organization (ICAO) develops and publishes international Standards and Recommended Practices (SARPS). A standard is any specification for physical characteristics, configuration, material performance, personnel, or procedure that is applied uniformly for the safety or regularity of international air navigation and to which the international aviation community conforms. A recommended practice is identical to a standard except that it is not considered necessary—only desirable.

#### 4.3.2.1.4.3 Federal Aviation Administration Policy

This category covers all FAA agency-wide management decisions and policy requirements imposed by FAA agency-wide mandate. The category may include technical, operational, acquisition, financial, and other requirements. FAA policy is invoked using the FAA Directives System, as described in FAA Order 1320.1, “FAA Directives System.”

#### 4.3.2.1.4.4 Acquisition Management System Limitations

New or revised directions and limitations established by the Acquisition Management System (AMS) are identified.

### 4.3.2.1.5 Legacy Systems

Requirements from past and current systems are captured and analyzed for applicability.

### 4.3.2.1.6 Stakeholder Needs

#### 4.3.2.1.6.1 National Airspace System Concepts of Operations Document

The NAS Concepts of Operations (CONOPS) document provides a CONOPS from the perspectives of NAS users and service providers. It is the basis for an incremental benefits-driven approach toward NAS evolution. The document is arranged in a phases-of-flight approach, including Flight Planning, Surface, Arrival/Departure, En Route, and NAS Management. It is the source document for all NAS operational requirements.

#### 4.3.2.1.6.2 Mission Need Statement

The MNS is the first document to translate the NAS CONOPS into the needs and requirements of the users and service providers. It identifies the decision factors relevant to a capability shortfall or a technological opportunity to satisfy a mission more efficiently or effectively. The MNS justifies, in rigorous analytical terms, the need to resolve a shortfall in services required by its users and service providers or to explore a technological opportunity for more efficient and effective mission performance. The MNS identifies the mission area, needed capability, current capability, capability shortfall, impact to users and service providers if the shortfall is not resolved, benefits, timeframe for resolving the shortfall, criticality of the mission, and resource estimate.

#### 4.3.2.1.6.3 Operational Scenarios

Operational scenarios provided by the user describe how the CONOPS is implemented. They may be incorporated into the MNS or provided as a separate document.

#### 4.3.2.1.6.4 Requirements Document

The requirements document establishes the operational framework and performance baseline, traces Functional Analysis to the NAS CONOPS and the MNS, and is the primary source document for the system requirements. This document is the principal force driving the search for a realistic and affordable solution to the mission need. The iRD is developed early in the process by the sponsoring organization. It translates the need in the MNS into initial top-level requirements that address concerns such as performance, supportability, physical and

functional integration, human integration, security, test and evaluation, implementation and transition, quality assurance, configuration management, and in-service management. The iRD does not describe a specific solution to a mission need. It is recommended that the iRD not preclude leasing, commercial, or nondevelopment solutions. The fRD defines exactly the operational concept and requirements that are to be achieved and is the basis for evaluating the readiness of resultant products and services to become operational.

#### 4.3.2.1.7 External Interface Studies

System external interface studies and analyses that characterize and define the interfaces between the system and external environment are reviewed or conducted. These studies identify functional and physical characteristics between two or more elements that are provided by different agencies, as well as resolve problems. Topics include issues, option assessments, impact assessments, interfaces and connections, interferences, and configuration options.

#### 4.3.2.1.8 National Airspace System Architecture

The NAS Architecture is a strategic and evolutionary plan for modernizing the NAS that supports investment analysis tradeoffs. It focuses on defining and delivering the services that meet aviation industry and public needs, which it accomplishes by decomposing the services into capabilities that are the functions and activities necessary to deliver a service. Each capability is defined by the implementations steps required to deliver the capabilities. Each implementation step is defined in terms of the mechanisms required to provide each step. Finally, each mechanism is defined in terms of the people, systems, and support activities provided by the procuring office. The NAS Architecture presents a comprehensive design that shows each major mechanism within the NAS, including interfaces and data flows. Use of a documented design, complete with traceable requirements, as the foundation for the architecture not only provides a complete picture of the NAS but also provides a roadmap for implementing future enhancements.

#### 4.3.2.1.9 National Airspace System System Engineering Management Plan

The NAS System Engineering Management Plan (NAS SEMP) defines the relationship between the NAS SE levels, including requirements management, and the roles and responsibilities of each level. The SE levels are defined in the NAS SEMP and include the Enterprise, Domain, and Functional levels.

### 4.3.2.2 Internal Inputs

Internal inputs come from inside SE's boundaries.

#### 4.3.2.2.1 Technical Planning

##### 4.3.2.2.1.1 Integrated Program Plan

The Requirements Management planning section of the IPP (Integrated Technical Planning (Section 4.2)) specifies the tasks, products, responsibilities, and schedules needed to manage requirements throughout product development. It details the total work effort for managing requirements. This work includes "Task 1: Identify and Capture Requirements" (Paragraph 4.3.3.1); "Task 2: Analyze and Decompose Requirements" (Paragraph 4.3.3.2); "Task 3:

Allocate Requirements" (Paragraph 4.3.3.3); "Task 4: Derive Requirements" (Paragraph 4.3.3.4); and "Task 6: Manage Requirements Changes" (Paragraph 4.3.3.6).

#### 4.3.2.2.2 Functional Analysis

##### 4.3.2.2.2.1 Concept of Operations

A CONOPS, which is a user-oriented document that describes a proposed system's functional requirements from the user's viewpoint, is obtained from the Functional Analysis process (Section 4.4). The CONOPS document is written to communicate overall quantitative and qualitative system characteristics to the user, buyer, developer, and other organizational elements. The CONOPS aids in requirements capture and communicates the need to the developing organization. The CONOPS describes the existing system, current environment, users, interactions among users and the system, and organizational impacts. A CONOPS is essentially a top-level narrative Functional Analysis and is the basis for developing the MNS.

##### 4.3.2.2.2.2 Functional Architecture

Every function required to satisfy a system's operational needs shall be identified and defined. Once defined, the functions are used to define system requirements, and a Functional Architecture is developed based on the identified requirements. The process is then taken to a greater level of detail, as the identified functions are further decomposed into subfunctions, and the Functional Architecture and requirements associated with those functions are each decomposed as well. This process is iterated until the system has been completely decomposed into basic subfunctions, and each subfunction at the lowest level is completely, simply, and uniquely defined by its requirements. In this process, the interfaces between each of the functions and subfunctions are fully defined, as are the interfaces within the environment and external systems. The functions and subfunctions are arrayed in a Functional Architecture to show their relationships and internal and external interfaces.

The Functional Architecture includes a definition of the functions that the system needs to perform and is developed into Primitive Requirements Statements (PRS). "Task 2: Analyze and Decompose Requirements" (Paragraph 4.3.3.2) of the Requirements Management process develops these PRSs into Mature Requirements Statements (MRS).

##### 4.3.2.2.2.3 Operational Services and Environmental Description

The Operational Services and Environmental Description (OSED) is a complete system description that includes information on all known hardware, software, people, procedures, and ambient and operational environments in the system. It consists of everything inside and outside the system that affects system performance and that is affected by system operation or both.

The OSED is used as a source to derive lower-level requirements. It describes many system characteristics that are nonfunctional, such as environments, and that are not described in the Functional Architecture. Nonfunctional requirements are derived from the OSED in "Task 4: Derive Requirements" (Paragraph 4.3.3.4).

#### 4.3.2.2.3 Synthesis

##### 4.3.2.2.3.1 Physical Architecture

The Physical Architecture allocates requirements to the physical hardware and/or software during the Synthesis process (Section 4.5). If requirements conflicts are discovered during the development of the Physical Architecture, those requirements are cycled back through the Requirements Management process for evaluation, which may result in conducting a Trade Study (Section 4.6), reallocating the requirement, or deriving lower-level requirements.

#### 4.3.2.2.4 Trade Studies

Trade Studies (Section 4.6) may be conducted within and across functions to support decisions during any stage of the system's lifecycle. They quantify through metrics the consequences of opting for various system alternatives, traceable to stakeholder requirements, that may be imposed by the requirements development process. They support allocating performance requirements and determining requirements or Design Constraints; they are also used in evaluating alternatives. Trade Studies usually result in derived requirements that are developed into MRSs in "Task 2: Analyze and Decompose Requirements" (Paragraph 4.3.3.2).

##### 4.3.2.2.4.1 Trade Study Reports

Trade Study Reports identify requirements that are affected by the results of each Trade Study (Section 4.6). The new, changed, or derived requirements flow through the entire Requirements Management process and may result in changes to the requirements baseline.

##### 4.3.2.2.4.2 Feasibility Assessments

The Feasibility Assessment may be conducted to assess the difficulty in achieving program goals within the Constraints. Assessment results consider various aspects, such as technical, cost, and schedule, across the lifecycle. It provides information on the expectations for success, considering identified technology development needs in view of program and mission schedule and cost constraints. It also assesses the range of costs and benefits associated with several alternatives for solving a problem.

##### 4.3.2.2.4.3 Derived Requirements

Derived requirements ("Task 4: Derive Requirements" (Paragraph 4.3.3.4)) may be developed through Trade Studies (Section 4.6) and not provided by external sources, such as the user, service provider, or government agencies.

#### 4.3.2.2.5 Interface Management

The inputs from Interface Management (Section 4.7) identify, describe, and define interface requirements to ensure compatibility between interrelated systems and between system elements.

#### 4.3.2.2.5.1 Interface Requirements Document

The Interface Requirements Document (IRD) defines requirements associated with external physical and functional interfaces between the particular system and other associated system(s).

#### 4.3.2.2.5.2 Interface Control Document

The Interface Control Document (ICD) is clearer, more detailed documentation of the interface requirements that define the “as built” interface.

### 4.3.2.2.6 Specialty Engineering

Specialty Engineering (Section 4.8) defines and evaluates a system’s specific areas, features, or characteristics. Specialty Engineering supplements the design process by defining these characteristics and assessing their impact on the program. Specialty Engineering studies often find characteristics that create a need for new or different requirements or a conflict between two or more requirements. The Specialty Engineering process develops the new or changed requirements, which become inputs to the Requirements Management process.

#### 4.3.2.2.6.1 Design Analysis Reports

Design Analysis Reports (DAR), which document the results of a specific Specialty Engineering analysis with rationale, are inputs to the Requirements Management process. Each DAR contains a description of the system's special characteristics, a list of existing requirements that have undergone the Validation and Verification process (Section 4.12), residual risks, and candidate requirements found as a result of the analysis.

The rationale supplementing the DARs includes the scope, ground rules, assumptions, constraints, methods, and tools applicable to the analysis.

#### 4.3.2.2.6.2 Derived Requirements

The Specialty Engineering process (Section 4.8) provides analysis that typically defines, validates, or verifies requirements. Occasionally, the analysis discovers system characteristics that are not adequately specified in the existing specification or requirements documents. When such discoveries occur, Specialty Engineering defines the necessary requirements that are consistent with the area of Specialty Engineering and the requirements standards described in Requirements Management.

#### 4.3.2.2.7 Integrity of Analysis

##### 4.3.2.2.7.1 Analysis Criteria

If the Requirements Management process requires an analysis or selection of a tool, Analysis Criteria for that analysis or selection are captured. The Analysis Criteria for conducting a required analysis is contained within the Analysis Management Plan.

#### 4.3.2.2.8 Risk Management

##### 4.3.2.2.8.1 Risk Mitigation Plans

Concerns/Issues identified by any SE process are analyzed in the Risk Management process (Section 4.10). Risk Mitigation Plans that result from risk analysis become inputs to the Requirements Management process. Requirements that present a risk are processed through the Requirements Management process for reanalysis, reallocation, and rederivation, as needed.

#### 4.3.2.2.9 Configuration Management

##### 4.3.2.2.9.1 Approved Baseline Changes

Approved changes to the baselined requirements set are captured from the Configuration Management process (Section 4.11). “Step 6: Manage Requirements Changes” (Paragraph 4.3.3.6) inserts the Approved Baseline Changes into the requirements set.

##### 4.3.2.2.9.2 Configuration Status Reports

Configuration Status Reports are captured from the Configuration Management process (Section 4.11). “Step 6: Manage Requirements Changes” (Paragraph 4.3.3.6) uses these reports to maintain a status accounting of all requirements.

##### 4.3.2.2.9.3 Updated Baselines

Updated Baselines are captured from the Configuration Management process (Section 4.11). “Step 6: Manage Requirements Changes” (Paragraph 4.3.3.6) controls the updated baseline configuration.

#### 4.3.2.2.10 Validation

The Validation process (Section 4.12) determines if the requirements produced by the Requirements Management process are sufficiently correct and complete. Requirements that are not validated are captured and resubmitted to the Requirements Management process.

##### 4.3.2.2.10.1 Validation Report

The Validation Report summarizes the results of the Validation process (Section 4.12) and communicates the Validation Table to the Requirements Management process.

The Validation Report contains:

- Summary of validation results

- Description of the system and program
- Validation methodology used
- Unvalidated requirements
  - List of nonconforming requirements
  - Recommendations for correction of nonconforming requirements
- Validation Table
- Discussion of trends and patterns of failure, evidence of systemic failings, and emerging threats to system services.

#### 4.3.2.2.10.2 Validation Table

The Validation Table is a listing of all requirements that describes if a requirement has been validated, where the requirement may be found, source of validation, corrective action to be taken if necessary, and the corrective action owner. Table 4.12-1 in Validation and Verification (Section 4.12) is an example of a Validation Table. The completed Validation Table is included in the requirements document and is the basis for the Verification process.

#### 4.3.2.2.11 Verification

The Verification process (Section 4.12) determines that applicable requirements are satisfied by the design solution.

#### 4.3.2.2.11.1 Verification Requirements Traceability Matrix

The Validation Table from the Validation process (Section 4.12) is further refined into a Verification Requirements Traceability Matrix (VRTM), the heart of the Verification process. The strategy or method used to verify each requirement is specified in a Verification Requirement, and the Verification Requirements are listed in the VRTM. The VRTM defines how each requirement (functional, performance, and design) is to be verified, the stage in which verification is to occur, and the applicable verification levels. The VRTM establishes the basis for the verification program. The VRTM is initiated by the Requirements Management process, which sends it to the Verification process, which returns it to Requirements Management when verification has been completed.

#### 4.3.2.2.11.2 Requirements Verification Compliance Document

The Requirements Verification Compliance Document (RVCD) provides evidence of compliance for each requirement at all levels and to each VRTM requirement. The flowdown from the requirements documents to the VRTM completes the full requirements traceability. Compliance with all requirements ensures that the system-level requirements have been met. The RVCD defines, for each requirement, the verification methods and corresponding compliance information. The results of the Verification process (Section 4.12), including evidence of completion, are recorded and documented in the RVCD. It is recommended that the RVCD contain information regarding the results of each verification activity, as well as a description and disposition of conformance, nonconformance, conclusions, and recommendations. Compliance information provides either the actual data or a reference to the location of the actual data that shows compliance with the requirement. The document also includes a section that details any noncompliance. It is recommended that this section also specify appropriate

reverification procedures. The Requirements Management process captures noncompliant requirements, leading to a decision on disposition of the noncompliant requirement.

### 4.3.3 Requirements Management Process Tasks

The following tasks are necessary to perform this process:

- Identify and Capture Requirements
- Analyze and Decompose Requirements
- Allocate Requirements
- Derive Requirements
- Establish Requirements Verification Methods
- Manage Requirements

#### 4.3.3.1 Task 1: Identify and Capture Requirements

##### 4.3.3.1.1 Description

The Identify and Capture Requirements activity identifies, prioritizes, and extracts all written directives, including documented stakeholder negotiations/discussions, and internally derived requirements that are relevant to the particular stage of the system lifecycle. This activity is performed on the entire system, including any requirements that are known at this stage about how the system shall perform during its lifecycle and any constraints imposed on the system design/production by stakeholders and internal functions (i.e., manufacturing, product support, agency-level policies, suppliers). There are many different types, or categories, of requirements, as identified and defined in Paragraph 4.3.3.2.1.4.3. Requirements are typically categorized by the stage of the system lifecycle in which the requirement is obtained and by the function/user that generates the requirement. The primary objective is to consolidate baseline or approved system requirements so that they may serve as a foundation for later refinement and/or revision by subsequent functions in SE. This consolidation also allows an unambiguous and traceable flowdown of source requirements throughout the NAS Architecture as well as the product hierarchy. It is also important to negotiate with both external and internal stakeholders to reach agreement on which documents and to what level requirements need to be traced. This activity helps to ensure that the visibility stakeholders expect to obtain from requirements traceability may be achieved. This foundation needs to be as complete and accurate as possible and shall be fully traceable to the requirements source documentation.

##### 4.3.3.1.2 Scope

The scope of the requirements set shall include sufficient specification of all the system functions and all the external interfacing systems, including the system environment. This task may require considering a wider domain than the immediate physical boundary of the product and its components. Different boundaries may need to be defined for different states, modes, and capabilities. Refinement of these boundary definitions is an iterative process that occurs as more information is discovered about the true nature of the required system functions and performance (Interface Management (Section 4.7)). In this process, hardware, software, and system requirements are analyzed and refined to ensure that they are consistent, clear, valid, feasible, compatible, complete, and verifiable and that they do not include detail design information.

#### 4.3.3.1.3 Result

The result of performing this activity shall be a baseline set of requirements. The requirements shall be captured in an organized fashion. It is recommended the that information be readily accessible for reference by other program personnel as needed. This activity is the basis for discovering and successively refining the requirements to be recorded and maintained over the product's lifecycle.

#### 4.3.3.1.4 Compatibility

The selected requirements methodology shall be compatible with other methodologies applied across the FAA, and the analysis methodology supported with the necessary tools, as required by the Integrity of Analysis process (Section 4.9).

#### 4.3.3.1.5 Detailed Task 1 Description

Figure 4.3-2 describes the flow of the Identify and Capture Requirements task.

##### 4.3.3.1.5.1 Task 1.1: Define Stakeholder Expectations

Stakeholder expectations are defined and quantified, and stakeholder expectations in the FAA come from the operational stakeholder in the form of:

- CONOPS
- MNS
- iRD or fRD

They are transformed into baselined requirements sets at a successively lower level through iteration of the Requirements Management process. It is recommended that the definition of stakeholder expectations be balanced with an analysis of their effects on the overall system design and performance as well as on human engineering; knowledge, skills, and abilities; availability; reliability; safety; and training requirements of the humans required to support lifecycle processes. Stakeholder expectations include:

- What the system is to accomplish (functional requirements)
- How well each function is to be performed (performance requirements)
- The operational and ambient environment in which the system is to be operated
- Constraints under which the system is to be developed or operated (e.g., funding, cost or price objectives, schedule, technology, nondevelopmental and reusable items, physical characteristics, and hours of operation per day)

##### 4.3.3.1.5.2 Task 1.2: Define Project and Corporate Constraints

Project and corporate constraints that impact design solutions shall be identified and defined.

The NAS Architecture may also impose long-range planning constraints through the approved capabilities and implementation steps.

#### 4.3.3.1.5.2.1 Project Constraints

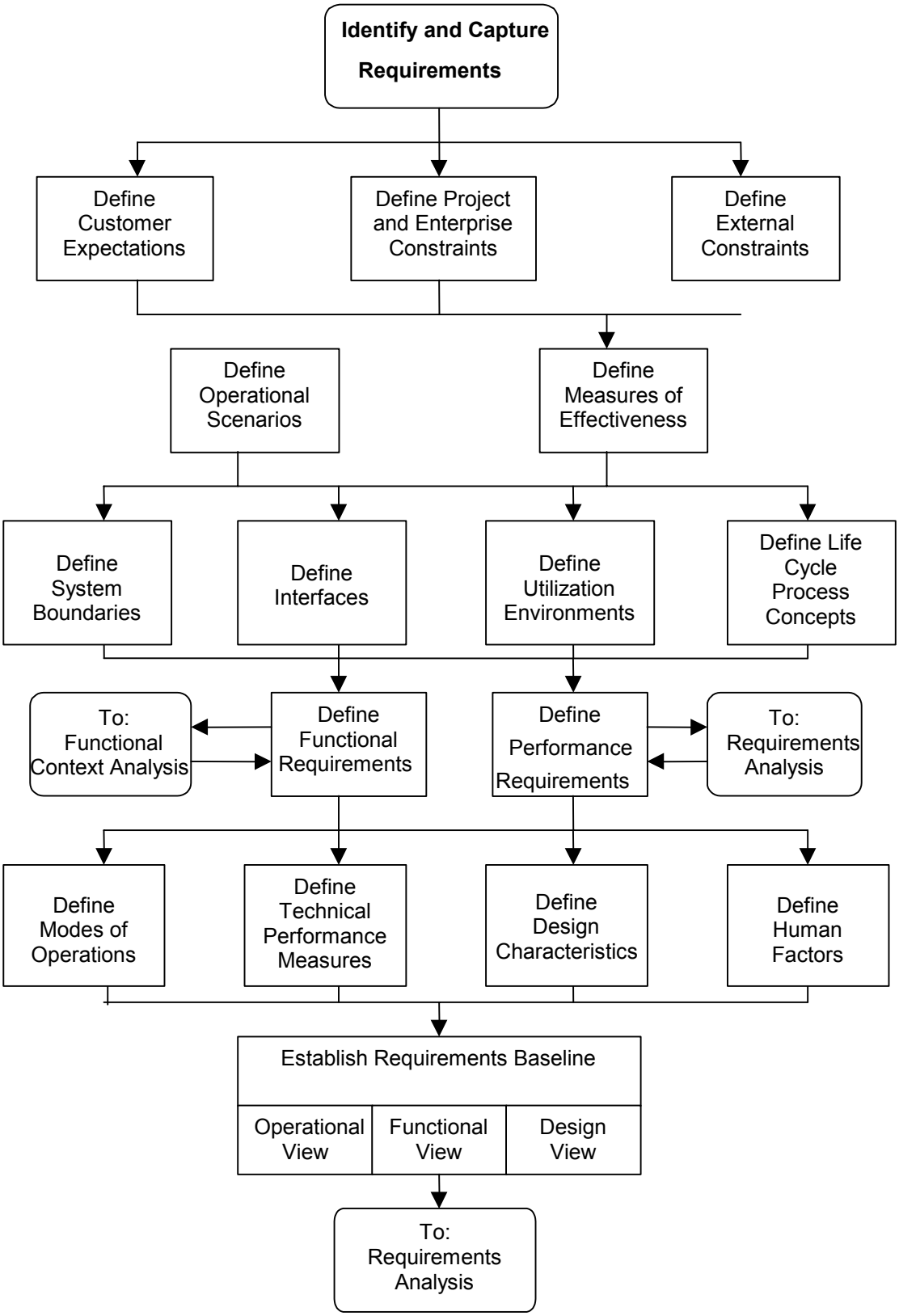
Project constraints include:

- Existing approved specifications and baselines
- Updated NAS Architecture implementation steps
- Updated NAS Architecture segments and mechanisms
- Availability of automated tools
- Required metrics for measuring technical progress

#### 4.3.3.1.5.2.2 Corporate Constraints

Corporate constraints include:

- Management decisions from the Joint Resources Council (JRC) or other management review
- FAA-wide general specifications, standards, handbooks, and guidelines
- FAA policy directives
- Established lifecycle processes
- Physical, financial, and human project resources




522 **Figure 4.3-2. Identify and Capture Requirements Flow**

#### 4.3.3.1.5.3 Task 1.3: Define External Constraints

External constraints that impact design solutions or implementation of SE activities shall be identified and defined. These include:

- U.S. Government and international laws and regulations
- Industry, international, and other general specifications, standards, and guidelines
- ICAO SARPS
- RTCA MOPS and MASPS
- Human-related specifications, standards, and guidelines
- The technology base
- Interfacing systems

#### 4.3.3.1.5.4 Task 1.4: Define Operational Scenarios

Operational scenarios that define the range of the anticipated system uses shall be identified and defined. For each operational scenario, expected interactions with the environment and other systems, human tasks and task sequences, and physical interconnections with interfacing systems and platforms shall be defined.

Data for this step comes from the CONOPS, iRDs and fRDs, and the NAS Architecture.

#### 4.3.3.1.5.5 Task 1.5: Define Measures of Effectiveness

System effectiveness measures that reflect overall stakeholder expectations and satisfaction are defined. Key MOEs may include performance, safety, operability, usability, reliability, maintainability, time and cost to train, workload, human performance requirements, or other factors. Data for this step comes from the CONOPS, iRDs and fRDs, the NAS Architecture, the NAS Requirements, and operational scenarios.

#### 4.3.3.1.5.6 Task 1.6: Define System Boundaries

System boundaries are defined as follows:

- System elements that are under design control and elements that are not
- Expected interactions among system elements under design control and external and/or higher-level and interacting systems outside the system boundary

Data for this step is obtained from any internal, external, policy, or technology constraints; CONOPS; MNS; iRDs and fRDs; and Functional Analysis.

#### 4.3.3.1.5.7 Task 1.7: Define Interfaces

The functional and physical interfaces are defined to external or higher-level and interacting systems, platforms, and/or products in quantitative terms. Functional and physical interfaces may include mechanical, electrical, thermal, data, communication procedural, human-machine, and other interactions required. Interfaces may also be considered from an internal/external perspective. Internal interfaces address elements inside the boundaries established for the

system; they are generally identified and controlled by the contractor responsible for developing the system. External interfaces involve entity relationships outside the established system boundaries.

Data for this step is in IRDs, ICDs, Functional Analysis, MNS, and iRDs and fRDs.

#### 4.3.3.1.5.8 Task 1.8: Define Utilization Environments

Utilization environments for each of the operational scenarios shall be defined. All environmental factors, operational and ambient, that may impact system performance need to be identified and defined. Also identified are factors that ensure that the system minimizes the potential for human or machine errors or for failures that cause accidents or death and that impart minimal risk of death, injury, or acute chronic illness, disability, and/or reduced job performance of the humans who support the system lifecycle. Specifically, weather conditions (e.g., rain, snow, sun, wind, ice, dust, and fog); temperature ranges; topologies (e.g., ocean, mountains, deserts, plains, and vegetation); biological factors (e.g., animal, insects, birds, and fungi); time (e.g., day, night, and dusk); induced factors (e.g., vibration, electromagnetic acoustic, x-ray, and chemical), or other environmental factors are defined for possible locations and conditions conducive to system operation. It is recommended that effects on hardware, software, and humans be assessed for impact on system performance and lifecycle processes.

Data for this step is contained in the OSED, Tradeoff Studies, Specialty Engineering analysis, and FAA and Military Standards, Specifications, and Handbooks.

#### 4.3.3.1.5.9 Task 1.9: Define Lifecycle Process Concepts

The outputs of Tasks 1.1 through 1.8 are analyzed to define lifecycle process requirements necessary to develop, produce, test, distribute, operate, support, train, and dispose of system products being procured.

##### 4.3.3.1.5.9.1 Manpower

The required job tasks and associated workload used to determine the number and mix of humans who support the system lifecycle processes shall be identified and defined.

##### 4.3.3.1.5.9.2 Personnel

The experiences, aptitudes, knowledge, skills, and abilities required to perform the job tasks that are associated with the humans who support the system lifecycle shall be identified and defined.

##### 4.3.3.1.5.9.3 Training

The instruction education and on-the-job or team training necessary to provide humans and teams with knowledge and job skills needed to support the system lifecycle processes at the specified levels of performance are to be identified and developed.

##### 4.3.3.1.5.9.4 Human Engineering

Human cognitive, physical, and sensory characteristics that directly contribute to or constrain lifecycle system performance and that impact human-machine interfaces shall be identified.

#### 4.3.3.1.5.9.5 Safety

The System Safety Engineering analysis derives and identifies requirements that are designed to control the risk of identified safety hazards.

#### 4.3.3.1.5.10 Task 1.10: Define Functional Requirements

Functional requirements for each function of the system as determined by the Functional Analysis process (Section 4.4) shall be defined, describing what the system may be able to do. The functions identified are used in Paragraph 4.3.3.1.5.11 to define how well the functions shall be performed and to establish the performance requirements. The functions identified through Functional Analysis shall be further decomposed during functional decomposition to provide a basis for identifying and assessing design alternatives. All system requirements shall involve a functional and performance aspect, which views system requirements as having both functional and performance aspects that ensure that requirements are complete, consistent, and verifiable.

#### 4.3.3.1.5.11 Task 1.11: Define Performance Requirements

Performance requirements for each system function shall be defined. Performance requirements describe how well functional requirements shall be performed to satisfy the MOEs. These performance requirements are the MOPS that are allocated to subfunctions during functional decomposition analysis and that are the criteria against which design solutions (derived from Synthesis (Section 4.5)) are measured. There are typically several MOPS for each MOE, which bound the acceptable performance envelope.

#### 4.3.3.1.5.12 Task 1.12: Define Modes of Operation

The system modes of operation (e.g., full system, emergency, training, and maintenance) are defined for the system being procured. The conditions (e.g., environmental, configuration, and operation) that determine the modes of operation are defined.

Data for this step shall come from the NAS or system-level CONOPS, MNS, OSED, or Functional Analysis.

#### 4.3.3.1.5.13 Task 1.13: Define Technical Performance Measures

Technical Performance Measures (TPM) are defined that describe the key indicators of system performance. It is recommended that selection of TPMs be limited to critical MOPs that, if not met, put the project at cost, schedule, or performance risk. Specific TPM activities are integrated into the System Engineering Master Schedule to periodically determine achievement to date and to measure progress against a planned value profile.

Data for this step comes from the CONOPS or the MNS.

#### 4.3.3.1.5.14 Task 1.14: Define Design Characteristics

Required design characteristics (e.g., color, texture, size, anthropometrical limitations, weight, and buoyancy) are identified and defined for the system being procured. Design characteristics that are constraints and which may be changed based on tradeoff analysis (Synthesis (Section 4.5)) are identified.

Data for this step comes from the CONOPS, MNS, OSED, Functional Analysis, Tradeoff Studies, and FAA and Military Standards, Specifications, and Handbooks.

#### 4.3.3.1.5.15 Task 1.15: Define Human Factors

Human factor considerations (e.g., design space limits, climatic limits, eye movement, reach ergonomics, cognitive limits, and usability) are identified and defined that affect operation of the system being procured. Human factors that are constraints and may be changed based on tradeoff analysis are identified.

Data for this step comes from the CONOPS, MNS, OSED, Functional Analysis, Tradeoff Studies, Specialty Engineering analysis, and FAA and Military Standards, Specifications, and Handbooks.

#### 4.3.3.1.5.16 Task 1.16: Establish Requirements Baseline

The output of Tasks 1.1 through 1.15 forms a requirements baseline that establishes the characteristics of the system problem to be solved. Three views—operational, functional, and design—are used to define the baseline. The Operational View describes how the system products serve users. It establishes who operates and supports the system and its lifecycle processes and how well and under what conditions the system is to be used. The Functional View describes what the system does to produce the desired behavior described in the Operational View and provides a description of the methodology used to develop the view and decision rationale. The Design View describes the design consideration of the system development and established requirements for technologies and for design interfaces among equipment and among humans and equipment. The content of these views may include the information discussed in the following paragraphs.

##### 4.3.3.1.5.16.1 Operational View

The Operational View addresses how the system serves its users. It is useful when requirements are being established that describe how well and under what condition the system is to be used. It is recommended that Operational View information be documented in an operational concept document that identifies:

- Operational need description
- Results of system operational analyses
- Operational sequences/scenarios, including utilization environments and MOEs and how the system may be used
- Conditions/events to which system products need to respond
- Operational constraints, including MOEs
- Human roles, including job tasks and skill requirements
- Training requirements, including how humans are trained to be a part of the system and support system lifecycle processes through formal, informal, embedded, on-the-job, or other forms of training
- What operations are required to ensure safety

- Lifecycle process concepts, including MOEs, critical MOPS, and already existing products and services
- Operational interfaces with other systems, platforms, humans, and/or products
- System boundaries

#### 4.3.3.1.5.16.2 Functional View

The Functional View focuses on what the system shall do to produce the required operational behavior. It includes required inputs, outputs, states, and transformation rules. The Functional View and the Operational View are the primary sources for the MNS and the requirements documents. The functional requirements, coupled with the design requirements, described in Design View below, are the primary sources of the requirements that may eventually be reflected in the system specification. Functional View information includes:

- Functional requirements that describe what system products and lifecycle processes shall do or accomplish
- Performance requirements, including qualitative (how well), quantitative (how much, capacity), and timeliness or periodicity (how long, how often) requirements
- Functional sequences for accomplishing system objectives
- TPM criteria
- Functional interface requirements with external, higher-level, or interacting systems, platforms, humans, and/or products
- Modes of operations
- Functional capabilities for planned evolutionary growth
- Verification requirements, including inspection, analysis/simulation, demonstration, and test

#### 4.3.3.1.5.16.3 Design View

The Design View focuses on how the system is constructed. It is key to establishing the physical interfaces among operators and equipment and technology requirements. Design View information includes:

- Previously approved specifications and baselines
- Design interfaces with other systems, platforms, humans, and/or products
- Human SE elements, including safety, training, knowledge, skills, and abilities required to accomplish system functions, and characteristics of information displays and operator controls
- Characterization of operator(s) and support personnel, including special design requirements and applicable movement or visual or workload limitations
- Characterization of information displays and operator controls
- System characteristics, including design limitation (e.g., capacity, power, size, weight); technology limitations (e.g., precision, data rates, frequency, language); inherent human limitations (e.g., physical and cognitive workload, perceptual abilities, and reach and

anthropometric limitations); and standardized end items, nondevelopmental items (NDI), and reusability requirements

- Design constraints, including project, corporate, and external constraints, that limit design solutions and/or developmental procedures
- Design capabilities and capacities for planned evolutionary growth

#### 4.3.3.2 Task 2: Analyze and Decompose Requirements

Architecture developed in Functional Analysis (Section 4.4) is translated into PRSs that, in turn, are translated into MRSs in this task.

##### 4.3.3.2.1 Analyze Requirements

The Functional Architecture is the primary input to the Requirements Management process. A Functional Architecture describes what a system shall accomplish. The Functional Architecture is composed of functional flow diagrams (FFD), timeline sequence diagrams, and functional $N^2$ charts described in Functional Analysis (Section 4.4). The Functional Architecture is a living document that increases in level of detail along with the decomposition of requirements. It is recommended that there be a level of Functional Analysis and corresponding Functional Architecture for every level of requirements (Table 4.3-1). The Requirements Management process uses the Functional Architecture to derive PRSs.

The Requirements Management process starts with recognition of a need or shortfall in system capability and progresses in increasing detail:

**Table 4.3-1. Functional Architecture to Requirements Traceability Hierarchy**

| Functional Architecture | Requirements |
|---|---|
| CONOPS → | Mission Need Statement |
| Functional Analysis 1 → | Initial Requirements Document |
| Functional Analysis 2 → | Final Requirements Document |
| Functional Analysis 3 → | System Requirements |
| Functional Analysis N → | System Specification to N level |

The objective of function transformation is to transform functions into the functional and performance PRSs that describe the system attributes that achieve customers' needs.

###### 4.3.3.2.1.1 Function to Requirements Transformation

A Functional Architecture (from Functional Analysis (Section 4.4)) is transformed into PRSs through two fundamental methods: (1) a structured analysis methodology called System Functional Requirements Analysis (SFRA) and (2) Functional Architecture Referencing (FAR).

Regardless of the method used, the result is a set of PRSs associated with the system functions.

#### 4.3.3.2.1.1.1 System Functional Requirements Analysis

SFRA is a structured methodology for developing requirements from a Functional Architecture. It requires building a matrix of functions and system characteristics then assigning a PRS to each function/characteristic pair if one is needed. The following steps produce a list of functions for which PRSs shall be developed.

##### 4.3.3.2.1.1.1.1 List Functions

From the Functional Architecture, the functions in a table, such as the example included in Table 4.3-2, are listed. A tree diagram may be used to assist creation of the function list.

###### 4.3.3.2.1.1.1.1.1 Tree Diagrams

A tree diagram is constructed from the top down. Each subfunction is shown as a branch of the tree. Using the FFD in Figure 4.4-23 as an example, the tree diagram in Figure 4.3-3 was developed as an incomplete example of what the tree diagram might look like. A completed diagram might result in a family tree hierarchy of functions.

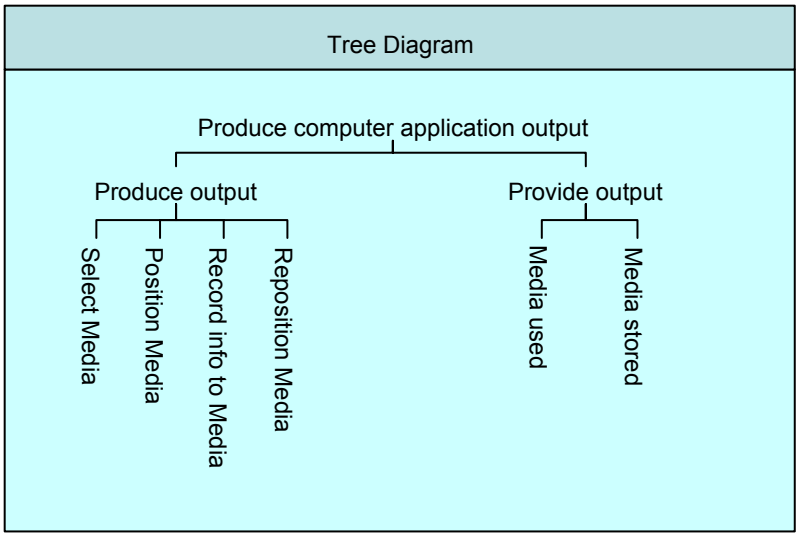



**Figure 4.3-3. Tree Diagram Example**

#### 4.3.3.2.1.1.1.2 Develop System Characteristics

System characteristics are generated by identifying all measurable product characteristics perceived as related to meeting customer requirements. These characteristics come from (1) the external inputs described in Paragraph 4.3.2.1 and (2) analyses conducted in Specialty Engineering (Section 4.8). The characteristics include specialty requirements, constraints, standards, handbooks, management decisions, policies, and legacy requirements and are listed in Table 4.3-2. The specific categories and characteristics are unique to and change with each system. The material shown is for illustration only.

#### 4.3.3.2.1.1.1.3 Determine Intersections

The purpose of this step is to determine if a need exists to translate a particular function into a PRS. If there is a significant relationship between the function and the characteristic, a PRS number is placed in that cell. “Significant” means that it was determined, using engineering judgment, that the function shall have one or more of the related characteristics in order to meet the customer’s need. Wherever there is a number, a unique PRS is required to describe that relationship. The number is associated with the unique PRS that describes the function-characteristic combination.

If it is determined that a function-characteristic combination is not significant or nonexistent, then a PRS is **not** written for that intersection.

#### 4.3.3.2.1.1.1.4 Develop Primitive Requirements Statement

The PRS is developed in accordance with the procedure in Paragraph 4.3.3.2.1.1.3.

### 4.3.3.2.1.1.2 Functional Architecture Reference

This method generates PRSs from the standards, handbooks, and Specialty Engineering analyses. The functional PRSs are developed by referencing the Functional Architecture. Because of the risk of missing critical requirements, it is recommended that this method be used only when there is not enough time to perform SFRA.

#### 4.3.3.2.1.1.2.1 Derive Primitive Requirements Statement From Standard Sources

A list or database of PRSs is developed. The PRSs are derived by using the sources described in Specialty Engineering (Section 4.8) and the inputs listed in Paragraph 4.3.3. The PRSs shall be developed in standard PRS format.

For example, assume that a reliability analysis derived a requirement that states: “Transmitter MTBF greater than 5000 op hours.” The PRS is listed as a requirement in this list. Table 4.3-3 provides an example.

**Table 4.3-2. System Characteristic Matrix**

| Functions | Characteristics | Performance | | Specialty Engineering | | | | Environment | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Accuracy | Thermal | Reliability | Safety | Spectrum | Oprtr workload | Radiation | Lightning | Precipitation |
| Detect ac state vector | Determine aircraft horizontal location | 2 | 1 | | 3 | N | N | N | N | N |
| | Determine aircraft vertical location | N | N | | N | N | N | N | N | N |
| | Determine aircraft velocity vector | N | N | | N | | N | | | |
| Transmit voice RF | Convert sound to high frequency (RF) signal | N | N | N | | N | N | N | N | N |
| | Convert signal to EM wave | N | N | N | N | N | | N | | N |
| | Propagate wave through space-time | | | | | N | | N | N | N |
| Distribute NOTAM | Encode NOTAM | N | N | | N | | N | | | |
| | Determine scope | N | N | | N | | N | | | |
| | Transmit NOTAM | N | N | | N | N | N | N | N | N |

**Table 4.3-3. Primitive Requirement Statements List**

| PRS Number | Primitive Requirement Statement | Functional Reference |
|---|---|---|
| Assign a unique number to the PRS | This is the derived PRS | Assign the PRS to a function in the Functional Architecture |
| 126 | Transmitter MTBF greater than 5000 op hours | F.3.2.1.1 |

#### 4.3.3.2.1.1.2.2 Relate Primitive Requirements Statement to Functional Architecture

The Functional Architecture and existing PRS are reviewed, and each PRS is assigned to a function in the Functional Architecture. Each requirement shall be assigned to a function, and it is recommended that each function have one or more requirements assigned to it.

#### 4.3.3.2.1.1.2.3 Sort the Primitive Requirements Statements by Functional Reference

Requirements shall be sorted or grouped so that grouped and sorted requirements allocated to an individual function are together. Table 4.3-4 is an example.

**Table 4.3-4. Primitive Requirement Statements List**

| PRS Number | Primitive Requirement Statement | Functional Reference |
|---|---|---|
| 126 | Transmitter MTBF greater than 5000 op hours | F.3.2.1.1 |
| 34 | Transmitter EMI hardened greater than 50000 volt-meters | F.3.2.1.1 |
| 212 | Transmitter power less than 10 watts | F.3.2.1.2 |
| 6 | Transmitted power less than or equal to table 4.3 in HERP standard 6. | F.3.2.1.2 |
| 57 | Transmitted power less than or equal to table 2.1 in HERF standard 4.4. | F.3.2.1.2 |

#### 4.3.3.2.1.1.2.4 Write the Functional Primitive Requirements Statement

Once requirements are sorted to functions, the functional PRSs are derived. First, the Functional Architecture used shall be appended to the requirements document. Then, for each group of PRSs, a functional PRS shall be defined in the following manner:

**[Element] functions + as defined in + [Functional Reference (include page and figure number)]**

For the above example table, two functional PRSs are added as shown in Table 4.3-5.

**Table 4.3-5. Grouped and Sorted Primitive Requirement Statements List**

| PRS Number | Primitive Requirement Statement | Functional Reference |
|---|---|---|
| 126 | Transmitter MTBF greater than 5000 op hours | F.3.2.1.1 |
| 34 | Transmitter EMI hardened greater than 50000 volt-meters | F.3.2.1.1 |
| 220 | Transmitter functions as defined in F.3.2.1.1, page A-26, figure A.2.2. | F.3.2.1.1 |
| 212 | Transmitter power less than 10 watts | F.3.2.1.2 |
| 6 | Transmitted power less than or equal to table 4.3 in HERP standard 6. | F.3.2.1.2 |
| 57 | Transmitted power less than or equal to table 2.1 in HERF standard 4.4. | F.3.2.1.2 |
| 221 | Transmitter functions as defined in F.3.2.1.2, page A-28, figure A.2.4. | F.3.2.1.2 |

#### 4.3.3.2.1.1.3 Primitive Requirements Statements

Requirements are first captured as a list of PRSs. A PRS is a primitive form of a requirement statement that has no punctuation or formal sentence structure and is not written in a formal specification style. The PRS form is used at this stage to improve the early requirements identification capability by removing the rigor of writing MRSs from the early concept development and to remove the considerable cost of forming mature requirements. Each PRS is uniquely numbered and follows a simple three-part format:

**Name + Relation + Value**

The name describes the characteristic or attribute to control; the relation details the connection between the attribute and its control value; and the value sets a quantifiable number with units or defines a standard. Numerical requirements use one of six possible relations: less than, greater than, equal to, less than or equal to, greater than or equal to, or between a range of values. For nonnumerical requirements, words such as “is,” “be,” and “conforms to” are used as the relation.

#### 4.3.3.2.1.1.4 Mature Requirements Statement

Once the PRSs at any level are identified, they shall be synthesized into MRSs that satisfy the characteristics and attributes of good requirements. Requirements characteristics are the principal properties of the MRS. Characteristics may apply to individual requirements or to an aggregate of requirements. A well-defined set of MRSs needs to exhibit certain individual and aggregate characteristics. The result of performing this activity shall be a baseline set of requirements that satisfy all of the characteristics described herein and that is recorded and maintained over the lifecycle of the product, as well as accessible to all parties.

The basics of well-defined requirements are clarity, conciseness, and simplicity; elegant, entertaining prose is not needed and is undesirable. This activity describes (1) how to build requirements and (2) the essential characteristics of well-defined requirements.

An MRS is a written statement of a requirement in one or more complete sentences in a familiar language (normally English) using the idiom of a particular business sector, such as air traffic control or avionics. Normal specification standards require that the content of a specification document include complete sentences organized in a particular way. Each requirement statement shall (1) be written in proper grammar, (2) make appropriate use of standard constructs, (3) possess the characteristics and attributes of good requirements, and (4) comply to a specified standard format.

Each PRS shall be converted to specification text. A specification for a system is a published set of requirements that has been properly refined and formatted into more precise language than used for the PRSs. Usually, each PRS becomes a short paragraph when converted into specification text. A primitive requirement is connected into specification text by adding the characteristics described in the following paragraphs.

**4.3.3.2.1.1.4.1 Paragraph Number**

The type of requirements is identified, and a paragraph number is assigned according to the required format. The numbering format may be ad hoc for some requirements documents or shall adhere to a rigid format, such as a Federal Aviation Administration Acquisition System Toolset (FAST) template or FAA-STD-005 or MIL-STD-961.

**4.3.3.2.1.1.4.2 Paragraph Title**

A paragraph title is identified that is linked to the named or controlled PRS attribute.

**4.3.3.2.1.1.4.3 Subject**

The subject of the requirements is the main topic of the sentence and is linked to the named or controlled PRS attribute.

**4.3.3.2.1.1.4.4 Directive Verb**

The directive verb in the requirement sentence directs the action required and shall relate the named or controlled attribute to the value. See Paragraph 4.3.3.2.1.1.5.1.

**4.3.3.2.1.1.4.5 Sentence Ending**

The requirements sentence is ended with a period with a commonly used word or phrase that provides a reference to a standard or specification. See Paragraph 4.3.3.2.1.1.5.2.

**4.3.3.2.1.1.4.6 Explanatory information**

Explanatory, defining, or clarifying information is added after the requirements sentence if necessary to ensure understanding and avoid ambiguity.

**4.3.3.2.1.1.5 Standard Constructs**

Standard constructs are used to record requirements so that they possess the characteristics of good requirements.

#### 4.3.3.2.1.1.5.1 Directive Verbs

All requirements documents shall have directive verbs that denote action, as follows:

- Use the verb "shall" to denote compulsory or mandatory action that the person being directed is obliged to take. (For example: The contractor shall furnish all facilities and equipment necessary for the tests specified herein.)
- Use the verb "may" to denote permission or an option that is not obligatory. (For example: For instruction books of 50 pages or less, multi-ring binding may be employed in lieu of saddle stitching.)
- Use the verb "will" to denote a declaration of purpose on the part of the government. (For example: The Contracting Officer will furnish shipping instructions upon request.)
- The verb "should" is not used in requirements documents. Although the word "should" is used to denote action that is recommended but not obligatory, it may imply duty or obligation in legal usage.

#### 4.3.3.2.1.1.5.2 Commonly Used Words and Phrasings

Certain words and phrases are frequently used in requirements documents. The following rules shall apply:

- Referenced documents requirements are to be written as follows:
  - " …in accordance with Specification (or Standard)…"
  - "…shall be as specified in Specification (or Standard)…"
  - "…shall conform to…
  - "…conforming to Specification (or Standard)…"
- The phase "unless otherwise specified" shall be used to indicate an alternate course of action. The phrase shall come at the beginning of the sentence and, if possible, at the beginning of the paragraph. This phrase shall be limited in its application and used sparingly.
- The term "and/or" shall not be used in requirements documents. The following example conveys the desired meaning: "The panel shall be supported on brackets, pillars, or both."
- Do not use "minimum" and "maximum" to state limits. Use "no less than" or "no greater than." This standard construct avoids the ambiguity associated with the limiting values. This does not mean that the words "minimum" and "maximum" may not be used at all, just not to state limits.

#### 4.3.3.2.1.1.5.3 Words and Phrases To Avoid

It is recommended that specific words and phases be avoided because they are vague, ambiguous, and general, such as "flexible," "fault tolerant," "high fidelity," "adaptable," "rapid" or "fast," "adequate," "user-friendly," "support," "maximize," "minimize," and "shall have the capability to."

#### 4.3.3.2.1.2 Characteristics of Individual Requirements

Characteristics of individual requirements may be used for requirements development as well as in requirements reviews and audits for assessing the quality of requirements. These characteristics are described below with synonyms in parenthesis.

##### 4.3.3.2.1.2.1 Necessary

The stated requirement is an essential capability, characteristic, or quality factor of the product or process. If removed or deleted, it may cause a deficiency that is unable to be fulfilled by other capabilities of the product or process.

This is a primary characteristic, and it shall be exhibited in the requirements statement to effect a well-defined requirement. There is no room in a specification for unnecessary requirements because they add cost to the product. If a necessary requirement is deleted from the specification, a major need may not be met, even if all other requirements are satisfied.

One good test of necessity is traceability to higher-level documentation. In the case of a system specification, traceability may be verified to user documentation, such as the Operational Requirements Document. If there is no parent requirement, the requirement may not be necessary.

##### 4.3.3.2.1.2.2 Concise (Minimal, Understandable)

The requirements statement includes only one requirement that simply and clearly states only what shall be done, making it is easy to read and understand. To be concise, the requirements statements shall not contain any explanations, rationale, definitions, or descriptions of system use, which are used in text analysis and trade study reports, operational concept documents, user manuals, or glossaries. A link may be maintained between the requirements text and the supporting analyses and trade studies in a requirements database so that the rationale and explanations may be referenced.

Determining what constitutes one requirement is a constant struggle in developing requirements and often requires engineering judgment. An example is the requirement in FAA automation systems for a Minimum Safe Altitude Warning (MSAW)/Conflict Alert alarm. This alarm requires an aural alarm and a visual alarm to warn the controller about potential unsafe conditions. Therefore, the question is: Is this one requirement, or does a requirement need to be written for each condition? Multiple requirements in one paragraph is undesirable, as is the proliferation of the number of requirements without reason. Each requirement needs to be managed and verified, and as such, has an associated cost.

One decisionmaking approach to the question is to determine how the requirement is to be verified. In the alarm example, it is recommended to verify that the alarms work together; therefore, any test to verify the alarms shall include both the aural and visual alarms, thus combining the aural and visual alarms into one requirement.

##### 4.3.3.2.1.2.3 Implementation-Free

The requirement states what is required, not how the requirement needs to be met. It is recommended that the requirement state the desired result in functional and performance terms, not in terms of a solution set. It is also recommended that a requirements statement not reflect

a design or implementation nor describe an operation. However, the treatment of interface requirements is generally an exception.

This characteristic of a requirement is perhaps the hardest to judge and implement. At the system level, requirements may be truly abstract or implementation-free. The system requirements have to be synthesized by a system design solution. After a trade study has been conducted between alternatives and a candidate solution has been selected, the system requirements have to be allocated to the elements defined by the system design. This incremental procedure of allocating requirements to the next lower-level elements, which is dependent on system design, leads to the observation that one level of design is the requirement at the next lower level. The conclusion is that a requirement is implementation-free at the level that it is being specified, but is a result of the design activity at the level above it.

Interface requirements are usually an exception to the implementation-free rule. Interface requirements are specified in interface control drawings or ICDs that describe a specific design or an interface or mating part. The interface requirement shall provide complete information so that the two sides of the interface may be designed to work as specified when connected to each other.

**4.3.3.2.1.2.4 Attainable (Achievable or Feasible)**

The stated requirement may be achieved by one or more developed system concepts at a definable cost. This implies that at least a high-level conceptual design has been completed and cost tradeoff studies have been conducted.

This characteristic is a test of practicality of the numerical value or values set forth in a requirement. It signifies that adequate analyses, studies, and trades have been performed to show that the requirement may be satisfied by one or more concepts and that the technology cost associated with the concept(s) are reasonable within program cost constraints.

**4.3.3.2.1.2.5 Complete (Standalone)**

The stated requirement is complete and does not need further amplification and provides sufficient capability.

This characteristic specifies that each requirement be stated simply using complete sentences. It is recommended that each paragraph state everything required on the topic and that the requirement be capable of standing alone when separated from other requirements.

**4.3.3.2.1.2.6 Consistent**

The stated requirement does not contradict other requirements and is not a duplicate of another requirement. The same term is used for the same item in all requirements.

This characteristic of well-defined requirements is usually well understood and does not cause much discussion. However, in a large set of requirements that are not well organized by some clearly defined categories, it may be hard to spot duplications and inconsistencies. Therefore, organizing requirements in accordance with a standard or template is important so that inconsistencies may be identified. It is also important to maintain a glossary of program terms because the meaning of some words are domain-dependent.

#### 4.3.3.2.1.2.7 Traceable

It is recommended that each stated requirement be developed in a way that allows it to be traced back to its source. A requirement also needs to identify related requirements (i.e., parents, children, peers) and requirements that might be impacted by changes to it.

This characteristic contributes to completeness by verifying that all requirements have a source or are allocated. It also helps to eliminate unnecessary or missing requirements.

#### 4.3.3.2.1.2.8 Unambiguous

Each requirement shall have one, and only one, interpretation. Language used in the statement shall leave no doubt as to the intended descriptive or numeric value.

This characteristic is difficult to achieve because the English language may be unstructured and, in some cases, the same sentence may mean different things to different people. It is helpful to use standard specification language constructs and commonly used words and phases and to avoid using the commonly used words and phrases cited in Paragraph 4.3.3.2.1.1.5.3.

#### 4.3.3.2.1.2.9 Verifiable

Each requirement shall have an identified means by which to verify that it meets the characteristics established above.

The stated requirement is not vague or general but is quantified in a manner that may be verified by one of the verification methods described in Validation and Verification (Section 4.12).

The characteristic of verifiability needs to be considered at the same time that a requirement is being defined. A requirement that is not verifiable is a problem because it involves the acceptability of the system. To be verifiable, a requirement shall be stated in measurable terms.

#### 4.3.3.2.1.2.10 Allocatable

It is recommended that the stated requirement be allocated to component(s) within the requirements hierarchy or assigned to an organization.

This characteristic is important because it helps to eliminate requirements that are not complete, concise, and clear. If a requirement is not allocatable to the Physical Architecture, it is probably not necessary.

#### 4.3.3.2.1.3 Characteristics of Aggregate Requirements

Aggregate requirements are a set of requirements for a system or element that specifies its characteristics in totality. Usually, these aggregates are found in specifications or Statements of Work (SOW). Characteristics of individual requirements also are applicable to aggregates.

#### 4.3.3.2.1.3.1 Complete

The set of requirements is complete and does not need further amplification. The set of requirements has addressed all categories (Paragraph 4.3.3.2.1.4.3) of requirements and covers all allocations from higher levels.

This characteristic addresses the difficult problem of identifying requirements that are necessary but are missing from the requirements set. One approach to identify missing requirements is to walk through the Operational Concept and its associated scenarios from start to finish, then walk through the same set of scenarios and ask “what if” questions. This approach usually uncovers a new set of requirements. A second approach is to develop a checklist of topics or areas, such as a specification outline, and verify that requirements exist in each topic area or, if they do not exist, that there is a good reason for it. A third approach is to check the aggregate requirements set against a higher-level document (if one exists) to verify that all allocated requirements have been included in the set.

#### 4.3.3.2.1.3.2 Consistent

The set of requirements has no individual requirements that are contradictory. Requirements are not duplicated, and the same term is used for the same item in all requirements.

This characteristic addresses the problem of identifying unnecessary or conflicting requirements that are inadvertently included in the set. Assigning program-unique identification to each requirement and conducting thorough reviews are ways to eliminate these requirements.

#### 4.3.3.2.1.4 Attributes of Requirements

This section describes secondary properties or attributes of individual requirements that provide supplementary information about the requirement and its relationship to other requirements and source documents. The properties or attributes also assist in requirements management. However, these attributes are not essential in all cases.

#### 4.3.3.2.1.4.1 Requirement Identification

Each requirement is assigned a program-unique identifier (PUI) for identification and tracking purposes. The PUI may be either numeric or alphanumeric and assigned automatically if a requirements management tool is used. The requirement identifier assists in identifying the requirement, maintaining change history, and providing traceability.

#### 4.3.3.2.1.4.2 Level

This attribute indicates the level at which the specific requirement is applicable in the system hierarchy or Work Breakdown Structure (WBS). A level I requirement may indicate a top- or system-level requirement; a level II requirement may be a segment- or component-level requirement.

#### 4.3.3.2.1.4.3 Requirements Category

Two categories are used to classify requirements: program and technical.

#### 4.3.3.2.1.4.3.1 Program Requirements

Program requirements are stakeholder or user requirements imposed on vendors through contractual vehicles other than specifications, including the contract or contract SOW. Program requirements include:

- Compliance with Federal, State, or local laws, including environmental laws
- Administrative requirements (e.g., security); stakeholder/vendor relationship requirements (e.g., directives to use government facilities for specific types of work such as test); and specific work directives (e.g., directives included in the SOW and Contract Data Requirements List (CDRL))

Program requirements may also be imposed on a program by agency policy, directives, or practice.

Program requirements are different from technical requirements: They are not imposed on the system or product to be delivered but on the process to be followed by the program. Program requirements, which are managed similarly to technical requirements, need to be necessary, concise, attainable, complete, consistent, and unambiguous in the same manner as technical requirements.

#### 4.3.3.2.1.4.3.2 Technical Requirements

Technical requirements are applicable to the system or service to be procured. Technical requirements are described in requirement documents, system specifications, and interface documentation. Types of technical requirements are described in the following paragraphs.

##### 4.3.3.2.1.4.3.2.1 Stakeholder Requirements

Stakeholder requirements are associated with the stakeholder's intended operating practices, maintenance concepts, and desired features.

##### 4.3.3.2.1.4.3.2.2 Operational Requirements

Operational requirements define the interfaces between the end-user and each functional system, maintenance concept and each system, and various other support and related functions or equipment.

##### 4.3.3.2.1.4.3.2.3 Performance Requirement

Performance requirements define how well the product performs its intended function (e.g., accuracy, fidelity, range, resolution, and response times).

##### 4.3.3.2.1.4.3.2.4 Functional Requirements

Functional requirements identify what the system may do, not how the system accomplishes it. They are based on Functional Analysis (Section 4.4).

#### 4.3.3.2.1.4.3.2.5 Interface Requirements

Interface requirements are the physical and functional requirements associated with the product interfaces (boundary conditions). Interface development is described in Interface Management (Section 4.7).

#### 4.3.3.2.1.4.3.2.6 Constraint Requirements

Constraint requirements are limitations or restrictions that bound the solution set.

#### 4.3.3.2.1.4.3.2.7 Regulatory Requirements

Regulatory requirements are imposed by statutes or regulations (e.g., FARs, Occupational Safety and Health Administration (OSHA) regulations, and Environmental Protection Agency (EPA) directives).

#### 4.3.3.2.1.4.3.2.8 Reliability, Maintainability, and Availability/Supportability

Reliability, maintainability, and availability/supportability requirements are based on the user's system readiness and mission performance requirements, physical environments, and resources (e.g., personnel, training, and facilities) available to support the mission.

#### 4.3.3.2.1.4.3.2.9 Safety Requirements

These requirements are defined to control the effects of failure conditions, hazards, and/or safety-related functions.

#### 4.3.3.2.1.4.3.2.10 Human Engineering Requirements

Human requirements define the human system interface(s).

#### 4.3.3.2.1.4.3.2.11 Producibility Requirements

Producibility requirements define the producibility of a product that involve identifying materials, special tools, test equipment, facilities, personnel, and procedures. They identify the manufacturing technology needs, availability of critical materials, long-lead procurement requirements, and manufacturing test requirements, among other aspects.

#### 4.3.3.2.1.4.3.2.12 Cost Requirements

Cost requirements define product budget constraints.

### 4.3.3.2.2 Decompose Requirements

The requirements may be decomposed to the lowest level and partitioned in such a way that integrating the partitioned requirements shall satisfy the higher-level requirement.

### 4.3.3.2.3 Checklist for Writing and Evaluating Requirements

The following guidelines for writing and evaluating requirements contain representative questions; however, the list is not intended to be complete and comprehensive.

#### 4.3.3.2.3.1 Technical Considerations

- Does the requirement state a valid need?
- Is the requirement verifiable?
- Has the verification approach been identified?
- Are the necessary interface requirements stated?
- Are appropriate data (e.g., tables, figures) included?
- Are the stated references clearly applicable to the requirement?
- Is the requirement within the span of knowledge of the requirement owner?
- Does the requirement have stated values for quantities?
- Are words that imply a design avoided?

#### 4.3.3.2.3.2 Traceability Considerations

- Are the applicable parent, child, and peer requirements identified?
- Are the source and rationale for the existence of the requirement documented?
- Is the basis for allocation identified?

#### 4.3.3.2.3.3 Writing Considerations

- Is the requirement stated as a requirement?
- Is the requirement stated clearly and concisely?
- Does the requirement represent only one thought?
- Is the requirement stated positively?
- Is the requirement void of ambiguous terminology?
- Is the requirement grammatically correct?
- Is the requirement punctuated correctly?
- Is excessive punctuation avoided?

### 4.3.3.3 Task 3: Allocate Requirements

#### 4.3.3.3.1 Allocation

The Allocate Requirements activity allocates or assigns requirements to system, personnel, or support activity components and/or appropriate organizational entities. This process verifies that the performance and verification requirements are correct and complete at each level before further allocation and decomposition, and it verifies them regarding feasibility and top-level design concept before allocation to software. The allocated requirements consist of all requirements, including the breakdown/decomposition of physical characteristics, functions, cost, schedule, reliability/maintainability parameters, and performance parameters. Mapping of these requirements identifies the owner that has Responsibility, Authority, and Accountability (RAA) for the respective requirement.

#### 4.3.3.3.2 Application

The Allocate Requirements activity is applied iteratively when new, changed, or derived requirements are generated. One cycle through the Allocate Requirements activity is complete when the currently identified requirements have been accurately allocated to the appropriate system, personnel, or support activity component(s). Subsequent analyses, requirement decomposition, and trade studies may produce additional requirements that define the most balanced requirements allocation for the product. When a system-level requirement is allocated to more than one configuration item, the allocation process ensures that the lower-level requirements, when taken together, satisfy the system requirements.

#### 4.3.3.3.3 Allocation Hierarchy

Typically, the requirements are allocated to components of the system hierarchy defined in the Physical Architecture provided by the Synthesis process (Section 4.5). System requirements (including test and verification requirements) are analyzed, refined, and decomposed to ensure complete functional allocation to system, personnel, or support activity components. When a system-level requirement is allocated to more than one configuration item, a process is used to ensure that the lower-level requirements, when taken together, satisfy the system-level requirement. Early allocations only designate high-level product components, as a complete design may not have been determined. As the product design matures, the identified requirements may be allocated to lower-level components in the Physical Architecture. The requirements documents below the system level are simply documents containing the requirements that have been allocated to particular product component(s). As requirements are identified and allocated at different levels of the product hierarchy, the requirements documents may be produced and formatted to fit the need at that particular level. As the requirements and system hierarchy are iteratively defined to lower levels, each requirement ultimately shall be allocated to the lowest possible level of the system component. The results of the allocation process are documented in the Requirements Allocation Matrix (RAM) described in Paragraph 4.3.4.1.1.3.

#### 4.3.3.3.4 Hardware/Software Allocation

The requirements allocation process allocates design requirements to hardware and software. Software, hardware, and interface specifications are analyzed and refined to ensure that all requirements allocated to software and hardware are adequately addressed and that they do not include inappropriate levels of design details. Occasionally, requirements are derived from software requirements; these requirements are documented and maintained. In addition to allocating requirements to system elements, the process allocates requirements to incremental blocks and builds. The process establishes functional, performance, and verification requirements for each incremental system or software block or build.

#### 4.3.3.3.5 Allocation Program Responsibility

Although SE does not establish program organization, the program organization shall contain elements responsible for allocating requirements and deriving design from the system specification to the software and hardware configuration items.

#### 4.3.3.4 Task 4: Derive Requirements

##### 4.3.3.4.1 Identify Derived Requirements

The objective of requirements derivation is to identify and express requirements that result from considering functional analysis, higher-level requirements, constraints, or processes. This results in additional clarification or amplification of higher-level requirements. These derived requirements need to be stated in measurable parameters at increasingly lower levels within the product hierarchy. Derived requirements may result from, but are not limited to:

- Regulatory policies, program policies, agency practices, and supplier capabilities.
- Environmental and safety constraints; the process translates and traces safety-specific system requirements into the software and hardware requirements baseline. Safety program requirements are also reflected in organizational standards and procedures. The process translates and traces safety-specific requirements into the system (hardware and software) baseline. The process assesses system safety program requirement tasks for applicability and incorporation into organizational standards and procedures.
- Architecture choices for performing specific system functions.
- Design decisions.
- Hardware-software interfaces not already specified in the baseline interface documentation.
- Establishment of detailed requirement values and tolerances (i.e., minimum, maximum, goal, threshold).

Impacts of derived requirements need to be analyzed progressively in all directions (parent, child, and peer) until it is determined that no additional impact is propagated. During this process, the hardware and software architecture design is reviewed for flexibility to adapt to new system requirements.

##### 4.3.3.4.2 Capture Derived Requirements

Derived requirements are captured and treated in a manner consistent with other requirements applicable during the development stage. This activity, like overall SE, is an iterative operation, constantly refining and identifying new requirements as the product concept develops and additional details are defined. As part of the requirements derivation process, areas of the system with volatile requirements are monitored, and requirements specifications are reviewed for ambiguities with the potential of causing software sizing and timing instability and other program impacts.

#### 4.3.3.5 Task 5: Establish Verification Methodology

In this step, a verification approach is developed for each requirement documented in the Validation Table, and the Validation Table is transformed into a VRTM. The strategy or method used to verify each requirement is specified in a Verification Requirement, and the Verification Requirements are listed in the VRTM. The VRTM defines how each requirement is to be verified, the stage in which verification is to occur, and the applicable verification levels. The verification approaches are:

- Inspection
- Analysis
- Demonstration
- Test

These methods are discussed in Validation and Verification (Section 4.12). Figure 4.12-2 is an example of a VRTM. Specific guidelines for the VRTM are included in the Test and Evaluation section of the FAST (http://fast.faa.gov/toolsets/index.htm).

#### 4.3.3.6 Task 6: Manage Requirements Changes

This activity manages and controls requirements throughout the product's lifecycle (before and after instituting formal configuration control) by means of a defined change process. The activity identifies and controls all issues and decisions, action items, formal and informal stakeholder/program management desires/directives, and any other real or potential changes to the requirements. The activity is invoked when a new requirement is identified or a change occurs during any other activity within the Requirements Management process. The activity is a project-wide, approved approach that documents and controls the identified requirement, its appropriate attributes, its relationship(s) to other requirements, and allocation to the product of functional and/or verification hierarchies. The activity ensures that all involved stakeholders concur with the baselined requirements and any changes. The change process controls the allocation of requirements between hardware and software. This activity shall be conducted in conjunction with the Configuration Management process (Section 4.11).

This process accounts for changes to Government-Furnished Equipment (GFE) and Contractor-Furnished Equipment (CFE) safety critical items that impact development efforts. The process also accounts for changes resulting from the Verification process (Section 4.12). That is, if a test or other form of verification determines that a change in requirements is necessary, the process ensures that the change process is initiated to accomplish that change. The steps described in the following paragraphs are performed.

##### 4.3.3.6.1 Identification

A new requirement or a change to an existing requirement is identified. The originator documents the new requirement or change to an existing requirement by providing, at minimum, the following information to the requirements analysis team:

- Statement of the requirement.
- Justification/rationale (e.g., trade study, documentation).
- Traceability, if applicable, to the parent child and/or peer requirements(s). Two-way traceability between the software requirements and the system requirements is established and maintained.
- List of other elements (e.g., physical or functional hierarchies) impacted. For example, whenever requirements change, there is a review of and an update to the hardware and software architecture design. This process ensures that the software impact for each proposed change is addressed. Software artifacts (e.g., requirements, design, code, and documentation), for example, are revised as changes to the requirements are

incorporated. In addition, software development plans and program baselines (e.g., cost and schedule) are reviewed and modified if necessary.

- Change requests and problem reports for all configuration items or units are initiated, recorded, reviewed, approved, and tracked.

#### 4.3.3.6.2 Control

The requirements analysis team prepares and disseminates a requirements change notification as follows:

- Assign due date
- Collect and resolve conflicting responses—if not received, assume acceptance
- Place on decision authority agenda
- Present to appropriate decision authority and record the disposition

Multiple approval levels may be established, depending on management methodology, size, or project phase. If concurrence is not reached, the requirement shall be elevated to the next higher-level review board or decision authority; that is:

- Project Configuration Control Board (CCB)—Changes that impact only the project products
- Program CCB—Changes that impact projects outside of individual projects
- NAS CCB—Changes that are NAS-wide in scope or affect NAS-level requirements or architecture

#### 4.3.3.6.3 Status Accounting

The disposition is recorded and the decision is disseminated to the involved stakeholders. At the program and NAS level, a Configuration Control Decision shall be issued. Otherwise, the project issues new/revised requirements document(s), Specification Change Notices (SCN), requirements verification document(s), and compliance report(s), as appropriate.

### 4.3.4 Outputs of Requirements Management

#### 4.3.4.1 External Outputs

#### 4.3.4.1.1 Requirements

##### 4.3.4.1.1.1 Requirements Documents

The term “requirements documents” refers to any media that record requirements, either in hard copy or electronic form. It is a basic rule that all requirements shall be recorded, including internally generated requirements as well as those generated external to the project. The process does not allow verbal or unwritten requirements.

#### 4.3.4.1.1.1.1 Stakeholder Requirements Documents

Standard requirements documents from an FAA stakeholder include the MNS, the iRD, and the fRD. Other organizations use the Operational Requirements Document (ORD) to communicate requirements. Stakeholders convey requirements through memoranda and other media.

#### 4.3.4.1.1.1.2 Specifications

Specifications are a standard form of requirements documents. The technical requirements for a system and its elements are documented through a series of specifications as described in this manual. FAA-STD-005, “Preparation of Specifications, Standards and Handbooks,” describes the requirements for preparing FAA specifications, standards, and handbooks. MIL-STD-961 is the current standard format for FAA specifications required by FAA-STD-005. FAA specifications were prepared in the MIL-STD-490 format until recently, and some legacy specifications remain in that format.

##### 4.3.4.1.1.1.2.1 Types of Specifications

The System Specification (Type A) is the single most important engineering design document, defining the system functional baseline and including the results from the needs analysis, feasibility analysis, operational requirements and the maintenance concept, top-level functional analysis, and the critical TPMs. This top-level specification leads to one or more subordinate specifications covering applicable subsystems, configuration items, equipment, software, and other system components. Although the individual specifications for a given program may assume a different set of designations, a generic approach is used here.

###### 4.3.4.1.1.1.2.1.1 System Specification (Type A)

Type A Specification includes the technical, performance, operational, and support characteristics for the system as an entity. It includes allocation of requirements of functional areas, and it defines the various functional-area interfaces. The information derived from the feasibility analysis, operational requirements, maintenance concept, and functional analysis is covered.

The System Specification shall provide the technical baseline for the system as an entity, shall be written in performance-related terms, and shall describe design requirements in terms of whats, including the functions that the system is to perform and the associated metrics.

The System Specification is the requirements document used by the FAA to procure most systems. It is placed under configuration management before the system Request for Proposal (RFP) is issued.

###### 4.3.4.1.1.1.2.1.2 Development Specification (Type B)

Type B Specification includes the technical requirements for any item below the system level where research, design, and development are accomplished. This may cover an equipment item, assembly, computer program, facility, or critical item of support. Each specification shall include the performance, effectiveness, and support characteristics that are required in evolving design from the system level down.

The Development Specification is usually produced by a system vendor in response to the

FAA-developed System Specification. It is placed under configuration management at completion of the Preliminary Design Review (PDR).

#### 4.3.4.1.1.1.2.1.3 Product Specification (Type C)

Type C Specification includes the technical requirements for any item below the top system level that is currently in the inventory and may be procured off the shelf. This may cover standard system components (e.g., equipment, assemblies, units, cables), a specific computer program, a spare part, or a tool.The Product Specification is usually produced by a system vendor in response to the
FAA-developed System Specification or to a vendor-developed Development Specification. It is placed under configuration management at completion of the PDR.

#### 4.3.4.1.1.1.2.1.4 Process Specification (Type D) (Rarely Used in Federal Aviation Administration Procurements)

Type D Specification includes the technical requirements that cover a service that is performed on any component of the system (e.g., machining, bending, welding, plating, heat treating, sanding, marking packing, and processing).

The Process Specification is usually produced by a system vendor in response to the FAA-developed System Specification. It is created by the vendor and is rarely used in FAA procurements.

#### 4.3.4.1.1.1.2.1.5 Material Specification (Type E) (Rarely Used in Federal Aviation Administration Procurements)

Type E Specification includes the technical requirements that pertain to raw materials, mixtures (e.g., paints, chemical compounds), or sem-fabricated materials (e.g., electrical cable, piping) that are used in the fabrication of a product.

The Material Specification is usually produced by a system vendor in response to the FAA-developed System Specification. It is created by the vendor and is rarely used in FAA procurements.

#### 4.3.4.1.1.2 Requirements Change Notices

An SCN is a formal document specifying that a baselined document has been changed.

#### 4.3.4.1.1.3 Requirements Allocation Matrix

The RAM allocates requirements to components and assigns responsibilities to organizations. Normally, a requirements management tool, such as Dynamic Object Oriented Requirement System (DOORS), is used for this purpose. A RAM contains the following data:

- Text-based requirement.
- Detailed source of the requirement (i.e., person, document and paragraph number).
- Assigned team(s).
- Traceable parent and/or child requirements. Two-way traceability between the design and the requirements is established and maintained. In addition, when software is

reviewed against the design, two-way traceability between the software code and design is established and maintained. Two-way requirements traceability is maintained from system specification to hardware and software configuration item specifications.

- Date of inclusion or deletion.
- Reference WBS number.
- Requirements verification method (i.e., test, analysis, inspection, demonstration).
- Allocated cost estimate, if any.
- Any CDRL item(s) associated with the requirement.

**4.3.4.1.1.4 Requirements Database**

Although requirements are normally provided in the hard-copy formats described above, they are also available in the original electronic format in automated tools such as DOORS.

**4.3.4.1.2 Requirements Verification Compliance Document**

The RVCD is output to program and project management for program control activities.

**4.3.4.1.3 Verification Requirements Traceability Matrix**

The VRTM is included as a part of every requirement and specification document. It provides information on the verification and traceability from a requirement to a higher-level requirement or to its ultimate source. Validation and Verification (Section 4.12) provides more information on this topic.

**4.3.4.2 Internal Outputs**

Internal outputs are products that are provided to other SE processes.

**4.3.4.2.1 Technical Planning**

**4.3.4.2.1.1 Planning Criteria**

Planning criteria describing planned activities for the Requirements Management process are output to the Integrated Technical Planning process (Section 4.2).

**4.3.4.2.2 Functional Analysis**

**4.3.4.2.2.1 Mission Need Statement**

The MNS is output to Functional Analysis (Section 4.4) for use as the baseline for developing the next lower-level Functional Architecture that is then used by the Requirements Management process to develop the next lower-level requirements.

**4.3.4.2.2.2 Requirements**

The requirements set at any stage in the requirements development process are output to the Functional Analysis process (Section 4.4) for developing the next lower-level functional analysis.

#### 4.3.4.2.3 Synthesis

##### 4.3.4.2.3.1 Requirements

The requirements set below the MNS are output to the Synthesis process (Section 4.5), which allocates requirements to the Physical Architecture.

#### 4.3.4.2.4 Trade Studies

##### 4.3.4.2.4.1 Requirements

During the Requirements Development process, alternative solutions may be proposed that require analysis by conducting trade studies. The Requirements Management process provides output requirements for analysis to the Trades Studies process (Section 4.6).

##### 4.3.4.2.4.2 Constraints

Constraints that are developed during the Identify and Capture Requirements task may be used in a trade study and are output to the Trade Studies process (Section 4.6) in addition to requirements.

#### 4.3.4.2.5 Interface Management

##### 4.3.4.2.5.1 Mission Need Statement

The MNS is provided to the Interface Management process (Section 4.7) so that functional and physical interfaces may be identified and placed under management.

##### 4.3.4.2.5.2 Requirements

Requirements are provided to the Interface Management process (Section 4.7) at all stages of requirements development so that interfaces are identified and controlled.

#### 4.3.4.2.6 Specialty Engineering

##### 4.3.4.2.6.1 Requirements

To perform Specialty Engineering analyses, the system under study shall be described. Requirements are a key component of any description, and they are an output to Specialty Engineering (Section 4.8).

#### 4.3.4.2.7 Integrity of Analysis

##### 4.3.4.2.7.1 Tools/Analysis Requirements

Requirements for tools or analysis that are needed during the Requirements Management process are output to the Integrity of Analysis process (Section 4.9) so that Analysis Criteria may be developed.

##### 4.3.4.2.7.2 Requirements

Requirements are output to the Integrity of Analysis process (Section 4.9).

#### 4.3.4.2.8 Risk Management

##### 4.3.4.2.8.1 Concerns/Issues

Concerns/Issues related to accomplishing the mission objectives and satisfying Stakeholder Needs that are discovered during the Requirements Management process are provided to the Risk Management process (Section 4.10) for review and resolution.

The cumulative status of requirements as a result of previous requirements reviews regarding coverage, balance, mutual conflicts, induced constraints, and so forth are analyzed, and Concerns/Issues are identified.

In the course of performing SE, it is possible that potential requirements management problems may surface in the form of Concerns/Issues. These Concerns/Issues may take many forms, but, for the most part, they may be potential risks to the program.

##### 4.3.4.2.8.2 Requirements

The Requirements Management process identifies requirements to Risk Management (Section 4.10) that are to be analyzed for potential risk.

#### 4.3.4.2.9 Configuration Management

##### 4.3.4.2.9.1 Requirements

The Requirements Management process identifies requirements to the Configuration Management process (Section 4.11) that are to be controlled.

#### 4.3.4.2.10 Validation

##### 4.3.4.2.10.1 Requirements

Requirements developed through the Requirements Management process are to be submitted to the Validation process (Section 4.12) to determine if they are complete, concise, and necessary.

#### 4.3.4.2.11 Verification

##### 4.3.4.2.11.1 Verification Requirements Traceability Matrix

The Requirements Management process expands the Validation Table into a VRTM with assigned verification methods and submits the VRTM to the Verification process (Section 4.12).

##### 4.3.4.2.11.2 Requirements

The Requirements Management process submits requirements to be verified to the Verification process (Section 4.12).

### 4.3.5 Requirements Management Process Metrics

Performance of this process is measured and recorded on a regular basis. The following metrics, at minimum, may be used to evaluate process performance:

- Number of requirements, including both stakeholder-specified and project-derived
- Number of changed requirements, including both stakeholder or project-initiated
- Technology requirements, including proven, to be defined, and unknown technology
- Unclear, undefined, or ambiguous requirements
- Cycle time from requirement change initiation to decision
- Cycle time from change decision to baseline incorporation
- Percent of validated requirements to total proposed requirements

### 4.3.6 Automated Tools for Requirements Management

Use of an automated requirements tool for documenting requirements and related information depends on a variety of factors (e.g., size and complexity of the program, number of requirements, budget). There are multiple automated software tools in the marketplace that adequately store and retrieve the requirements and their traceability. A program's tool shall be capable of maintaining two-way traceability, from system specifications to hardware and software configuration item specifications. It shall be capable of being integrated into an overall SE tool suite so that data are seamlessly portable between applications.

For small programs, a spreadsheet may be more than adequate to document and control the requirements set. As a program grows and becomes more complex, a tool designed for requirements management may be necessary. The primary requirements tool used by the FAA and many of the FAA's systems vendors is DOORS.

#### 4.3.6.1 Requirements Database Accessibility

The requirements information shall be accessible by all program personnel. This may be accomplished by allowing user access to the database itself or by providing availability to the documentation out of the database. A program decision shall be made concerning the availability and changeability of the requirements data. All personnel may be trained in using the requirements management tool or database, or a select group may manipulate the database and use a distribution media (e.g., intranet Web site, paper) to disseminate the information and collect comments and changes.

#### 4.3.6.2 Requirements Tool Characteristics

It is recommended that the database be capable of identifying (i.e., in the form of attributes and relationships) and presenting (e.g., internal queries, standard and project-unique reports) the following types of information:

- Requirements documentation—statements of the requirements, status, requirement type, rationale, and history (including data configuration control) regarding each requirement, and the ability to present the requirements in an appropriate user-defined format (e.g., requirement documents, specifications)
- Traceability—linking requirements to their parent, child, and peer requirements, resulting in user-defined requirement traceability matrices
- Allocation—linking requirements to the product hierarchy, resulting in user-defined requirements allocation documents

- Verification—linking the requirement to specific verification approach attributes, resulting in requirements verification and compliance documents
- Traceability Impact Assessment—ability to assess the impact of proposed changes to the requirement, product, and verification hierarchies
- Compatibility—ability to communicate (minimum of import and export capabilities) with other automated tools

### 4.3.7 References


1. Blanchard, B. "System Engineering Management." (2nd ed.). John Wiley & Sons, Inc: New York, NY.
2. Defense Systems Management College. "Systems Engineering Fundamentals." Defense Systems Management College Press: Fort Belvoir, VA, 1999.
3. Hooks, Ivy. "Guide for Managing and Writing Requirements." Compliance Automation, Inc: Houston, TX.
4. "IEEE Standard for Application and Management of the Systems Engineering Process." The Institute of Electrical and Electronic Engineers, New York, NY.
5. Kar, Philip, and Bailey, Michelle. "Characteristics of Good Requirements." Compliance Automation, Inc: Houston, TX.
6. MIL-STD-961D. 22 March 1995. "Department of Defense Standard Practice for Defense Specifications."
7. Mission Needs Statement.
8. National Airspace System Architecture.
9. National Airspace System Concept of Operations. Version 2.
10. O'Grady, Jeffrey. "System Requirements Analysis." McGraw Hill: New York, NY, 1993.
11. Requirements Documents.
12. Sage, A., and Rouse, W. "Handbook of Systems Engineering and Management." John Wiley & Sons, Inc: New York, NY.
13. Young, R. "Effective Requirements Practices." Addison-Wesley: New York, NY.

## 4.4. Functional Analysis (Meets EIA/IS 731 FA 1.2 and iCMM PA 4)

Functional Analysis details the use of functional flow diagramming as a representative structured analysis process that is the preferred approach of the Federal Aviation Administration (FAA). In addition, this section covers several alternative approaches, as FAA system engineers come in contact with organizations that apply techniques other than functional flow diagramming. Therefore, it is necessary that the engineers be able to communicate with members of those organizations and integrate their results with the work performed by other organizations. The following paragraphs detail functional flow diagramming; alternative approaches appropriate for systems and hardware; alternative models for problems to be solved with computer software; and references that cover these techniques in more depth.

### 4.4.1. Introduction to Functional Analysis

The process of analyzing functions provides System Engineering (SE) with a functional system description that becomes a framework for developing requirements and physical architectures. Utilizing the Functional Analysis process significantly improves synthesis of design, innovation, requirements development, and integration. The Functional Analysis process provides two key benefits to SE: (1) it discourages single-point solutions, and (2) it describes the behaviors that lead to requirements and physical architectures. The essential elements of Functional Analysis are illustrated in Figure 4.4-1, which lists the key inputs necessary to initiate the task, providers, process tasks, outputs required, and customers of process outputs. The beginning and ending boundary tasks, as well as the intermediate tasks, are described in later paragraphs.

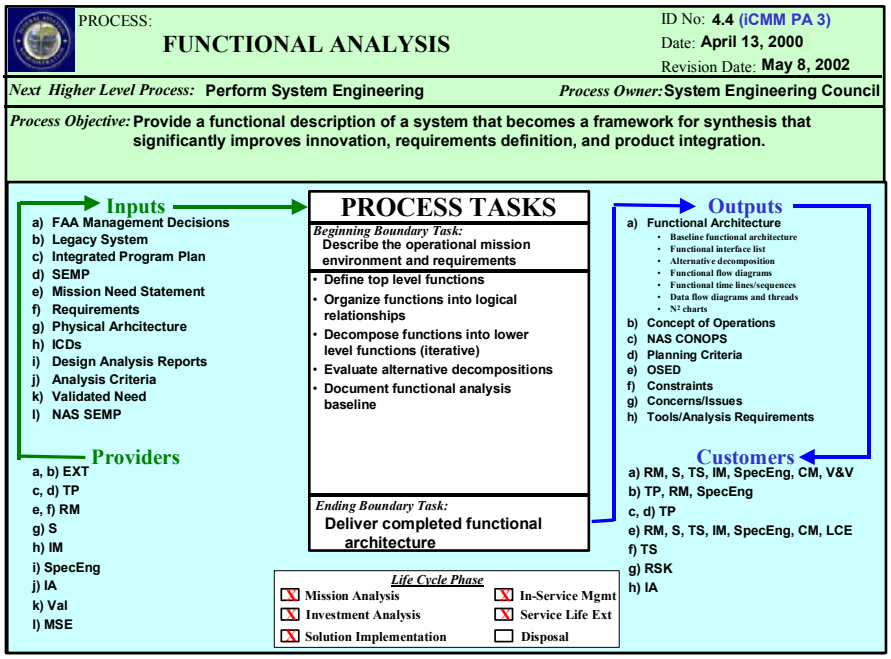



**Figure 4.4-1. Functional Analysis Process-Based Management Chart**

Systems may be described using two different facets. First, a system may be described as a physical architecture with elements that interact with themselves and the system environment in accordance with a predefined process to achieve the system mission. At the same time, a system may be described by the functions that it performs. A system is intended to satisfy predefined functions, with the highest-level function defined as the stakeholder need (also the

ultimate system requirements). A *function* is a characteristic action or activity that has to be performed in order to achieve a desired system objective (or stakeholder need). A *function name* is stated in the form of an action verb followed by a noun or noun phrase; it is an action that describes the desired system behavior. Examples of common functions include "read book," "eat food," and "go to store." A function is accomplished by one or more system elements composed of equipment (hardware, software, and firmware), people, and procedures. The function occurs within the system environment and is performed to achieve system operations. In Functional Analysis, because a function may be accomplished by more than one system element, functions are unable to be allocated. Rather, functions are used to develop requirements, which are then allocated to solutions in the form of a physical architecture.

When unprecedented systems or systems are being developed that radically differ from those currently in use, the approach named "form follows function" is applied. The first function to identify stems from the need, which is then decomposed into lower levels of needed functionality. The functional description is translated into the physical by assigning functionality to requirements and requirements into a Physical Architecture. While function names may be allocated to specific Physical Architecture entities directly, it is often the case that some combination of two or more architectural entities accomplishes one function. The FAA preference is to translate functions into primitive performance requirements and then allocate these performance requirements to physical architecture entities.

#### 4.4.1.1. Functional Analysis Objectives

The Functional Analysis process helps to ensure that:

- All facets of a system's lifecycle, as illustrated in Figure 4.4-1, are covered from development to production, operation, and support
- All functional elements of the system are described, recognized, and defined
- All system concepts and requirements for specific system functions are related
- Requirements definition is improved
- Product integration is improved
- New and innovative designs and solutions are incorporated

#### 4.4.1.2. Process Overview

Functional Analysis examines a system's functions and subfunctions that are necessary to accomplish the system's operation or mission. It describes *what* the system will do, not *how* it will do it. Functional Analysis is conducted at the level needed to support later synthesis efforts, with all operational modes and environments included. Each function required to meet the operational needs of a system is identified and defined; once defined, the functions are then used to define the system requirements, and a functional architecture is developed based on the identified requirements. The process is then taken to a greater level of detail as the identified functions are further decomposed into subfunctions, and the requirements and physical architecture associated with those functions are each decomposed as well. This process is iterated until the system is completely decomposed into basic subfunctions, and each subfunction at the lowest level is completely, simply, and uniquely defined by its requirements. In this process, the interfaces between each of the functions and subfunctions are fully defined, as are the interfaces with the environment and external systems. The functions and subfunctions are arrayed in a Functional Architecture to show their relationships and interfaces (internal and external). Figure 4.4-2 illustrates the Functional Analysis process flow.

Functions shall be:

- Arranged in their logical sequence
- Well defined in their inputs, outputs, and functional interfaces (internal and external)
- Traceable from beginning to end conditions
- Analyzed, determined, and defined for time-critical requirements
- Successively established from the highest to lowest level for each function and interface
- Defined in terms of what needs to be accomplished in verb–noun combinations without describing *how* it is to be accomplished
- Traceable downward through successive functional decompositions

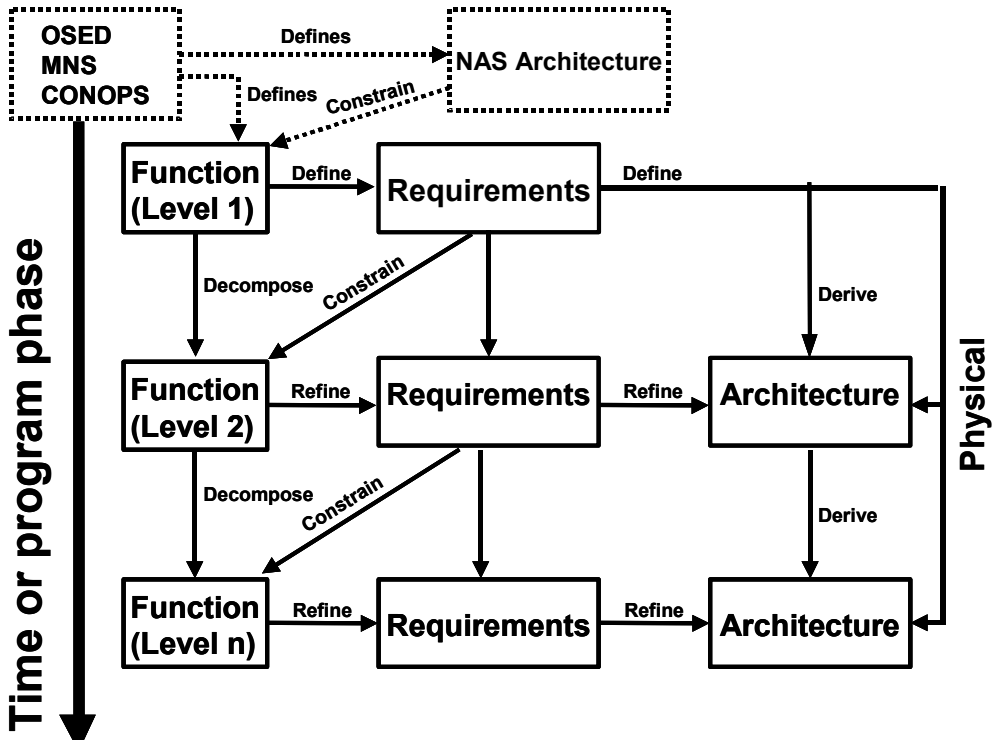



**Figure 4.4-2. Functional Analysis Process Flow and Interface with Physical Architecture and Requirements**

It is recommended that the Functional Analysis process be conducted in conjunction with Requirements Management (Section 4.3), Synthesis (Section 4.5), and Trade Studies (Section 4.6) (Figure 4.4-3) to:

- Define successively lower-level functions required to satisfy higher-level requirements and to define increasingly detailed sets of Functional Architectures
- Define mission- and environment-driven performance requirements and determine that higher-level requirements are satisfied

- Flow down performance requirements and design constraints
- Refine the definition of product and process solutions

### 4.4.2. Inputs to Functional Analysis

The more that is known about a system, the more complete the Functional Architecture. At the highest level of Functional Analysis for the FAA (the National Airspace System (NAS)), only the Mission Need Statement (MNS) may be available as input. The needs reflected in the MNS are translated into a Concept of Operations (CONOPS). A CONOPS is a high-level form of Functional Analysis that is solely derived from the user's perspective. It is recommended that the CONOPS serve as a baseline for the more detailed Functional Analyses to follow. (Paragraph 4.4.4.2 provides more information on CONOPS.) As iterations progress, it is recommended that higher-level Physical Architectures and Requirements be considered as they become available. If the output of the Requirements Management (Section 4.3) task is incomplete, the Functional Analysis task reveals missing Requirements and helps to refine or clarify others.

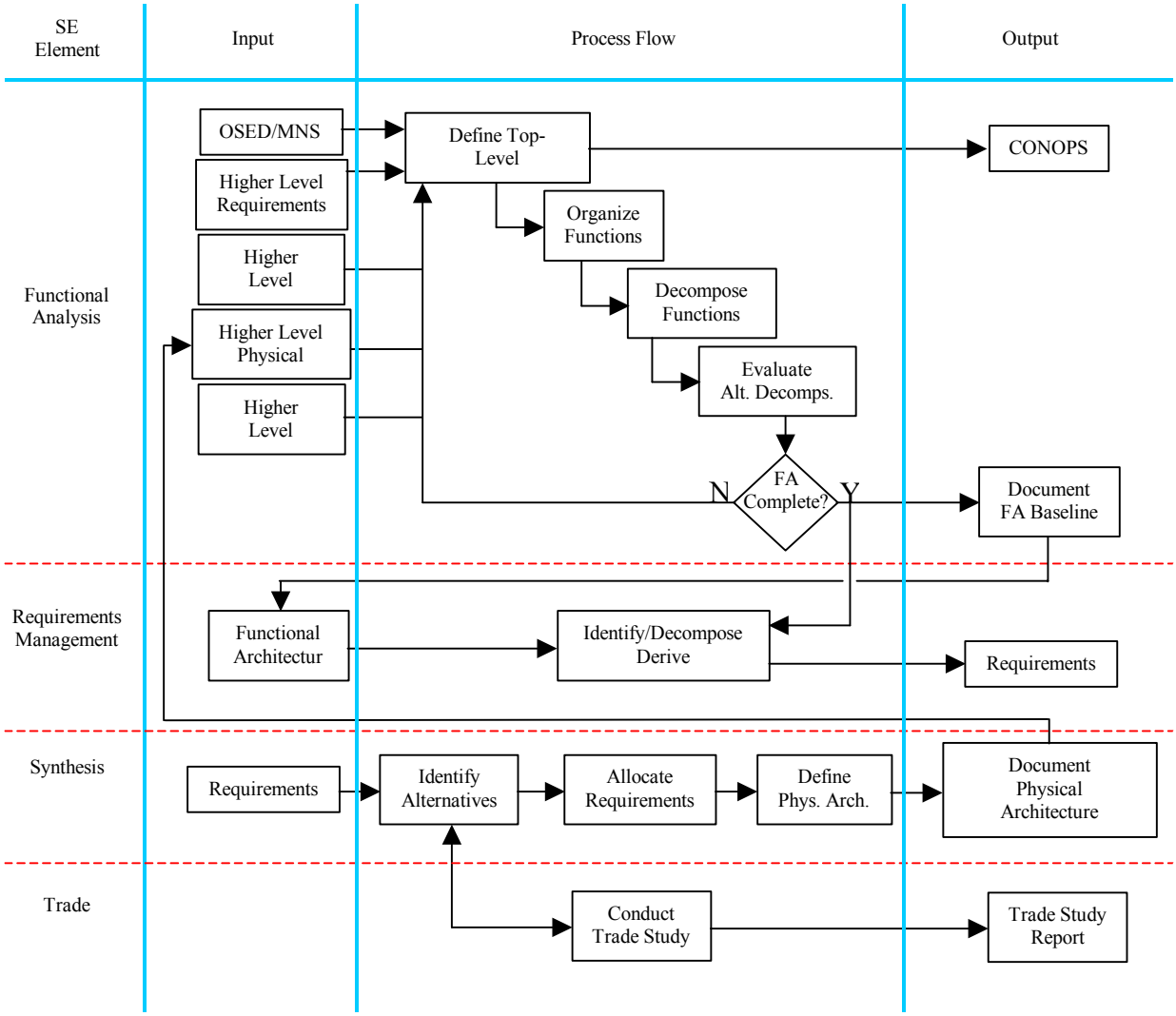



**Figure 4.4-3. Functional Analysis Process Flow**

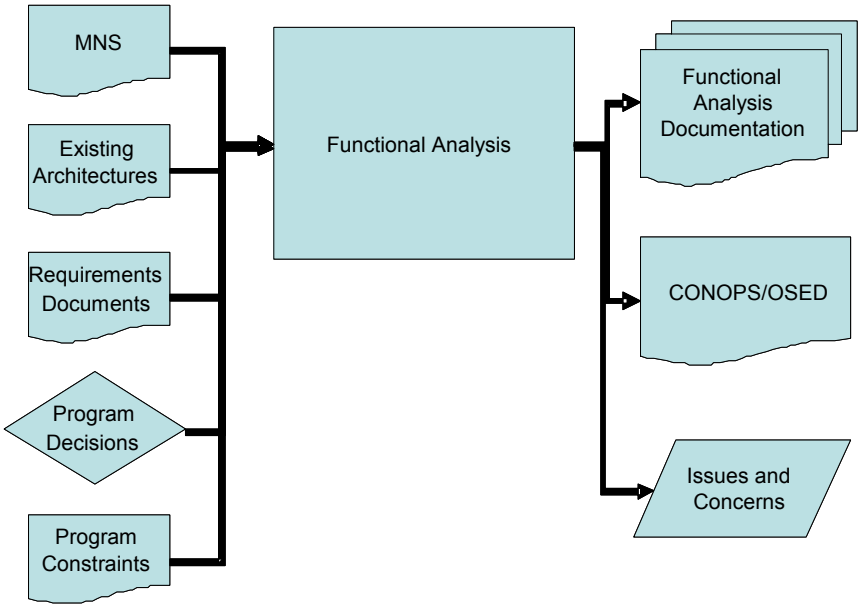



**Figure 4.4-4. Several Representative Inputs to Functional Analysis**

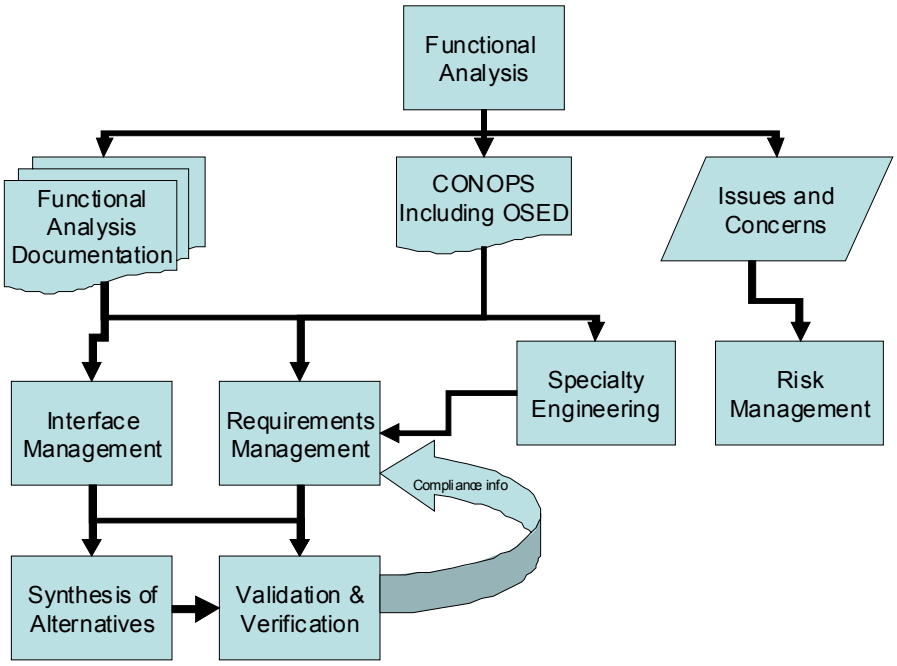



**Figure 4.4-5. Several Representative Outputs of Functional Analysis**

Figure 4.4.3 depicts Functional Analysis's process flow, while Figures 4.4-4 and 4.4-5 illustrate several representative inputs and outputs to/from Functional Analysis. The following is a more comprehensive list of the inputs to Functional Analysis:

- FAA Management Decisions
- Information on Legacy Systems

- Planning documents, such as the Integrated Program Plan (IPP)
- NAS Level (and program, if available) System Engineering Management Plan (SEMP)
- Defined NAS capability shortfalls and/or needs in the MNS (including validated needs statement)
- Requirements, such as any existing specifications and standards requirements, including requirements documents (reference documents)
- Program decisions (such as Constraints relating to existing hardware and software)
- Existing Physical Architectures
- Higher-level Functional Architectures
- Information on interfaces, including Interface Control Documents (ICD)
- Design Analysis Reports (DAR)
- Analysis Criteria

### 4.4.3. Functional Analysis Process Tasks

The Functional Analysis process is summarized in Figure 4.4-1. The five major process tasks listed in Figure 4.4-1 are described in the remainder of this section.

#### 4.4.3.1. Task 1: Define Top-Level Functions (from Inputs)

The first task in defining the system from a functional standpoint is to review the MNS, existing Operational Services and Environmental Descriptions (OSED), and any existing requirements documents to ensure a complete understanding of the top-level system missions/functions, environments, Requirements, and imposed Constraints. The MNS defines the needs the system is expected to meet. The CONOPS is developed from the MNS and normally includes an OSED. (The OSED is defined in Paragraph 4.4.4.2.1.) A system understanding from the perspective of these documents ensures that the system's relationship to its environment and external systems is considered during the development of the primary system functions. Figure 4.4-6 is a simplified example of an MNS and CONOPS for an office requiring the capability to record and store information from a computer. This example is used only to develop a sample functional flow diagram (FFD) (Paragraph 4.4.3.2.2.1). An actual MNS and CONOPS include much greater detail.

The system's primary mission(s) is defined using the MNS and any available system

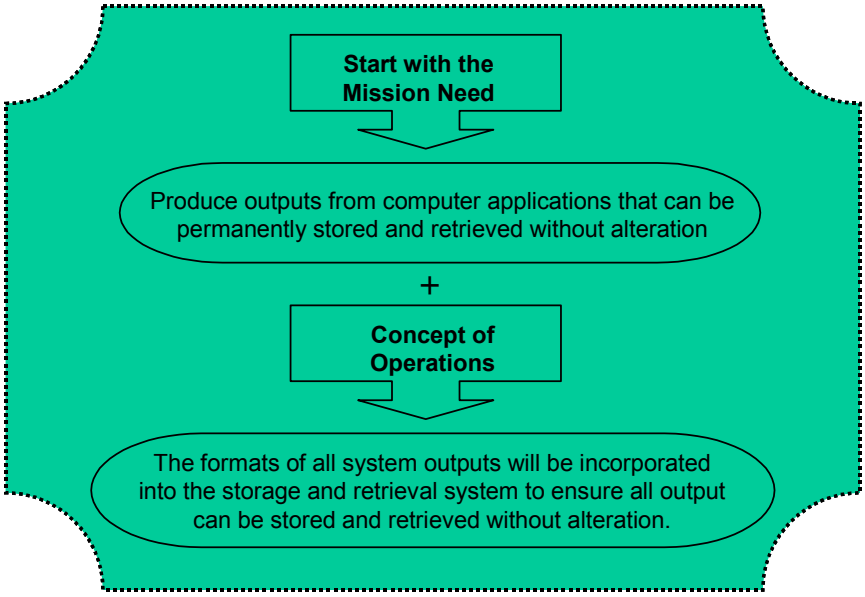


**Figure 4.4-6. Mission Need Statement and Concept of Operations**

descriptions, such as the OSED, the system's Requirements, and Constraints. This mission(s) is the primary function that the system fulfills, and it is named using the guidelines and naming convention described in the "Introduction to Functional Analysis" (Paragraph 4.4.1).

In addition, the internal and external interfaces (including ambient and operational environments) of the system are identified, and the functional relationships are defined. In the next task, these relationships are depicted through structured analysis using sequence diagrams, FFDs, and $N^2$ diagrams, which meet nearly all FAA program needs. In these depictions—examples of which appear in Figures 4.4-9 through 4.4-20—a large rectangular box represents the system, and the smaller boxes represent external elements outside of the main system. Flow arrows represent interfaces between the system and the external elements that describe which external element the system is transmitting to/receiving from and what data is being transmitted/received. Figure 4.4-7 shows the standard symbols used in these diagrams. ("Functional Analysis ools and Techniques" (Paragraph 4.4.5) provides other techniques that may be used with approved tailoring to the process.)

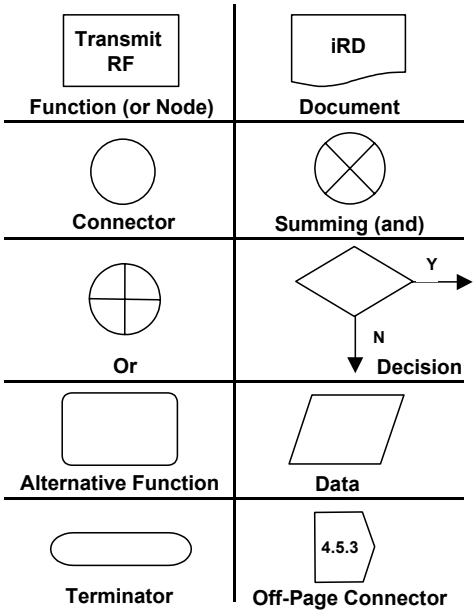



**Figure 4.4-7. Symbology Template for Functional Flow Diagramming**

In Task 1, the necessary functions that provide the required capabilities of the system, as specified by the need or Requirements, are defined. The activity represented by each of the functions shall be well defined, able to be implemented, and testable; and the interfaces to other functions shall be as simple as possible. It is recommended that these functions be developed with an eye toward the conversion of the Functional Architecture into Requirements and Requirements into a Physical Architecture. The development of complementary Functional and Physical Architectures requires multiple iterations between Functional Analysis, Requirements Management (Section 4.3), and Synthesis (Section 4.5).

Figure 4.4-8 lists functions based on the example MNS and CONOPS depicted in Figure 4.4-6. There are many approaches to describe these functions. The main criterion for task completion is a comprehensive list of the functions the system has to perform in order to meet its mission. For this task, the list does not need to follow a logical order.

**List of Functions**

- **Mission or central function: Produce outputs from computer applications**
  - **Provide output**
  - **Process output**
  - **Produce output**
  - **Generate output**
  - **Accept output information**
  - **Store output information**
  - **Retrieve stored output information**

**Figure 4.4-8. List of Primary Functions**

An analysis of operations and environment may be tailored to represent the available source of information. If detailed references to environmental data are absent in the initial Requirements Documents (iRD), Quality Function Deployment (QFD) or other methods described in Requirements Management may be used as supplements to elicit the information necessary to support follow-on Physical Architecture and Requirements tasks.

Affirmative answers to the following questions signify completion of Task 1:

- Have all missions, phases, and modes of operation been considered for the system?
- Have all functional elements been properly identified?
- Have all functional interfaces of the system to and from the environment been adequately identified and listed (physical/functional interface, connection parameters and modes, etc.)?
- Have the results of this review been captured in a list that identifies the system's mission and primary functions as well as interfaces with other systems and the environment?

#### 4.4.3.2. Task 2: Organize Functions Into Logical Relationships

The function list developed in Task 1 serves as an input to Task 2. The function list includes the central functions required for the system to accomplish its mission, but the list is not necessarily arranged in a sequence or logical relationship. During Task 2, the functions are arranged in at least one of the primary logical flow diagrams, which are applicable to most programs and indicate relationships based on function sequence and/or functional flow (input-function-output). The arrangement of the functions includes independent functions in parallel and dependent functions in series (e.g., when completion of the upstream function is necessary in order to begin the downstream function). A discussion of other techniques (used only when tailoring is approved) is included in "Functional Analysis Tools and Techniques" (Paragraph 4.4.5).

#### 4.4.3.2.1. Sequence Relationships

The sequence family of relationships includes both sequence and timing. Sequence relationships shall be used if sequence or timing is critical to the overall system function and when the relationships are simple. When sequencing is selected, the functions are arranged in order of sequence (i.e., preceding functions depicted before subsequent functions).

##### 4.4.3.2.1.1. Network Diagrams

Sequence relationships may be depicted as network diagrams. These diagrams shall be used if sequence is important to the function operation, but timing is not necessarily critical. Network diagrams display functions and sequential dependencies in a network format. A box (called a node) represents each function, and a line connecting two boxes represents the sequential dependency between the two functions. Figure 4.4-9 depicts a simple network diagram. Some analysts apply an action on line pattern, where the nodes represent events that partition the actions (on the lines) into time frames.

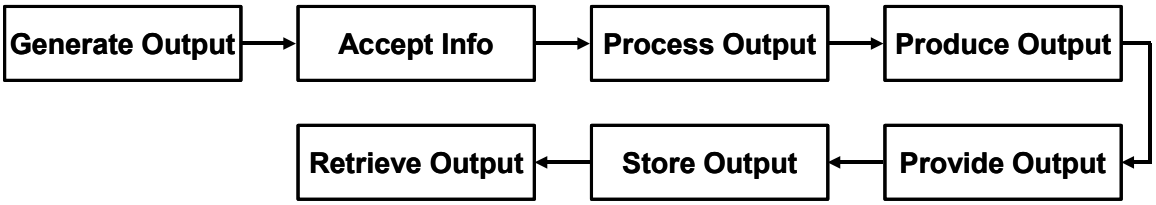



**Figure 4.4-9. Depiction of a Sequence Relationship Using a Network Diagram**

##### 4.4.3.2.1.2. Time Line Sequence Diagrams

Another way to organize functions in sequence is to use time line sequence diagrams. A time line sequence diagram depicts each function as a line or rectangle on a chart similar to a Gantt chart. The functions are stacked with preceding functions depicted to the upper left of subsequent functions. Time line sequence diagrams shall be used when a sequence relationship is selected, and timing is critical to the function operation. Figure 4.4-10 depicts a simple graphical deterministic time line sequence diagram.

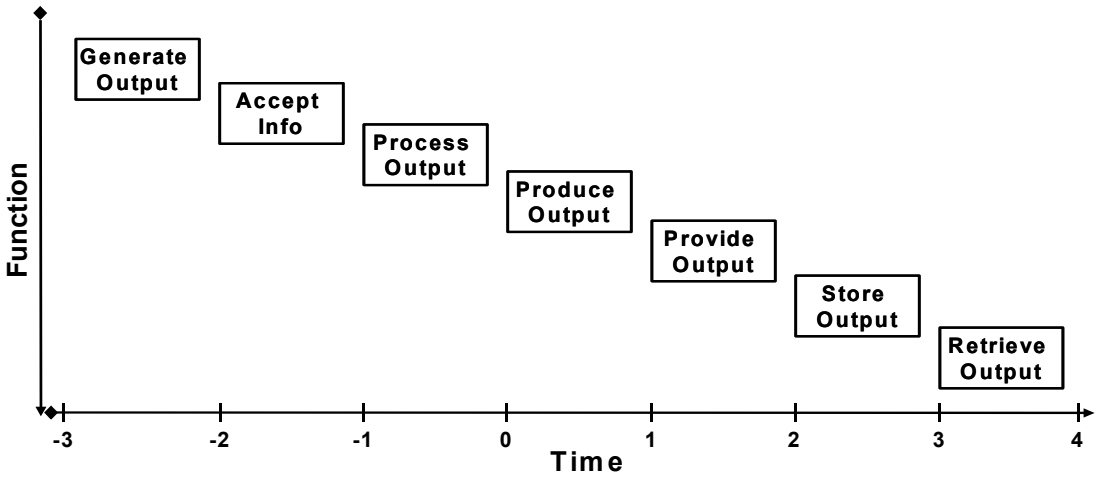



**Figure 4.4-10. Depiction of a Sequence Relationship Using a Time Line Sequence Diagram**

Time line sequence analysis considers functional durations and provides a more definitive description of the functional sequences than network diagrams are able to convey. It graphically depicts the concurrence, overlap, and sequential relationships of functions and related tasks. Time line sequence analyses are important in the tradeoff process between man and machine, including decisions regarding manual and automatic methods and allocation of times to subfunctions. In addition to defining subsystem/component time requirements, time line sequence analysis is used to develop Trade Studies (Section 4.6) in areas other than time considerations (e.g., should the spacecraft location be determined by the ground network or by on board computation using navigation satellite inputs?). Figure 4.4-11 depicts a maintenance time line sheet (TLS) that shows that the availability of an item (distiller) is dependent upon the concurrent completion of numerous maintenance tasks. Furthermore, the figure illustrates the traceability to higher-level requirements by referencing the appropriate FFD.

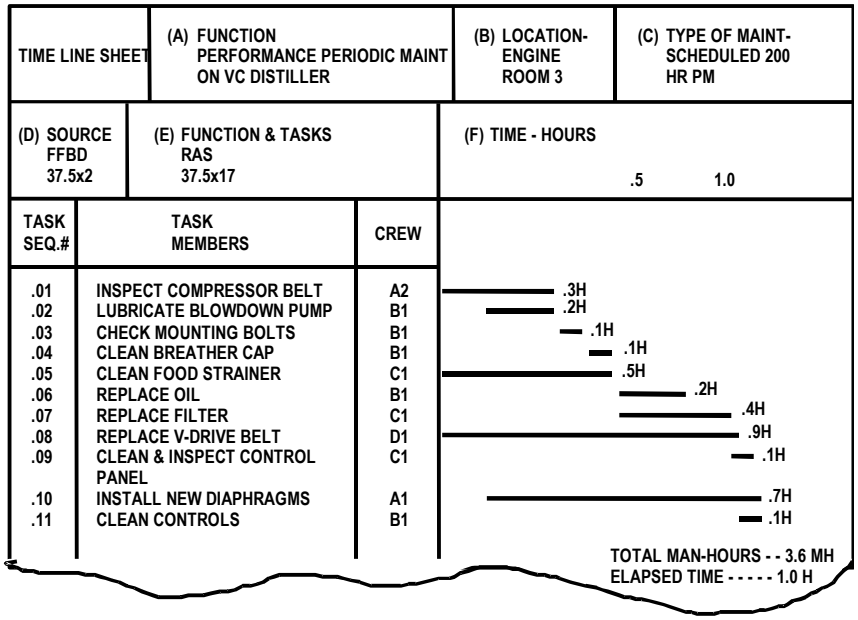



**Figure 4.4-11. Time Line Sheet for Maintenance of a Distiller**

Time line sequence analysis is performed on areas where time is critical to mission success, safety, utilization of resources, minimization of downtime, and/or increasing availability. The following areas are often categorized as time-critical:

- Functions affecting system reaction time
- Mission turnaround time
- Time countdown activities
- Functions requiring time line sequence analysis to determine optimum equipment and/or personnel utilization

Time line sequence analysis supports the development of design requirements for operation, test, and maintenance functions (additional techniques such as mathematical models and computer simulations may be necessary). In addition, the TLS is used to perform and record the analysis of time-critical functions and functional sequences. For time-critical functional sequences, it is necessary to specify the time requirements with associated tolerances.

#### 4.4.3.2.2. Functional Flow (Input-Function-Output) Logical Relationships

The FFD family consists of a group of analyses that depicts functional (input-function-output) relationships between functions. This family includes the Department of Defense (DoD) standard FFDs, $N^2$ diagrams, Integrated Definition for Function Modeling (IDEF) techniques, which are described in the following paragraphs, and the Unified Modeling Language (UML), which is described in Paragraph 4.4.5.2.6.2.

##### 4.4.3.2.2.1. Functional Flow Diagrams

The FFD, the FAA's recommended technique for Functional Analysis, is a multi-tier, time-sequenced step-by-step diagram of the system's functional flow. FFDs usually define the detailed, step-by-step, operational and support sequences for systems, but they are also used effectively to define processes in developing and producing systems. The software development processes also use FFDs extensively. In the system context, the functional flow steps may include combinations of hardware, software, personnel, facilities, and/or procedures. Although functional flow relationships are more complicated, they also convey more information than sequence diagrams. In the FFD method, the functions are organized and depicted by their logical inputs and outputs. Each function is shown in relation to the other functions by how the inputs and outputs feed and are fed by the other functions. A node labeled with the function name depicts each function. Arrows leading into the function depict inputs, while arrows leading out of the function depict outputs. Figure 4.4-12 depicts the output of function F0 as an input to function F1.

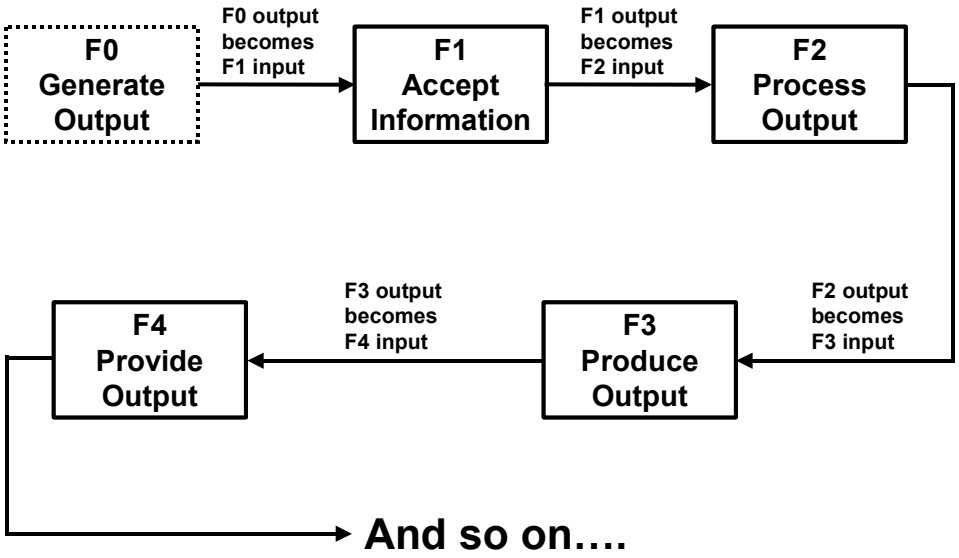



**Figure 4.4-12. Functional Flow Relationship**

With FFDs, the function is the machine or process that uses inputs to produce outputs. To illustrate: If a turbine engine is the system, then the function is the conversion of oxygen and fuel into mechanical energy in the form of thrust or torque. In Figure 4.4-13, the function is depicted as a box (node) with a function title of "Generate Torque/Thrust." Inputs are the elements needed for the function to operate correctly; the production of mechanical energy using a turbine engine requires oxygen and fuel. Therefore, oxygen and fuel are inputs to that function. Inputs are depicted as arrows leading into the functional node with the input arrows labeled appropriately. The output is the product of the function. In Figure 4.4-13, the engine function generates the "Torque/Thrust." The output is depicted as the arrow leading out of the functional node with the output arrow labeled appropriately.

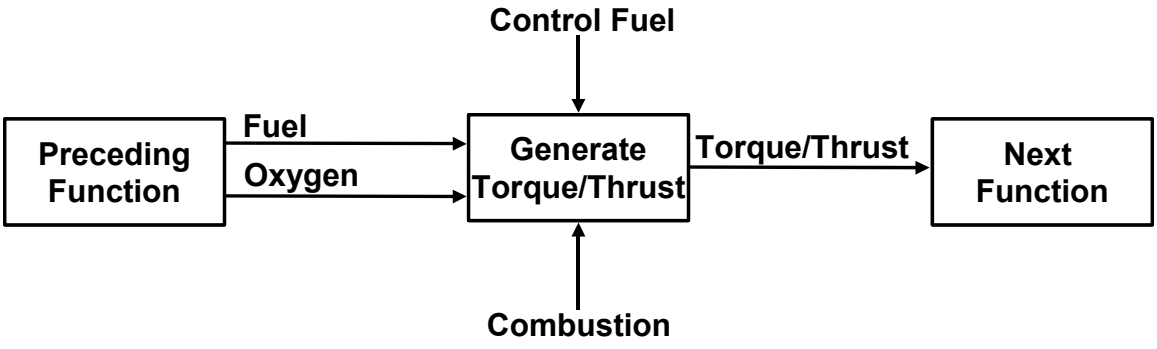



**Figure 4.4-13. Input-Function-Output Relationship**

**NOTE**

*When IDEF techniques (Paragraph 4.4.3.2.2.3) are being used, controls and mechanisms may be added. Controls are elements that manage the function and*

*are outputs of other functions, while mechanisms are elements that enable the function to work. Controls are depicted as arrows entering the function from the top, and mechanisms are depicted as arrows entering the function from the bottom.*

The functions depicted so far have been serial functions; however, many functions are parallel (i.e., they are functions that (1) independently feed the same downstream function, and/or (2) occur simultaneously). Figure 4.4-14 illustrates parallel functions.

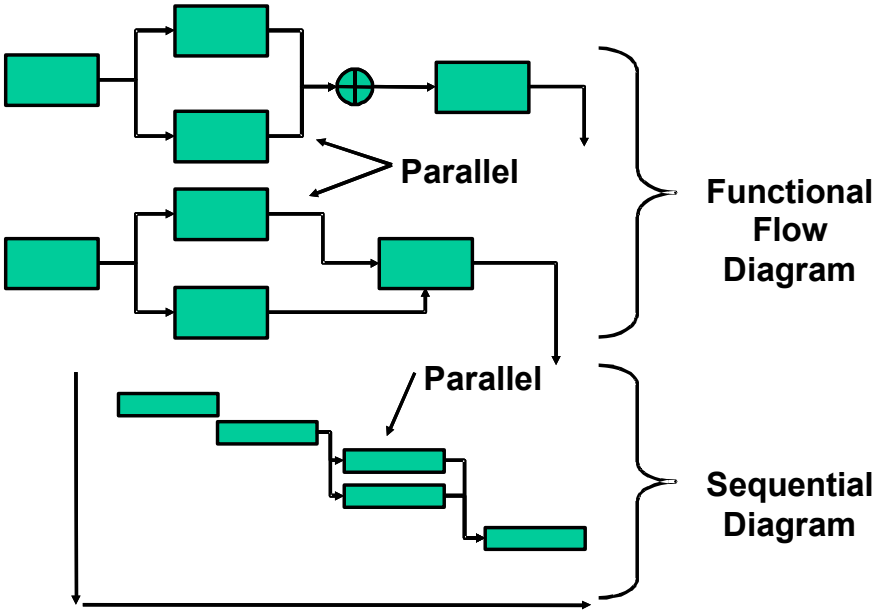



**Figure 4.4-14. Parallel Functions**

Figure 4.4-15 depicts a functional flow organization of the functions, in which functions are broken down into subfunctions using the techniques described in Task 2. This figure represents a simplified FFD, in which the inputs and outputs are not labeled, and the controls and mechanisms are not identified. Functional Analysis is performed to the level of detail needed to depict the functional description of the system.

Figure 4.4-15 also shows multiple functional levels; however, only the top level is complete. Each lower level shows an example expansion of one function. For example, at the second level, the top-level function F1 is expanded into its second-level functions, F1.1 through F1.6. At the third level, second-level function F1.4 is expanded. Finally, at the fourth level, function F1.4.3 is expanded. Each level indicates a different example of typical functional flow paths. Usually, only one or two levels are shown in one diagram to avoid confusion.

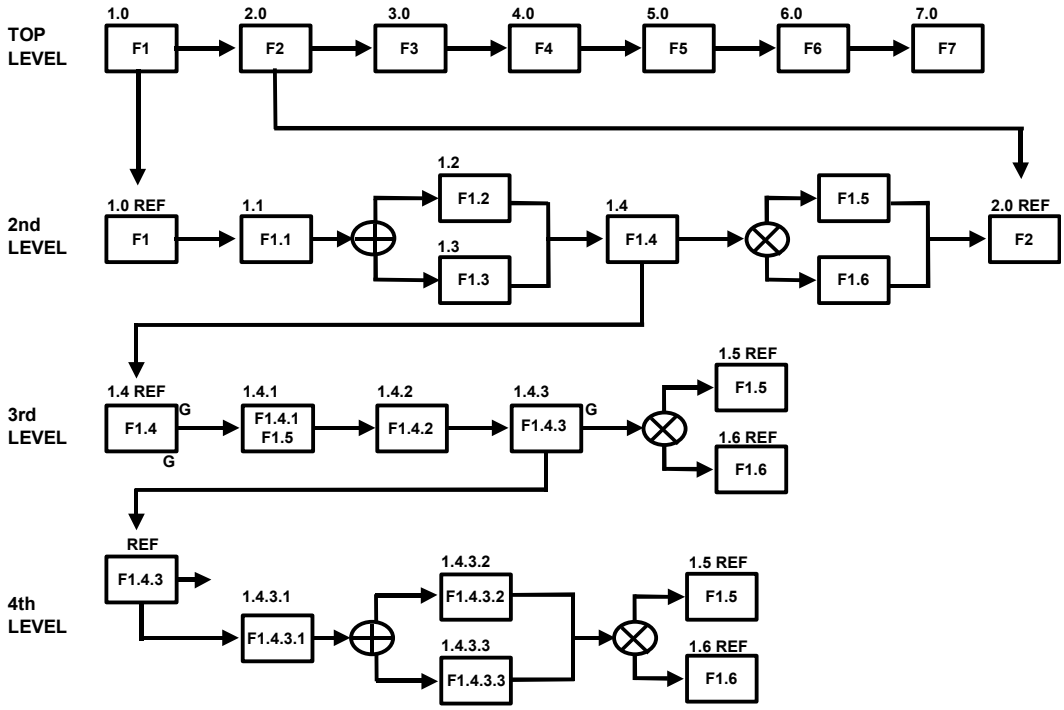



**Figure 4.4-15. Generic Functional Flow Diagram Example**

Adherence to the following rules promotes common understanding of FFDs:

- Number top-level functions with even integers and zero decimals (e.g., 1.0, 2.0, etc.) and cover the complete span of anticipated lifecycle functions
- Depict inputs to functions as entering from the left side and outputs as leaving from the right side
- Depict mechanisms as entering from the bottom and controls as entering from the top
- Display lower-level functions as emanating from the bottom
- Define the name of the function inside the box, replacing F1, F2, etc.
- Indicate a reference function (ref) at the beginning and end of all functional sequences, *except at the top level*
- Use an “OR” gate to indicate alternative functions; use an “AND” gate to indicate summing functions, where all functions are required (Figures 4.4-7 and 4.4-16)
- Indicate a “GO” “NO GO” sequence with an arrow leaving the right side of the function with the letter “G” for “GO” and an arrow out the bottom with “G-bar” for “NO GO”
- As is customary, when the second level or lower level is shown on a separate page, list the title of the function at the top center of the page for reference
- Typically, do not show the information flow, content of each functional step, and timing details on FFDs

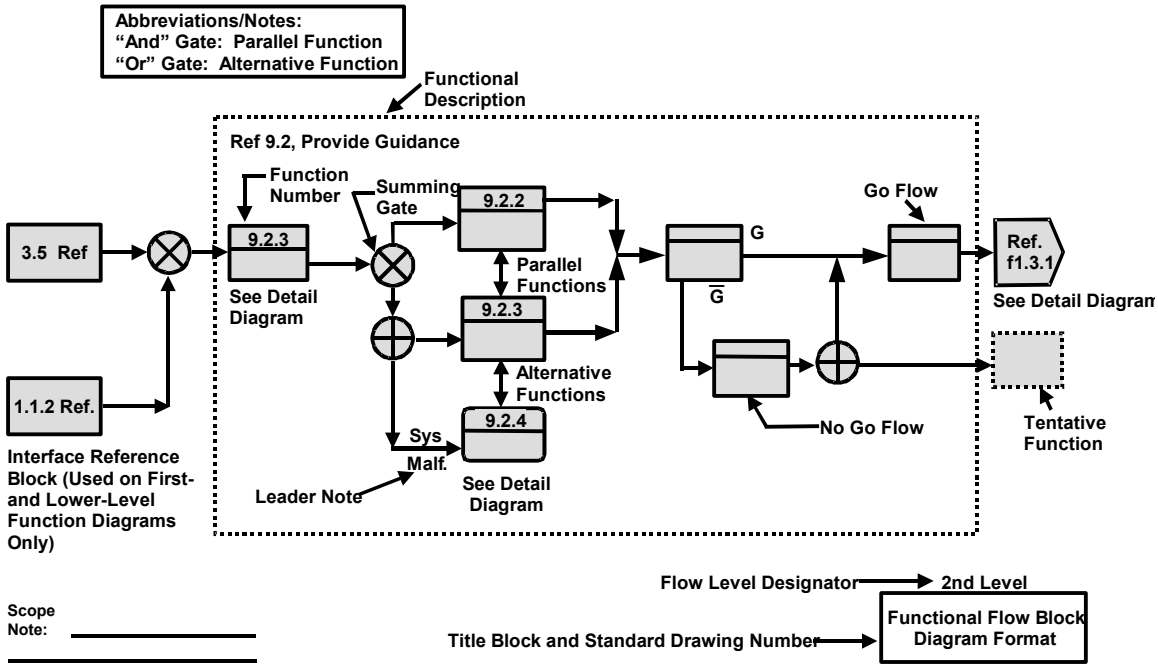



**Figure 4.4-16. Functional Flow Diagram Example**

### 4.4.3.2.2.2. Functional $N^2$ Diagrams

The $N^2$ diagram is a systematic approach to identify, define, tabulate, design, and analyze functional and physical interfaces. A functional $N^2$ diagram depicts the interfaces between functions in a system. The $N^2$ diagram is a visual matrix that requires the user to generate complete definitions of the system functional interfaces in a rigid, bidirectional, fixed framework. A basic $N^2$ diagram is illustrated in Figure 4.4-17.

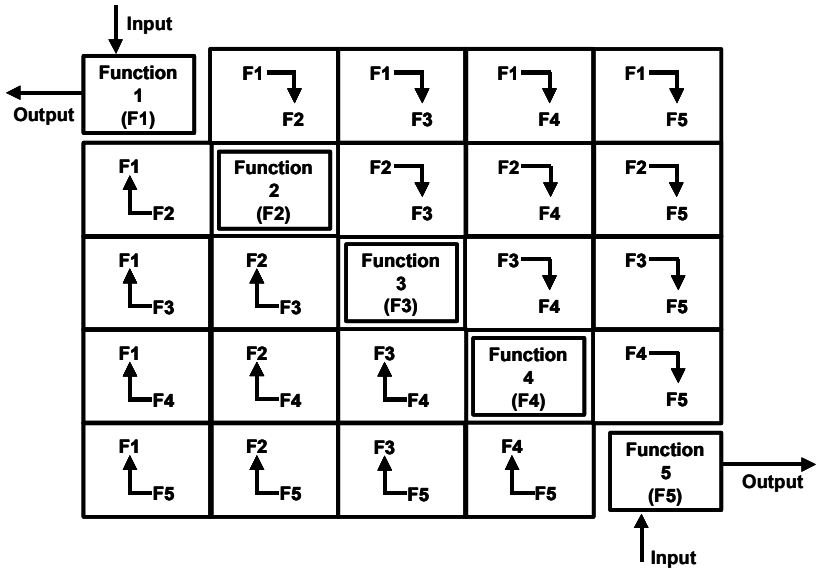



**Figure 4.4-17. Functional $N^2$ Diagram**

The $N^2$ diagram customarily is used to develop data interfaces, primarily in the software areas; however, it also may be used to develop other interfaces, including functional and physical interfaces. In this method, the system functions are placed on the diagonal axis; the remainder

of the squares in the N x N matrix represents the interface inputs and outputs. The presence of a blank square indicates that there is no interface between the respective system functions. Data flows in a clockwise direction between functions (i.e., the symbol F1 ↴ F2 indicates data flowing from function F1 to function F2; the symbol F2 ↥ F1 indicates the feedback). The transmitted data is defined in the appropriate squares. The diagram is complete when each function has been compared to all other functions. The $N^2$ diagram may be used in successively lower levels down to the component functional level.

$N^2$ diagrams are a valuable tool for not only identifying functional interfaces, but also for pinpointing areas where conflicts may arise between functions so that system integration proceeds smoothly and efficiently.

#### 4.4.3.2.2.3. Integrated Definition for Function Modeling Diagrams

IDEF is a common modeling tool for conducting analysis, development, and integration of information technology systems and software engineering analysis. Whereas FFDs show the functional flow of a product, IDEF diagrams show:

- Data flow
- System control
- Flow of lifecycle processes

The U.S. Air Force originally developed IDEF for manufacturing planning. IDEF is a compound acronym that stands for Integrated Computer-Aided Manufacturing (ICAM) Definition. Originally called IDEF, other IDEF languages have since been developed, forcing the languages to adopt numbering system; thus, this technique is now called IDEF 0. IDEF 0 has demonstrated an ability to depict a variety of engineering, operational, manufacturing, and other types of processes at any level of detail. It provides disciplined, rigorous, and precise descriptions while promoting standardization in use and interpretation.

IDEF is a model composed of a hierarchical series of diagrams, text, and a glossary that are cross-referenced. The two primary modeling components are functions and data objects that interrelate the functions. As shown in Figure 4.4-18 (IDEF box format), the position at which the arrow attaches to a box conveys the role of the data object interface. These roles consist of:

- Input
- Mechanism
- Output
- Control

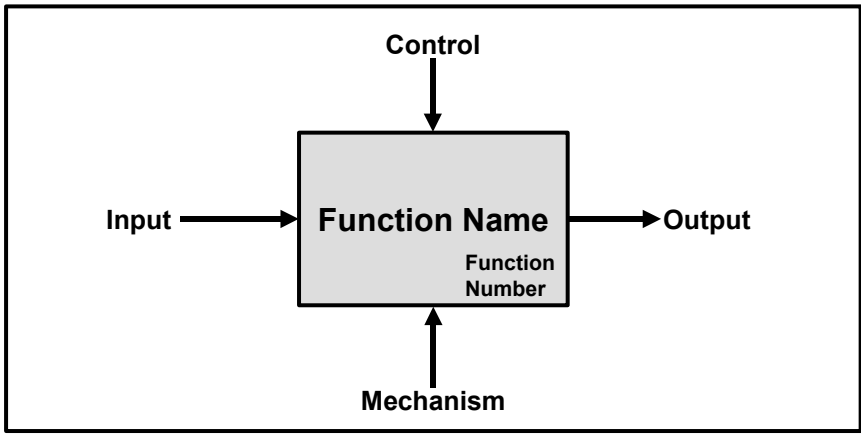



**Figure 4.4-18. Integrated Definition for Function Modeling Function Diagram**

The inputs, the data objects acted upon by the function or operation, enter from the left. The mechanism (additional support to perform the function) arrow attaches to the box from the bottom. The outputs of the function leave the function box from the right. The controls enter the top of the box.

The IDEF process begins with the identification of the prime function to be decomposed. This function is identified on a top-level context diagram that defines the scope of the particular IDEF analysis. Figure 4.4-19 illustrates a top-level context diagram for an information system management process. From this diagram, lower-level diagrams are generated. An example of a derived diagram—called a “child” in IDEF terminology—for a lifecycle function is shown in Figure 4.4-20.

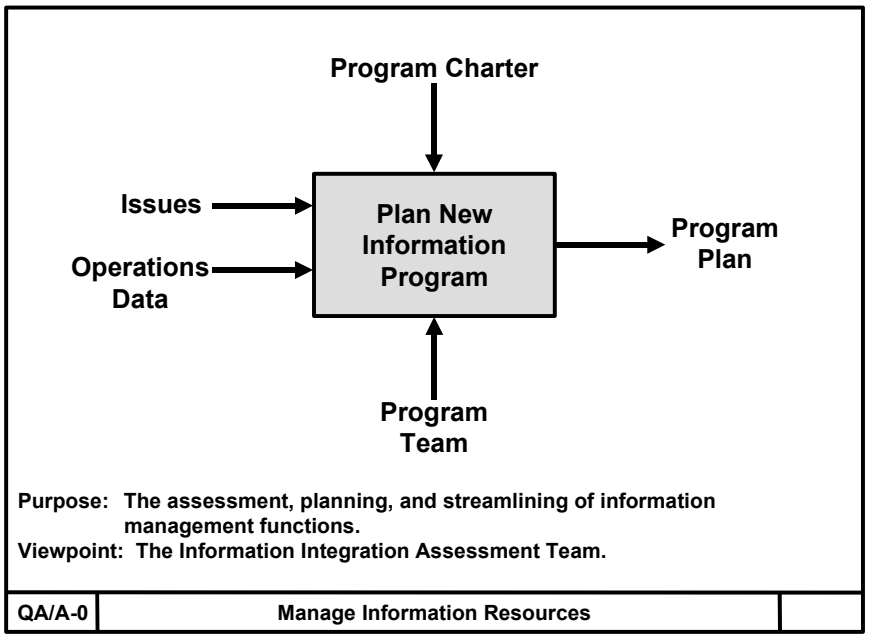



**Figure 4.4-19. Top-Level Context Diagram**

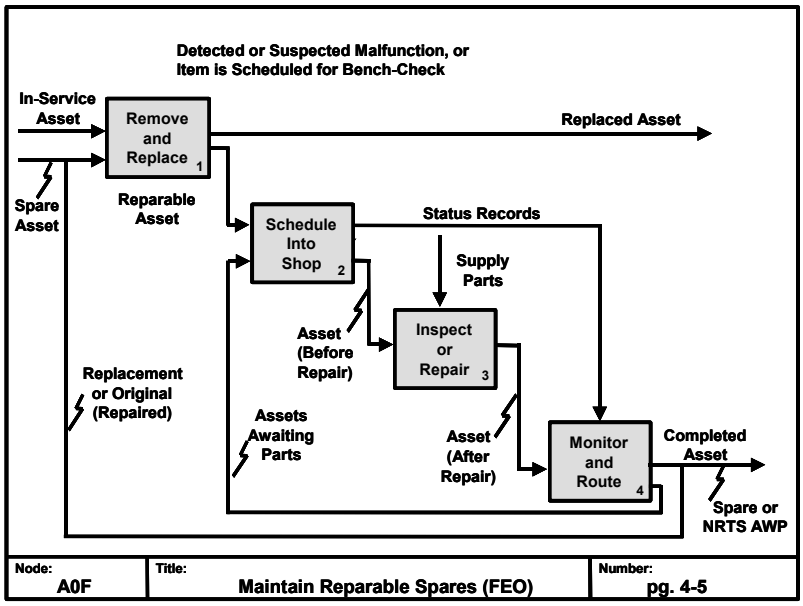



**Figure 4.4-20. Derived Diagram ("Child" in Integrated Definition for Function Modeling Terminology)**

Affirmative answers to the following questions signify completion of Task 2:

- Are all functions in the function list depicted?
- Are all functions written in the form verb–noun format?

- Are all functional interfaces depicted graphically?
- Does the depiction show end-to-end functional relationships?
- Are parallel and serial relationships accurately depicted?

#### 4.4.3.3. Task 3: Decompose Higher-Level Functions Into Lower-Level Functions

In this task, higher-level functions are decomposed into subfunctions, with specificity increasing at each level of decomposition. Functional decomposition is performed using the techniques described in Tasks 1 and 2 with respect to sequence and logical diagramming or alternatively with the techniques described in "Functional Analysis Tools and Techniques" (Paragraph 4.4.5). The stepwise decomposition of a system basically is a top-down approach to problem-solving. Shown graphically in Figures 4.4-21 through 4.4-24, the decomposition is carried to a level at which the functions have been totally decomposed into basic subfunctions, and each subfunction at the lowest level is completely, simply, and uniquely defined by its Requirements. This means that functional decomposition continues as long as there is a further need to define lower-level Requirements. When the requirements development process ceases, Functional Analysis may cease.

The objective of Task 3 is to develop a hierarchy of Functional Analysis diagrams that describes the functions at all levels of the system. This hierarchy is only a portion of the Functional Architecture, which is not complete until all Requirements and other Constraints have been appropriately decomposed.

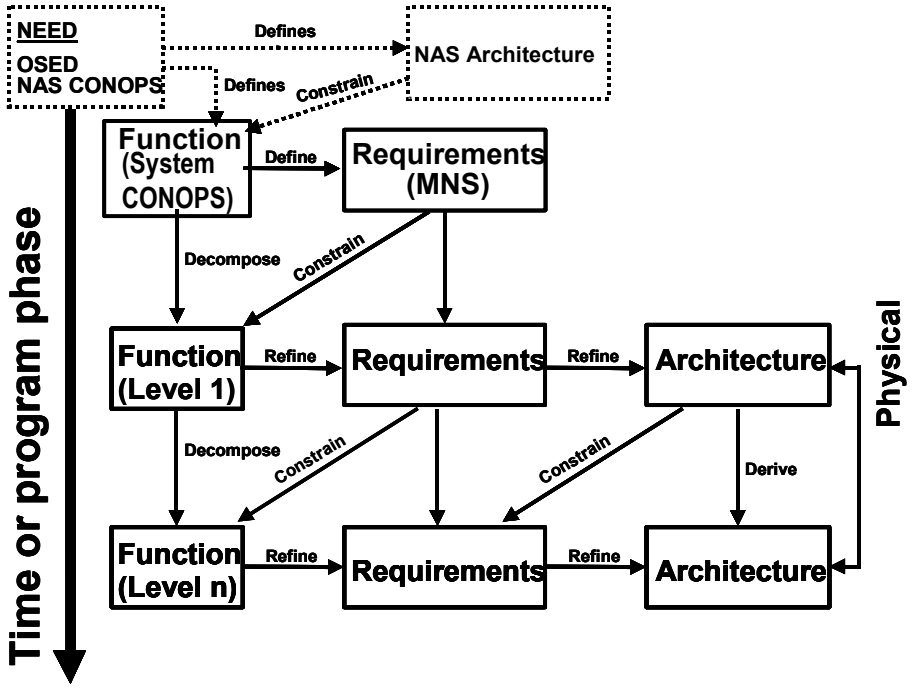



**Figure 4.4-21. Decomposition of Higher-Level Functions into Lower-Level Functions**

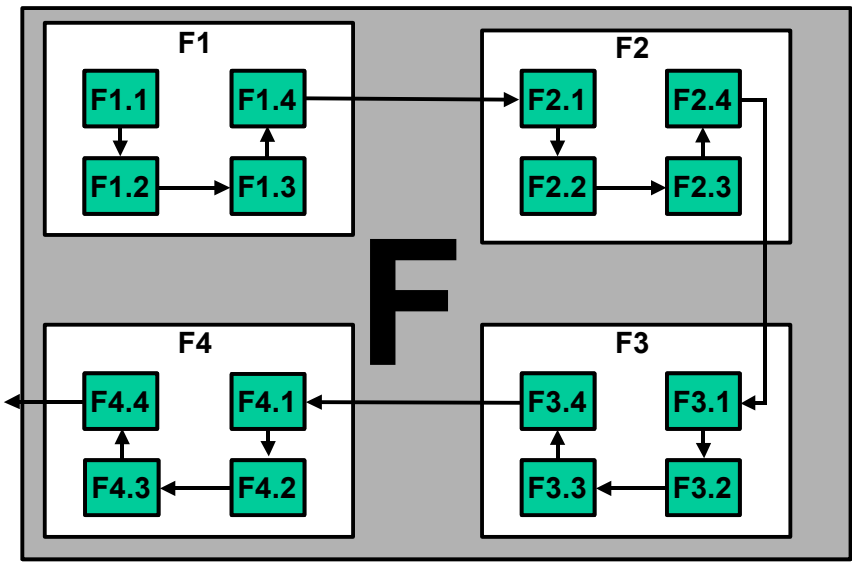



**Figure 4.4-22. Another View of Decomposition of Higher-Level Functions into Lower-Level Functions**

Task 3 is performed iteratively using the steps and techniques described in Tasks 1 and 2. Since higher-level functions exist for this task, the subfunctions are based on the higher-level functions developed in the previous tasks. In Figure 4.4-23, which uses the functions list from Figure 4.4-8, function F3 is decomposed into subfunctions labeled as the second level. Next, the functions in the second level are further decomposed to the third level. This process continues until all the functions are totally decomposed into basic subfunctions, and each subfunction at the lowest level is completely, simply, and uniquely defined by its Requirements. At each level, Functional Analysis feeds Requirements Management (Section 4.3) and Requirements feeds Synthesis (Section 4.5), as shown in Figure 4.4-21 and further illustrated in Figures 4.4-24 through 4.4-27.

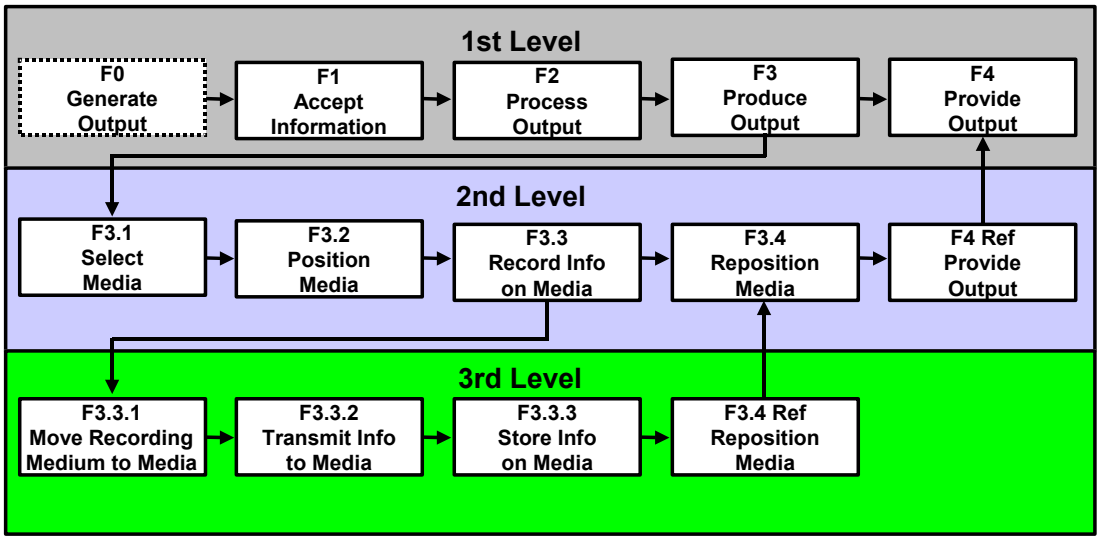



**Figure 4.4-23. Higher-Level Functions Broken Down into Lower-Level Subfunctions**

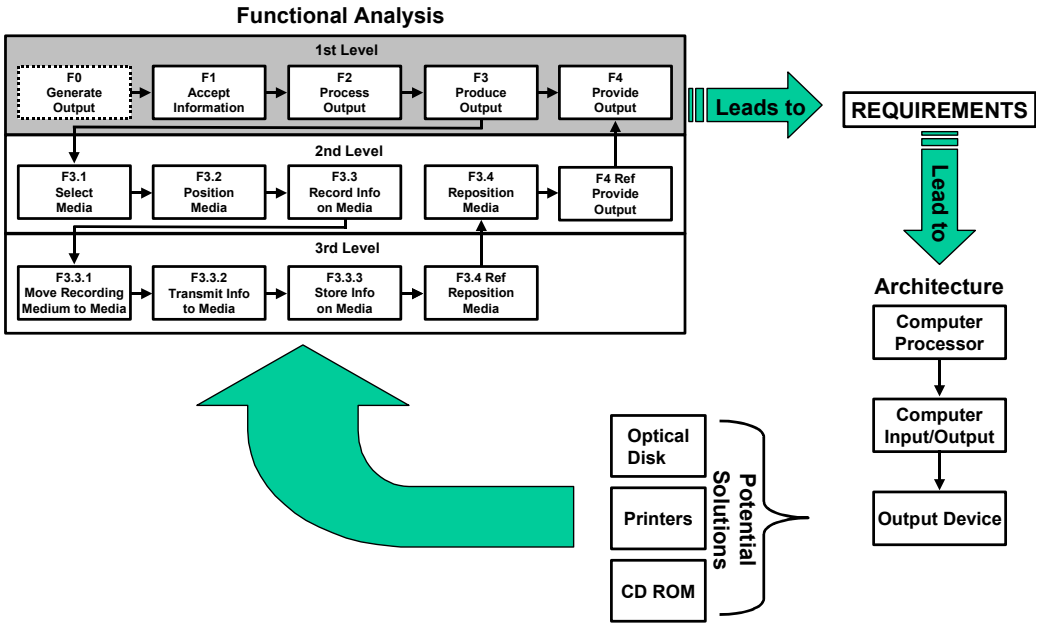



**Figure 4.4-24. Functions Lead to Requirements; Requirements Lead to Physical Architectures**

Requirements Management and Synthesis detail the process that turns functions into Requirements and Requirements into a Physical Architecture. It is important to note that the next Functional Analysis level is bound and framed by the Requirements and Physical Architecture refined from the preceding Requirements Management and Synthesis activities (Figure 4.4-25).

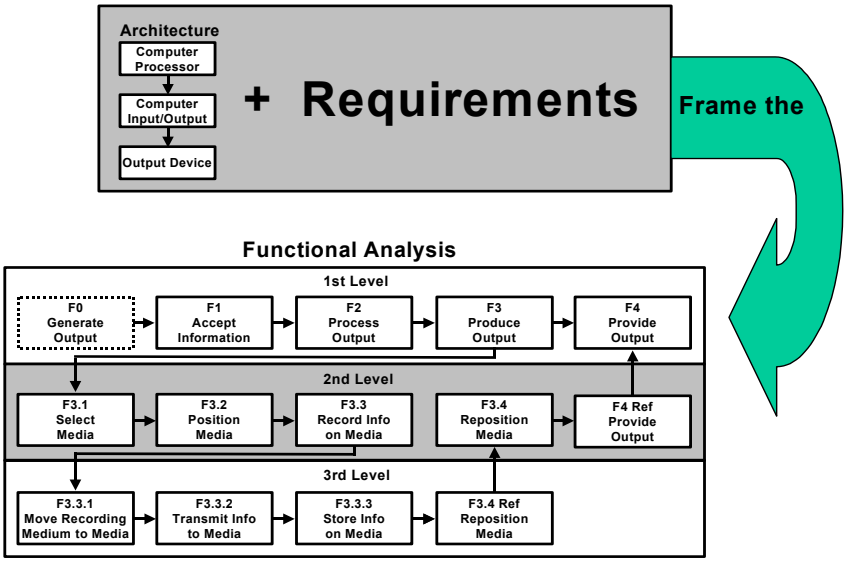



**Figure 4.4-25. Requirements and Physical Architecture Frame the Next Functional Analysis Level**

When this spiral process completes one rotation, the Functional Analysis process recommences (Figures 4.4-26 and 4.4-27) at the next lower level and repeats until each function is totally decomposed into its basic subfunctions, and each subfunction at the lowest level is completely, simply, and uniquely defined by its Requirements.

Affirmative answers to the following questions signify completion of Task 3:

- Has a complete set of Functional Analysis diagrams been prepared?
- Has each function been decomposed to its lowest level within program needs?
- Is each function completely, simply, and uniquely defined by its Requirements?
- Has a description of each function been developed?
- Is the requirements development complete?

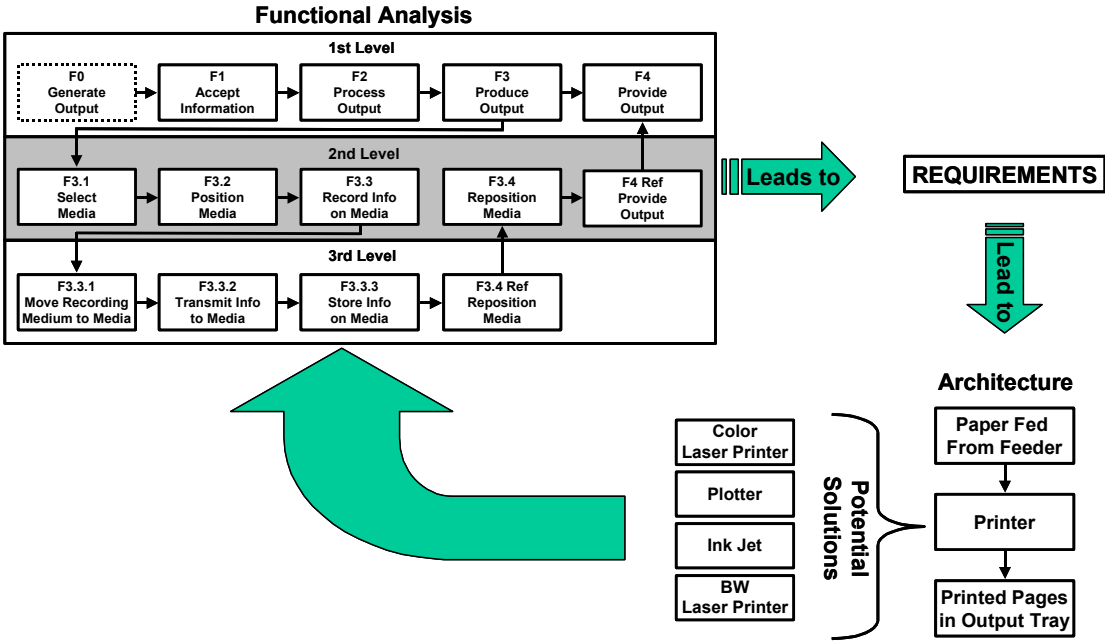



**Figure 4.4-26. Repeating the Functional Analysis Process at the Next Lower Level**

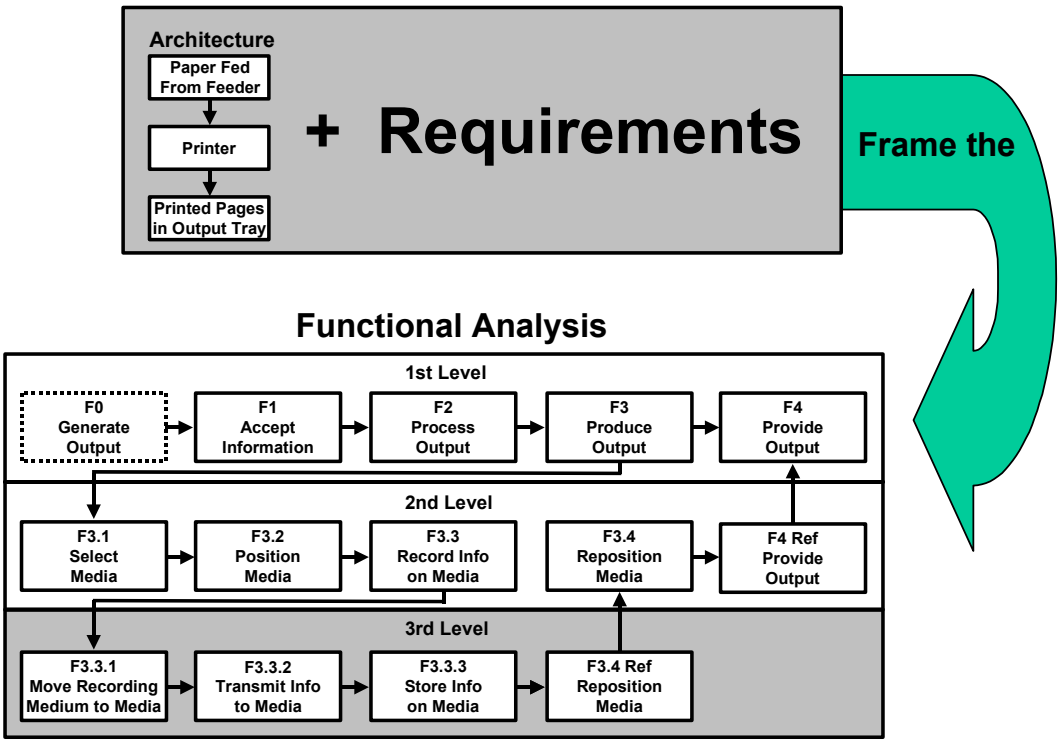



**Figure 4.4-27. Preceding Requirements and Physical Architectures Continue to Frame Next Lower Level**

### 4.4.3.4. Task 4: Evaluate Alternative Decompositions

In this task, alternative decompositions of functions (Functional Architectures) and Requirements at all levels are evaluated. These evaluations are necessary because there is no single "correct" decomposition; however, not all decompositions are of equal merit. It is necessary to evaluate alternative decompositions in order to select the decomposition best suited to the Requirements. Also, there are other reasons to evaluate alternative decompositions (e.g., because of the reality of the system's Design Constraints or the desire to use commercial-off-the-shelf (COTS) or nondevelopmental item (NDI) components, often, multiple alternative Functional Architectures are produced, which are then compared during the Trade Studies process (Section 4.6) in order to select the Functional Architecture that most effectively meets mission objectives).

The evaluation of alternative decompositions of functions is subjective and dependent upon personal preference and taste. The purpose of Task 4 is to ensure that other methods to conduct the decomposition are evaluated. In this task, personal preference and consensus among the stakeholders become factors in the selection of the best Functional Architecture. Any selected Functional Architecture shall reflect the system's functions. However, variances in the alternative Functional Architectures may provide a competitive edge to one or more of the alternatives.

By the end of this process, the Requirements for each subfunction at the lowest levels of the Functional Architecture are allocated via the Requirements Management process ("Task 3: Allocate Requirements" (Paragraph 4.3.4.3)) to hardware, software, interfaces, operations, or a database, and then to a specific configuration item. As it is necessary to verify Requirements,

the objective of Task 4 is to select those decompositions that promote straightforward Requirements that may be validated and verified. (Validation and Verification (Section 4.12) further addresses this issue.) In addition, decompositions that allow a single function to be used at several places within the hierarchy, thereby simplifying development, may be identified.

Task 4 requires "best engineering judgment," as the "goodness" of each functional decomposition is evaluated by measuring the degree to which each module displays the following attributes:

- Performs a single function
- Is a logical task
- Leads to a Requirement(s) that may be separately validated
- Has a single input point and a single output point
- Is independent within each level of the hierarchy (higher independence allows the implementation of the module independent of the other modules)

Because system design does not occur in a vacuum, it is necessary to consider opportunities to use COTS or NDI hardware and software. As a result, a subfunction that has already been implemented in a compatible form on another system may be preferred to one that has not.

The selection of a final system functional decomposition signifies completion of Task 4.

#### 4.4.3.5. Task 5: Document Functional Analysis Baseline

The last task in the Functional Analysis process is documenting the process, including the selected Baseline as the basis for Requirements Management (Section 4.3) and Synthesis (Section 4.5). The documentation includes the outputs listed in Figure 4.4-1. At this point, any necessary revisions or changes to the functional decomposition, sequence and time lines, functional interfaces, etc., are made to ensure their completeness and consistency with one another. Also, the products of the Functional Analysis process (e.g., FFDs, functional descriptions, function interface descriptions, and time lines) are developed. These products may be documented separately or combined in a Functional Analysis Document (FAD).

Affirmative answers to the following questions signify completion of Task 5:

- Have all of the initial functions been decomposed into subfunctions?
- Do the subfunctions cover the total scope of the parent function?
- Are the functions arranged correctly with respect to the dependence of the functions?
- Have all functional interfaces been defined?
- Have any new functional interfaces between initial functions been identified that were discovered during the function decomposition process? (These may drive new system element interfaces.) If so, have the new interfaces been documented in control sheets?
- Has a Functional Analysis document been prepared to document the functional Baseline?
- Have all functional Requirements been identified and decomposed?

### 4.4.4. Outputs of Functional Analysis

#### 4.4.4.1. Functional Architecture

The most common output of the Functional Analysis process is a "living" Functional Architecture document that contains a tailored combination of the following:

- Functional Architecture Baseline
- Functional interface list
- Alternative decompositions
- FFDs
- Functional time lines and sequences
- Data flow diagrams (DFD) and threads
- $N^2$ diagram
- Other functional descriptions

#### 4.4.4.2. Concept of Operations

In addition to the previous list, the CONOPS, which essentially is a top-level narrative Functional Analysis document, is also an output of Functional Analysis process. A CONOPS is a user-oriented document that describes system functional characteristics for a proposed system from the user's viewpoint. It explains the existing system, current environment, users, the interaction among users and the system, and organizational impacts. The CONOPS document is written in order to communicate overall quantitative and qualitative system characteristics to the user, buyer, developer, and other organizational elements. The CONOPS aids in requirements capture and communication of need to the developing organization. Posing the need in the user's language helps to ensure that the user is able to more accurately express the problem. Subsequently, the system engineers have a better foundation upon which to begin the lower-level Functional Analyses, requirements definition, and initial design of the system.

The following is a list of the essential elements indicative of all CONOPS:

- Description of the current system or situation
- Insight into the user's environment
- Description of the functions to be performed
- Description of the needs that motivate development of a new system or modification of an existing system
- Insight into the new Requirements
- Opportunity for the developer to recommend alternative solutions
- Description of the operational features of the proposed system
- User's view of the Requirements

At minimum, there are two levels of CONOPS: (1) NAS Level and (2) System Level CONOPS.

A NAS Level CONOPS is performed at the highest level of the Functional Analysis process and is a narrative expression of the user's desired change with some performance indicators. It is a high-level document that indicates, from the user's perspective, the desired end-state for the respective system in the NAS.

A System Level CONOPS is an extension of a NAS Level CONOPS with an emphasis on a particular system. It is more detailed and substantial, but it is still an expression of the user's needs with respect to a specific system within the NAS. It is recommended that a System Level CONOPS, in particular, have the following characteristics:

- Written in the user's language using the user's preferred format
- Written in narrative prose (in contrast to a technical requirements specification)
- Organized so as to tell a story and accompanied by visual forms (diagrams, illustrations, graphs) and storyboards whenever possible
- Provide a bridge between the user's needs and the developer's technical requirements documents
- Describe the user's general system goals, mission, function, and components
- Evoke the user's views and expectations
- Provide an outlet for the user's preferences
- Provide a place to document vague and unmeasurable Requirements (i.e., the user is able to state his/her desire for fast response or reliable operation; these desires are quantified during the process of developing the requirements specifications and during the flowdown of Requirements to the Physical Architecture
- Make the user feel in control

Figure 4.4-28 serves as a guideline for System Level CONOPS content:

1. Introduction
   a. System overview
   b. Definition of terms
   c. References
2. Operational need
   a. Operational problems solved
   b. Opportunities created
   c. Existing operations/functions that require change
   d. Organization constraints
   e. Actors that will interact with system
3. System justification
   a. Capability shortfalls of current system
   b. Potential benefits of new system
   c. Identified priorities of new features
      i. Critical
      ii. Essential
      iii. Routine
   d. Assumptions and constraints
4. OSED (include if available)
5. Business impact
   a. Impact on current business operations
   b. Changes to organization

**Figure 4.4-28. Content Format for a System Level Concept of Operations**

#### 4.4.4.2.1. Operational Services and Environmental Description

The OSED is a comprehensive, holistic Communications, Navigation, and Surveillance (CNS)/Air Traffic Management (ATM) system description. It describes the services, environment, functions, and mechanizations that form a system's characteristics.

"What Is a System?" A system (as defined in Section 2.1) is:

> *An integrated set of constituent pieces that are combined in an operational or support environment to accomplish a defined objective. These pieces include people, hardware, software, firmware, information, procedures, facilities, services, and other support facets.*

The 5M Model, illustrated in Figure 4.4-29, graphically represents this system view. Useful system descriptions exhibit two essential characteristics: correctness and completeness. Correctness in a system description means that the description accurately reflects the system with an absence of ambiguity or error in its attributes. Completeness means that no attributes have been omitted and that the attributes stated are essential and appropriate to the level of

detail called for in the description. System descriptions that include all 5M Model elements achieve these two characteristics.

The 5M Model states that there are five basic integrated elements in any system. These elements are (1) the functions that the system needs to perform; (2) the human operators and maintainers; (3) the equipment used in the system, composed of the hardware and software; (4) the procedures and policies that govern the system's behavior; and (5) the environment in which it is operated and maintained.

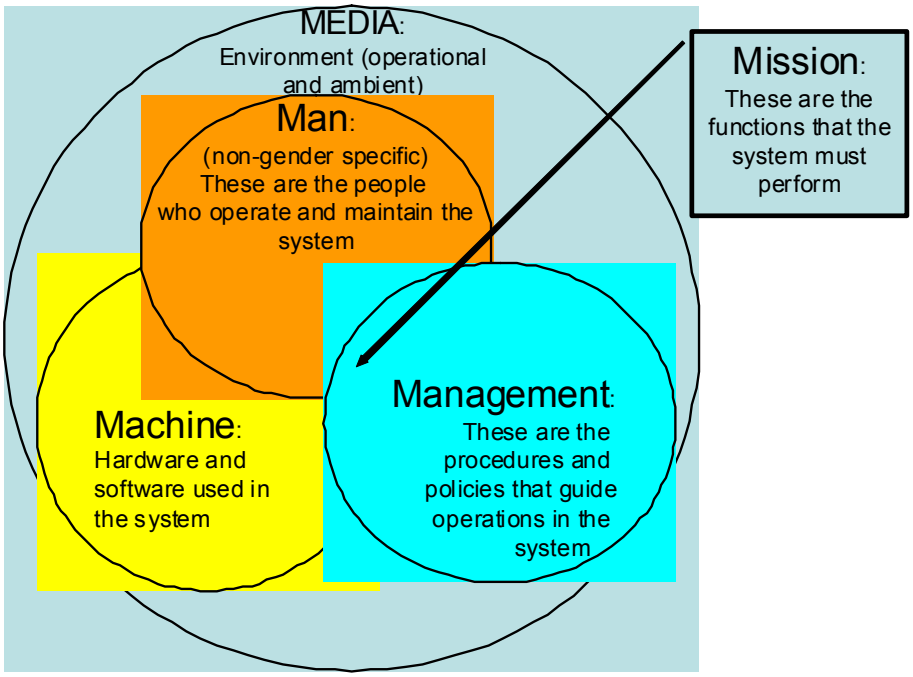



**Figure 4.4-29. 5M Model**

RTCA/DO-264, Appendix C, contains detailed guidelines for the OSED for use as a starting point. For the purposes of SE in the FAA, these guidelines were tailored. It is recommended that an OSED have, at minimum, the information in Figure 4.4-30.

1. Operation Service Description: This section of the OSED is used to summarily describe the air traffic services and operational context of the new capability. This section describes the new air traffic service from an operator's viewpoint.
2. Functional description or architecture: This section describes the functions and Functional Architecture in accordance with Functional Analysis.
3. Procedures: This section describes the existing and new procedures and policies that govern the system's operation or maintenance and includes:
   a. Operational requirements and regulations, including separation minima
   b. Deployment requirements
   c. Operational scenarios
4. Human elements of the system: This section describes the operators and maintainers of the system, including information regarding:
   a. Anthropometric requirements
   b. Training requirements
   c. Specific skill set requirements
   d. Human-system integration requirements
5. Equipment and software: This section describes any known hardware and software that is required for system operation. This section, in particular, may not be appropriate in the early stages of development.
6. Environment description: This section is an expression of the various conditions in which the system is operated, including:
   a. Operational: The operational environment includes factors, such as traffic density and flow, flight phases, traffic complexity, route configuration, type of control, use of visual or instrument flight rules, etc.
   b. Ambient: The ambient conditions refer to visual and instrument meteorological conditions, altitudes, terrain elevations, and physical conditions, such as electromagnetic environment effects, precipitation, icing, etc.
7. Nonfunctional requirements: This section describes any other Requirements that are not covered in the other sections and includes, but is not limited to, the following:
   a. Time constraints
   b. Information exchanges
   c. Exception handling

**Figure 4.4-30. Guidelines for an Operational Services and Environmental Description**

#### 4.4.4.3. Concerns/Issues

Appendix D contains guidance on concerns/issues as a product of Functional Analysis.

#### 4.4.4.4. Tools/Analysis Requirements

Tools/Analysis Requirements for performing Functional Analysis throughout the remainder of the program's lifecycle needs to be provided to the Integrity of Analyses process (Section 4.9).

#### 4.4.4.5. Planning Criteria

Any Planning Criteria necessary for performing Functional Analysis throughout the remainder of the program's lifecycle needs to be provided to the Integrated Technical Planning process (Section 4.2)

#### 4.4.4.6. Constraints

Constraints on trade studies that surface as a result of performing Functional Analysis are to be provided to the Trade Studies process (Section 4.6).

### 4.4.5. Functional Analysis Tools and Techniques

Contractors working for the FAA may choose to employ structured analysis models rather than the FFDs preferred by FAA. To facilitate communication between FAA system engineers and these contractors, it is recommended that FAA system engineers understand these models in order to engage in technical conversations with contractors who employ them.

#### 4.4.5.1. Tools

Analysis tools may include but are not limited to general SE and design/simulation aids. Because requirements represent the basic thread through SE, Functional Analysis data shall be interoperable with requirements definition information. The results of the Functional Analysis process shall be captured in order to modify system requirements and other derived products.

The selection of tools shall ensure that the data is transportable and able to be integrated with other related Functional Analysis results. A list of tools that may be used to perform Functional Analysis is available on the International Council on System Engineering Web site (www.incose.org).

#### 4.4.5.2. Techniques

In addition to the techniques described in "Task 2: Organize Functions Into Logical Relationships" (Paragraph 4.4.3.2), this paragraph covers several alternative approaches that FAA system engineers may come in contact with from organizations that apply techniques other than FFDs. These techniques are provided in order to cover two issues: (1) cases in which time line sequence diagrams and FFDs do not adequately address FAA needs, and (2) cases in which contractors use these techniques to perform Functional Analysis, and the FAA engineers need to understand what they mean. The alternatives include the following:

- Hierarchical functional block diagramming
- Modern structured analysis
- Models and simulation
- Thread analysis
- Object-oriented analysis (OOA)

##### 4.4.5.2.1. Hierarchical Functional Block Diagramming

By listing the functions for an expansion of one block on a higher-tier block diagram, the engineer is actually engaging in function outlining, which is equivalent to hierarchical block diagramming. Rather than building sequences of functions in FFDs, the engineer may build an indentured list or hierarchy of functions, where in order to accomplish a particular function, it is necessary to complete the immediate lower-tier functions first.

Generally, this process is easier to follow than a sequence-oriented model. Thus, the question arises: Why build a sequence- or flow-oriented diagram when the goal is a hierarchical physical architecture? It is easier for system engineers to move from a hierarchical functional diagram to a hierarchical architecture diagram than from a sequence-oriented functional diagram to a hierarchical physical architecture diagram. However, therein lies the problem. Engineers employing hierarchical functional diagrams often define a functional architecture based on the last physical architecture they worked on, which generally causes a one-to-one correspondence between their functional and physical architectures. The danger in this approach is the potential for the engineer to fail to consider all of the alternative implementations of the needed functionality and, subsequently, miss opportunities to take advantage of new technology. It is more difficult to move from a sequence-oriented functional model to a hierarchical physical architecture diagram; however, that difficulty encourages a more comprehensive examination of methods to implement exposed functionality.

#### 4.4.5.2.2. Modern Structured Analysis

Modern structured analysis offers a more free-form analytical environment than the block-oriented models. The modern structured analysis model is constructed using DFDs that feature bubbles rather than blocks, a data dictionary (DD), and process specifications (p-spec).

##### 4.4.5.2.2.1. Data/Control Flow Diagrams and Context Diagrams

Data/control flow diagrams (D/CFD) graphically model the processes that transform data/control in a system. These diagrams model the system's work as a network of activities that accept and produce data/control messages. Alternatively, they are also used to model the system's network of activities as work accomplished on a processor. Each successive level of D/CFDs represents the internal model of the transformations contained in the previous level of D/CFDs.

The context DFD—the ultimate DFD—consists of one bubble depicting the system connected to terminators drawn as blocks and named to identify the external inputs and outputs of the system. The bubble of the context DFD is decomposed to expose more detailed needs of the system. The lower-tier DFDs, of which there may be hundreds or even thousands for a complex problem, consist of only four objects:

- Bubbles (drawn as simple circles) identify needed computer processing. Bubbles have functionality that may be further decomposed in a lower-tier DFD or defined in a p-spec. P-specs are written only for the lowest-tier bubbles in the diagrams. Needed product behavior may be explored and illustrated by structured English, tables, or state diagrams within the p-spec.
- Directed line segments (arrows) show the flow of data between the bubble and temporary data stores.
- Temporary data stores (represented by a pair of parallel lines) identify a need to temporarily store data created in a bubble or applied from outside the system.
- Data sources and external inputs are represented by rectangles.

Figure 4.4-31 illustrates the application of DFDs and the top-down decomposition process used to produce a system model.

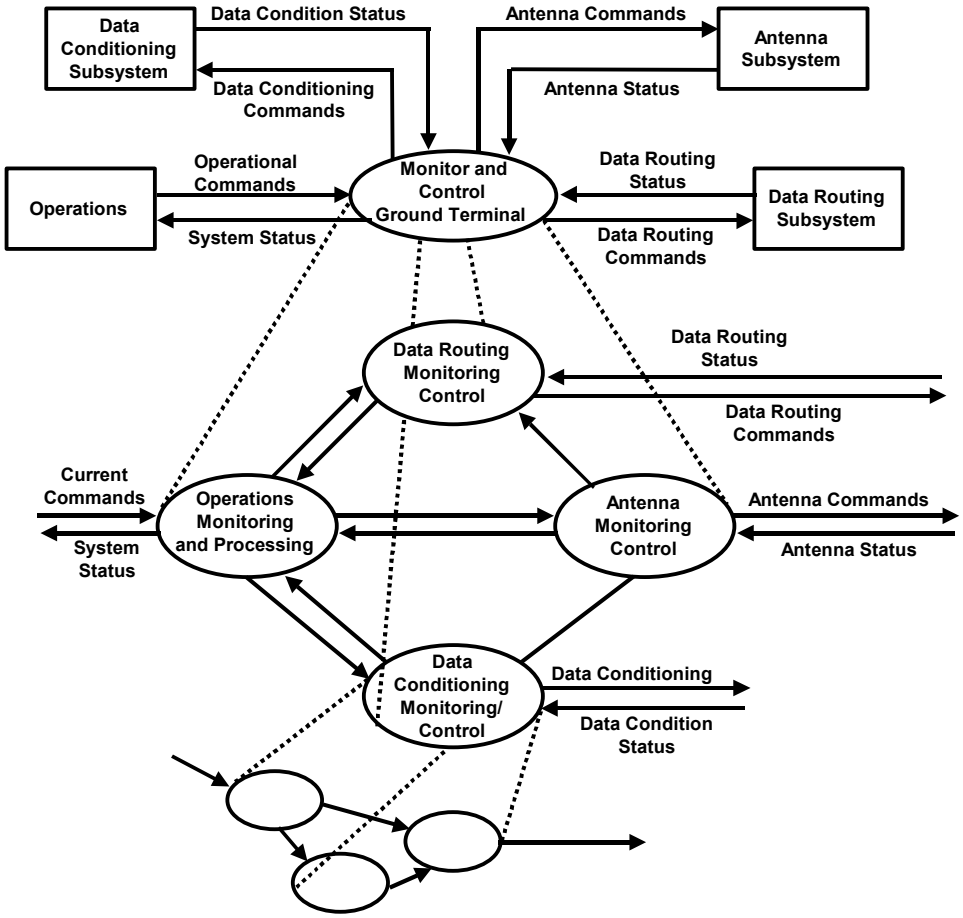



**Figure 4.4-31. Data Flow Diagram**

Building a system model by interviewing users usually begins with the processes defined at the primitive level and data defined in forms and manual files. Figure 4.4-32 illustrates part of a model built from user interviews. After building the model, the next task is to organize the data flows logically and then collapse the lower-level functions into higher-level functions. Figure 4.4-33 illustrates a logically organized version of the model built from interviews.

The DFD function titles, when wrapped in “shall statements,” become requirements statements.

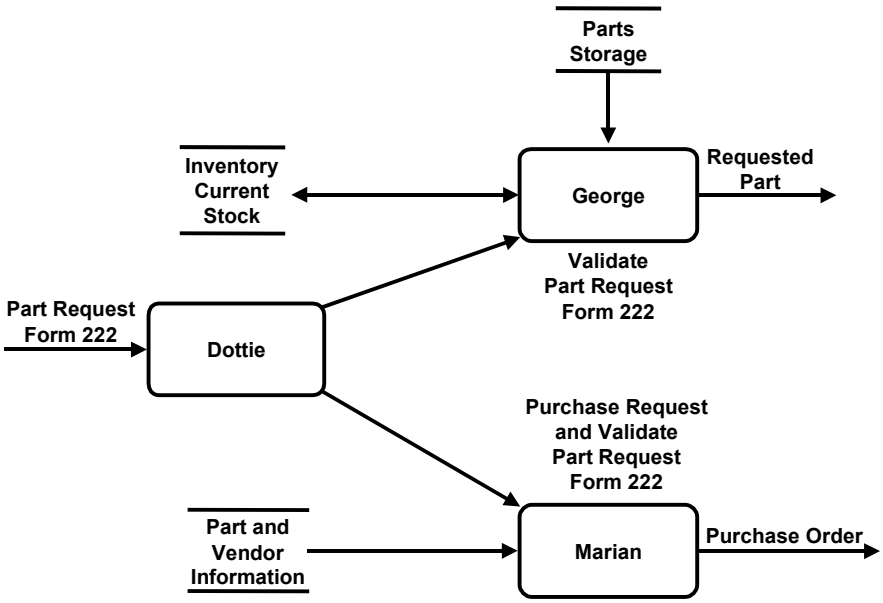



**Figure 4.4-32. Primitive Data Flow from User Interviews**

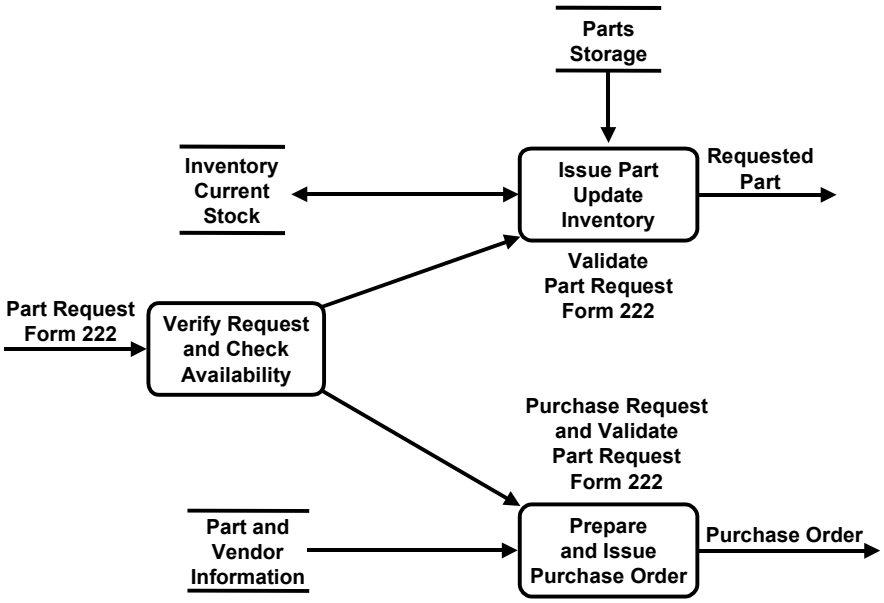



**Figure 4.4-33. Logically Organized Data Flow**

#### 4.4.5.2.2.2. Data Dictionary

DDs are documents that provide a standard set of definitions of data flows, data elements, files, databases, and processes for a specific level of system decomposition. These documents aid communication across the development organization. The DD also defines data items

mentioned in the transformation specifications. For every data line and every data store illustrated on every DFD, a unique line in a DD that clearly defines the data item is required.

A DD defines the content of each data item, table, and file in the system. P-specs describe the capabilities that each process is required to provide. The specifications may be written in structured English and/or in the form of decision tables and decision trees. State diagrams graphically depict the logical states that the system may assume. Associated process descriptions specify the conditions that require fulfillment for the system to transition from one logical state to another.

When working from a set of customer documents, a top-down approach is used to decompose customer-defined processes. As each process is decomposed, so is the data. Only the data that a process requires to produce the specified outputs is documented in a DD. Functional decomposition usually proceeds to a level where the requirements for each lower-level function are stated on one page or less (i.e., the primitive level). Interaction with the customer may be necessary to decompose and define data elements at lower levels.

#### 4.4.5.2.2.3. Process Specifications

For every lowest-tier bubble in the DFD analysis, it is necessary to write a p-spec that contains the p-spec for the bubble. This specification may be phrased in normal English text, structured English that follows a particular computer tool syntax, tables, state diagrams, or any combination of these constructs. P-specs, at the primitive level, when wrapped with “shalls” subsequently become requirements statements.

#### 4.4.5.2.2.4. State Transition Diagram

After the DFDs and DD are complete, the next step is to identify the various states the system may assume and to produce diagrams depicting how the system transitions between states. It is suggested that a top-down approach, such as a state transition diagram (STD), be used to identify various states of the system, working down through the subsystem.

An STD is a graphical model of the dynamic behavior of a system—it is a sequential state machine that graphically models the time-dependent behavior of a control transformation. Figures 4.4-34 and 4.4-35 are examples of STDs for system and subsystem functions. Descriptions of how the system transitions from one state to another become “shall statements” in the requirements document.

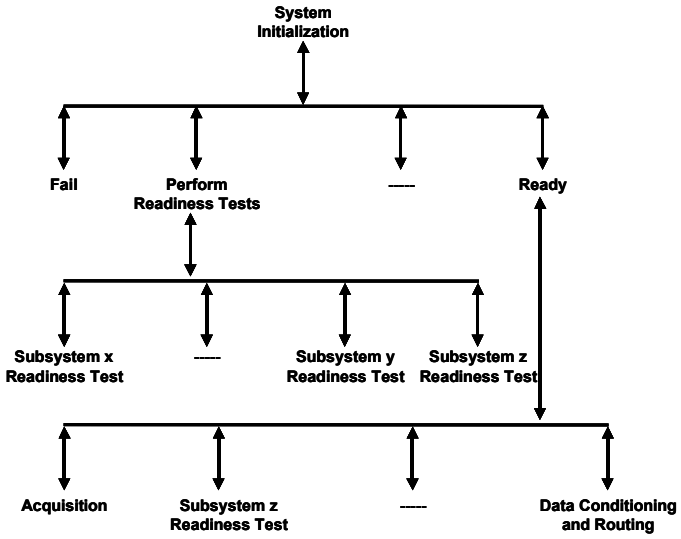



**Figure 4.4-34. State Transition Diagram for System Functions**

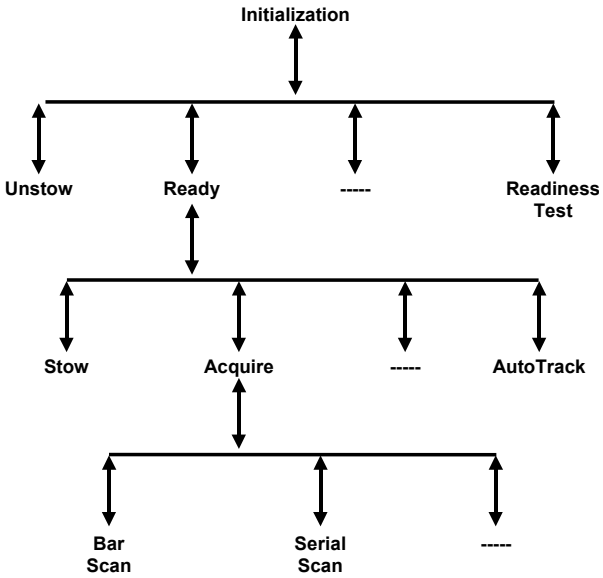



**Figure 4.4-35. State Transition Diagram for Subsystem Functions**

#### 4.4.5.2.3. Hatley-Pirbhai Extension to Modern Structured Analysis

Because the traditional modern structured analysis process has proven inadequate for modeling real-time systems, the Hatley-Pirbhai Extension to modern structured analysis was created. This model extended the requirements model of modern structured analysis process to include an additional construct called a control flow diagram (CFD)—an augmentation of the corresponding DFD that has control as well as data processing functions. The CFD has the same bubbles as its companion DFD. The data lines that join the bubbles on the DFD are related only to the data associated with processing needs. The data lines shown on the CFD are only those data items related to control functions. This model may be considered a special

case of modern structured analysis and is particularly useful when the problem entails difficult control problems.

#### 4.4.5.2.4. Models and Simulation

Models are abstractions of relevant characteristics of a system that are used to understand, communicate, design, and evaluate (including simulation) the system. They are used before the system is built and while it is being tested or in service. A good model has essential properties in common with the system/situations it represents. The nature of the properties it represents determines the uses for the model. A model may be functional, physical, and/or mathematical.

For complex system problems, it is necessary to analyze and design a number of different systems, each of which is represented by a specific model. The different models permit individual investigation of different aspects. These different modeling perspectives are incrementally constructed and integrated in a unified description (system model) to maintain a holistic system perspective from which the emergent properties of the system are deduced and verified.

The system model emphasizes the interactions of the objects in the context of the system, including the objects in the environment. Object semantics represent the components of a system, their interconnections, and their interactions when they are responding to the stimulus from the objects in the environment. These object semantics are partitioned into a static as well as dynamic modeling representation that describes the system's structure and behavior, respectively.

In this sense, the models embody the decisions made over the different steps of the Lifecycle Engineering process (Section 4.13). The models are developed as part of the decisionmaking process and support the evolution of the system design process as well as the iterative nature of the engineering in an environment where changes and enhancements to the models are managed in a controlled manner.

#### 4.4.5.2.5. Thread Analysis

One major challenge to Functional Analysis is the development of software that implements the desired behavior of the system. Because system behavior is primarily implemented in software, a critical issue in system development is "how system engineers interact with the software engineers to ensure that the software requirements are necessary, sufficient, and understandable." This problem is addressed at the practitioner level, and experience has shown that the approach of passing paper specifications between systems and software developers does not yield satisfactory results.

Stimulus-condition-response threads are an excellent way to control the software development process, including translation from system to software requirements, design verification, review of software test plans, and integration of software and system testing. The threads enumerate the number of stimulus-condition-response capabilities to be tested. Threads also tie to performance requirements. Experience in the past 20 years on a variety of thread versions shows that such approaches are both feasible and effective.

##### 4.4.5.2.5.1. The Use of Threads

System and software engineers shall work together to identify the system-level threads and the subset of the threads that the computer system supports. In this context, a thread consists of a system input, system output, description of the transformations to be performed, and conditions

under which the transformations hopefully occur. Such threads may be represented textually or graphically in a variety of ways, some of which are supported by tools. The following guidelines apply to the use of threads:

- The threads satisfy the need for efficient communication between system and software developers
- The identification of a thread from input to output allows the identification of the subthread to be allocated to the processing subsystem and, hence, software
- The description of stimulus-condition-response threads eliminates the ambiguities found in current specifications
- The description of threads is inherently understandable, particularly if provided in some graphical format
- The use of such threads aids in evaluating the impact of proposed changes

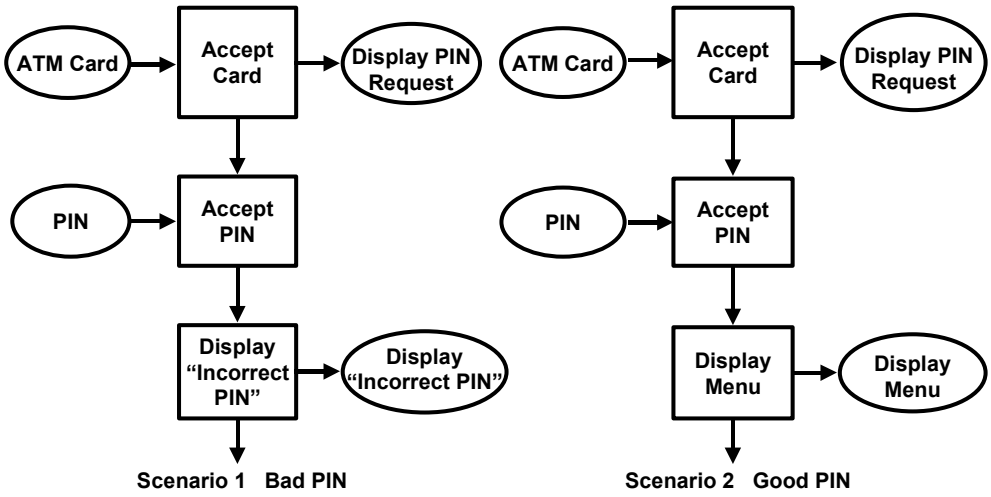
In the following steps, the development of software requirements is evolutionary, starting with allocation of processing requirements to a processing system and ending with publication and review of the software requirements.

#### 4.4.5.2.5.2. Step 1: Deriving the System-Level Threads for Embedded Systems

No matter how the system description is developed, even if it is no more than the identification of system functions for different modes of operation, system inputs and outputs shall be identified in order to anchor the specification to reality. This process starts with the initial scenarios that describe the system's intended operations, which may be rewritten into the form of stimulus-condition-response threads.

To illustrate, consider the bank automated teller machine (ATM) system, which, by processing ATM cards and personal identification numbers (PIN), enables customers to perform banking transactions. Figure 4.4-36 presents two top-level scenarios that describe the top-level behavior of the ATM system when presented with an ATM card and a PIN. The two scenarios are PIN is accepted and PIN is rejected. From the scenarios or the integrated behavior, the stimulus-condition-response threads are identified. This set of threads may be specified in a number of notations. Figure 4.4-37 presents the stimulus-condition-response threads in a functional format. Note that the conditions for each of the threads are to avoid ambiguity. These conditions are a combination of two factors:

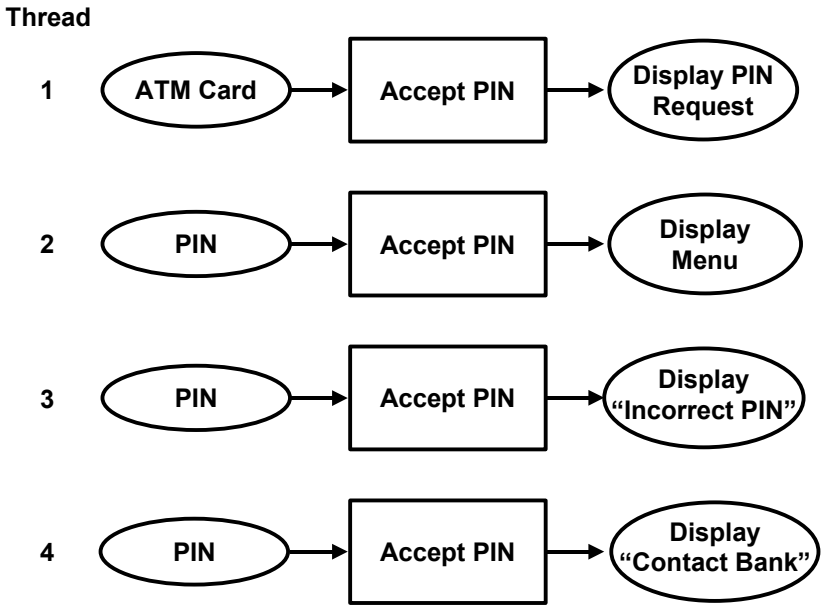
- The mode of the system, which determines which kind of input is expected
- The combination of values of the system state information and the contents of the input




**Scenarios**

**Figure 4.4-36. Top-Level Scenarios in Thread Analysis**

Thus, a correct PIN yields a menu, while an incorrect PIN results either in a message to "try again" or the machine "swallowing" the card, depending on the mode of the system. These conditions require that a thread be associated with the conditions in order to make them testable. To show the conditions explicitly, the "Accept PIN" function is decomposed to show its input-output behavior under different conditions.



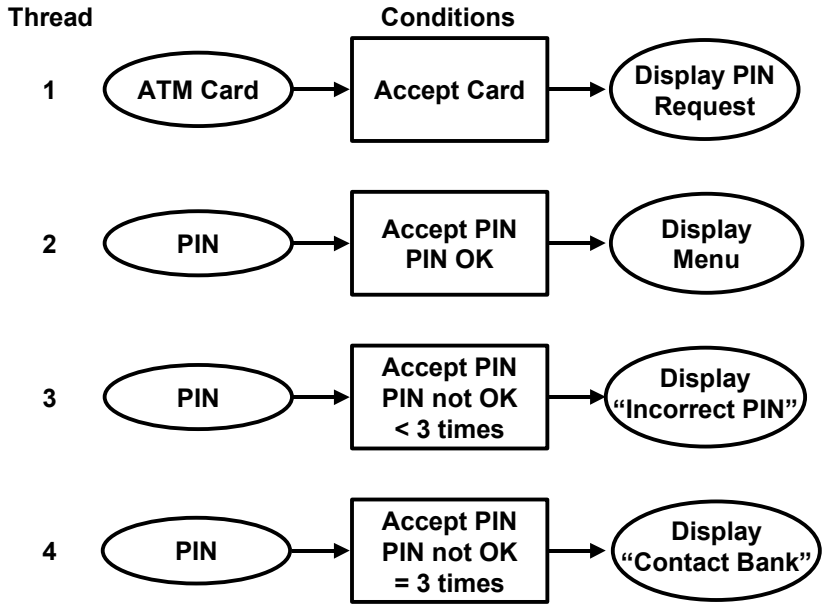

**Sample ATM Threads**

**Figure 4.4-37. Stimulus-Condition-Response Threads**

#### 4.4.5.2.5.3. Step 2: Decomposing the Threads to the Subsystem

It is necessary to decompose functions to the level where functions are uniquely identified by their requirements. This process is illustrated in Figures 4.4-36 through 4.4-40. In Figure 4.4-36, the top-level scenarios are defined; in Figure 4.4-37, examples of stimulus-condition-response threads are provided; in Figure 4.4-38, stimulus-condition-response threads are defined with their conditions; in Figure 4.4-39, the threads are defined in condition format; and in Figure 4.4-40, the system-level function "Accept PIN" is decomposed into functions to read the card (allocated to a card reader) and the functions and conditions allocated to the computer. Usually, most or all of the conditions are allocated to the computer system, with mechanical functions allocated to the other less intelligent components. Hence, most of the system threads yield a thread, with conditions, allocated to the computer subsystem, of which the majority is then allocated to the computer software with the software driving the computer hardware requirements. Thus, there is a direct traceable relationship between the system level, computer system level, and software level of requirements.

Figures 4.4-38 through 4.4-40 identify the difference between the system and computer system threads. The system uses a card reader component to read the card, a terminal component to accept push button inputs from customers, and a processor component to provide the intelligence to process the requests. Note that this process results in the requirement for the computer system to direct its threads to translate "card info" and "PIN info" to various output displays.




**Threads With Conditions**

**Figure 4.4-38. Threads Identified with Conditions**

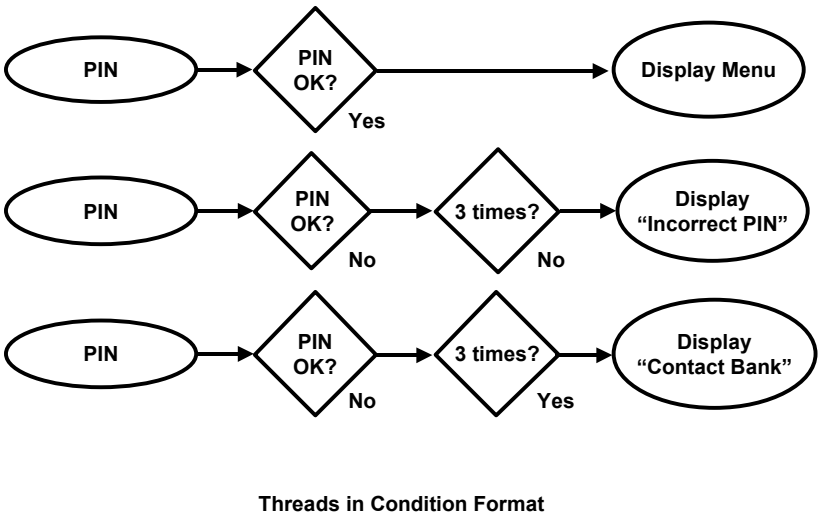



**Figure 4.4-39. Threads in Condition Format**

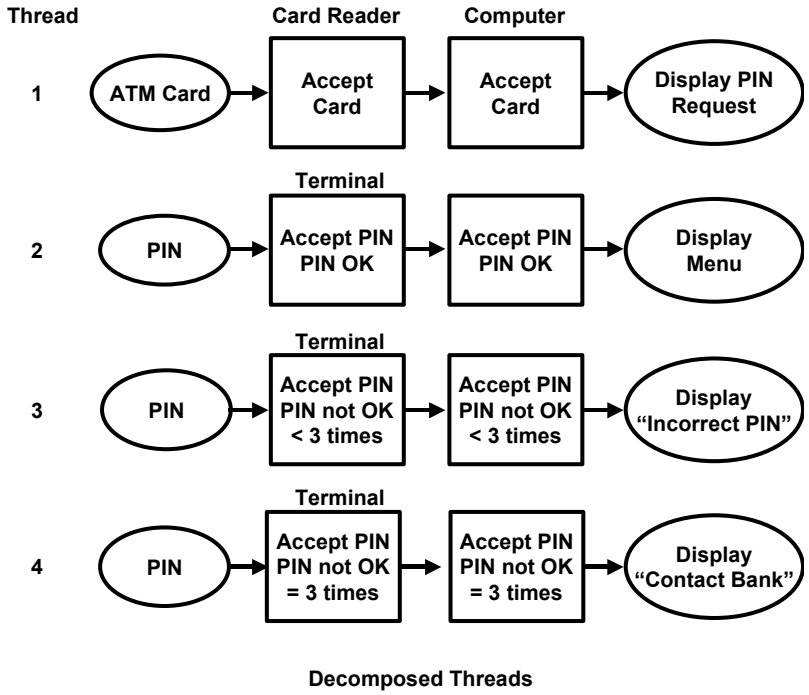



**Figure 4.4-40. System Level Functions Decomposed**

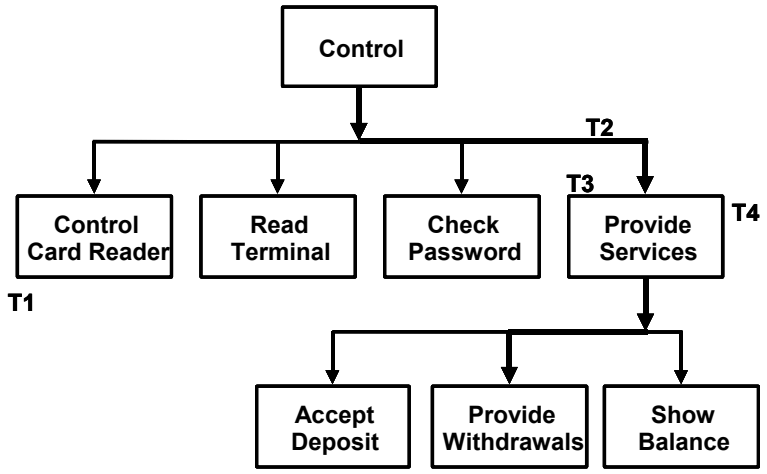
#### 4.4.5.2.5.4. Step 3: Reviewing the Requirements and Design

Irrespective of the requirements format, computer system-level threads need to be traceable through the document. If a thread is untraceable, it signifies an omission in the requirements. If additional threads are identified that do not deal with interface designs or computer system level fault detection/recovery, such threads may represent unnecessary processing; and, if so, it is

necessary to omit them. To illustrate, Figure 4.4-41 presents a simple software design in which a top-level program control calls lower-level units of code to execute operations.

Threads 1 through 4 are traceable through this design, thus validating it. The same approach works when an object-oriented design presents a number of objects implemented as independent software processes. When software design occurs that divides the overall software into computer software configuration items (CSCI), computer software components (CSC), and computer software units (CSU), the above process of decomposing and allocating the system-level threads into the components is repeated for each level of component. Again, if a thread is untraceable, it signifies an omission in the design. The system engineer traces the allocated system requirements to the software requirements review, which is followed by the CSCIs, CSCs, and CSUs for the software preliminary and critical design reviews.

If the software designers trace the computer system to software design threads as part of the requirements satisfaction demonstration, then the system engineer need only verify the completeness of the traceability. Tools strengthen the reliability of such a traceability evaluation. If the software designers do not perform this activity, then it is recommended that a joint team of system and software engineers perform the tracing to verify the design in preparation for the design reviews. In any event, the software-level threads shall be identified in order to provide systematic test planning.




**Mapping S/W Threads onto a S/W Design**

**Figure 4.4-41. Tracing the Threads**

#### 4.4.5.2.5.5. Step 4: Tracing the Threads to the Test Plans

It is recommended that the collection of threads be exercised by the collection of tests outlined in the test plan—at the software level, computer system level, and the system level. This collection may be represented by a database and displayed in a cross-referenced matrix showing the relationship of system to software thread, software thread to software design threads, and threads to test cases at the various levels of integration, enabling the tools to ensure that every level of thread is tested at some point.

It is strongly recommended that the software threads be used to drive the test planning process using the concept of builds of software. For a system test, other components are added, and the system test threads are tested. For the ATM example, the difference is that in the software

only test, the software receives information in the format expected from the card reader; in the system test, the card reader component itself is used as the source of the data when an ATM card is the input.

This same approach is also used to construct the system-level test plans in a way that exploits the early availability of computer software that provides user-oriented capabilities. Thus, an early build of software may be integrated with a card reader to perform a test of Thread 1 through the system before the remainder of the software is developed. If the card reader is not available until later in the test cycle, then other threads may be tested first.

#### 4.4.5.2.5.6. Notation

Several kinds of notations may be used for tracking the threads, but these notations usually fall under requirements- and design-oriented notations. Requirements-oriented notations describe the inputs, conditions, and outputs, while the design-oriented notations describe the threads with respect to the major design elements. Because both eventually describe the same stimulus-condition-response information, their use is essentially equivalent (though the design-oriented notation is more useful for actually defining the builds of software).

#### 4.4.5.2.5.7. Conclusion

System engineers need to take the lead in constructing a sufficient process for system and software engineers to communicate, as it is the responsibility of the communicator to communicate in a language that the recipient understands. It is not feasible for system engineers to wait for software requirements methodologies to stabilize and accomplish this objective because software requirements and design techniques show no signs of stabilizing. The problem needs to be addressed within the existing context of multiple software requirements languages.

### 4.4.5.2.6. Object-Oriented Analysis

#### 4.4.5.2.6.1. Early Versions

Early models advanced by Yourdon (2000) focused on objects that encapsulated computer processing and data, thus ending the separate analysis of these two previously inseparable facets of any computer software entity and, thereby, providing a tremendous improvement in software analysis. The model encouraged problem space entry using objects that represented the physical entities in the problem and solution space. Functionality and behavior of the problem space was explored based on these objects; therefore, it was not possible to follow the concept of form follows function when applying it. Early OOA models may be effective in analyzing systems with heavy precedence but problematic when exploring unprecedented problems.

Most authors who supported early OOA encouraged identification of objects that reflected elements of the problem space, that these objects be linked and organized into major subject areas, and that they be followed by refining the objects by identifying object functionality in terms of a DFD and behavior using state diagrams. Note that it was not easy to begin with functionality or behavior, rather one had to explore functionality and behavior in terms of previously defined objects.

#### 4.4.5.2.6.2. Unified Modeling Language

##### 4.4.5.2.6.2.1. Background

The UML is a language for specifying, visualizing, constructing, and documenting the artifacts of software systems as well as for business process modeling.[1] The UML represents a collection of "best engineering practices" that have proven successful in modeling large and complex systems. Rational Software and its partners developed the UML, which is now an industry standard (www.omg.org/uml). It is widely supported, and there are numerous commercial packages available (www.incose.org) that may be used to develop UML-compliant models. These packages provide a collection of functionality ranging from purely drawing UML diagrams (low cost) to full round-trip engineering with model syntax checking and code generation (higher cost).

The principal benefit obtained from employing a standardized modeling language is that it provides a common framework for communicating system design and behavior between the organizations and various individuals, including users, architects/developers, and operators, involved with the system under development. Developing a model for an industrial-strength software system prior to its construction or renovation is as essential as having a blueprint for a building. Comprehensive models are essential for communication among project teams to ensure architectural soundness. As the complexity of the system increases, so does the importance of efficient modeling techniques.

The UML focuses on a standard modeling language, not a standard process. Although the UML has to be applied in the context of a process, experience has shown that various organizations and problem domains require a different process. The UML authors promote a development process that is use-case driven, functional architecture centric, iterative, and incremental. However, this specific development process is not required or enforced by the language. The UML merely provides the capability for:

- Model elements—fundamental modeling concepts and semantics
- Notation—visual rendering of model elements
- Guidelines—idioms of usage within the trade

Additionally, the UML provides extensibility and specialization mechanisms to extend core components. Though the UML is object-oriented by default, it is independent of particular programming languages.

#### 4.4.5.2.6.2.2. Development Artifacts

The decision regarding which diagrams to create is largely dependent upon the system under development. Focusing on the relevant aspects of the system is critical in the abstraction process. The UML provides a rich notation to describe the static and dynamic behaviors of the system through several diagrams. These diagrams provide complementary views of the system, which are then developed and used by the various stakeholders.

The UML diagrams fall into the four following groups: use-case diagrams, class diagrams, behavior diagrams, and implementation diagrams.

**Use Case Diagram:** A use-case diagram depicts one or more use-cases with its associated primary and secondary actors. An actor defines a role that a person plays with respect to the system. A use-case, by definition, yields an observable result of value to its primary actor. A

[1] OMG Unified Modeling Language Specification, Version 1.4, September 2001.

secondary actor may be invoked by the use-case and provides a service. Actors may have a primary role in one use-case and a secondary role in another use-case. Use-cases are particularly well suited for capturing requirements. Figure 4.4-42 is an example of a use-case diagram.

**Class Diagram:** A class diagram provides a static view of the system's classes and depicts the relationships between the various classes. A class is a fundamental construct in all object-oriented languages and includes the notion of data and functions that are logically grouped. Individual class diagrams may depict attributes (e.g., data) and operations (e.g., functions) with varying levels of detail as necessary.

**Behavior Diagrams:** Behavior Diagrams are used to depict the dynamic operation of the system and include statechart diagrams, activity diagrams, sequence diagrams, and collaboration diagrams. A statechart diagram typically describes all possible states that a particular object may inhabit and how the object's state changes with regard to external events. Activity diagrams describe sequences of activities in which an activity typically represents a real-world process. Sequence diagrams depict a time-ordered flow of events between classes and actors and frequently describe a complex interaction between a small number of classes. Similarly, collaboration diagrams depict a time-ordered flow of events between actors and classes using a different layout; they are frequently drawn at a more abstract level. Collaboration diagrams are well suited for identifying underlying design patterns. Figures 4.4-43 and 4.4-44 are examples of behavior diagrams.

**Implementation Diagrams:** Implementation diagrams include both component and deployment diagrams. A component diagram depicts the various components and their dependencies in which a component typically represents a physical module of code. Alternatively, a deployment diagram depicts the physical relationships between software and hardware components.

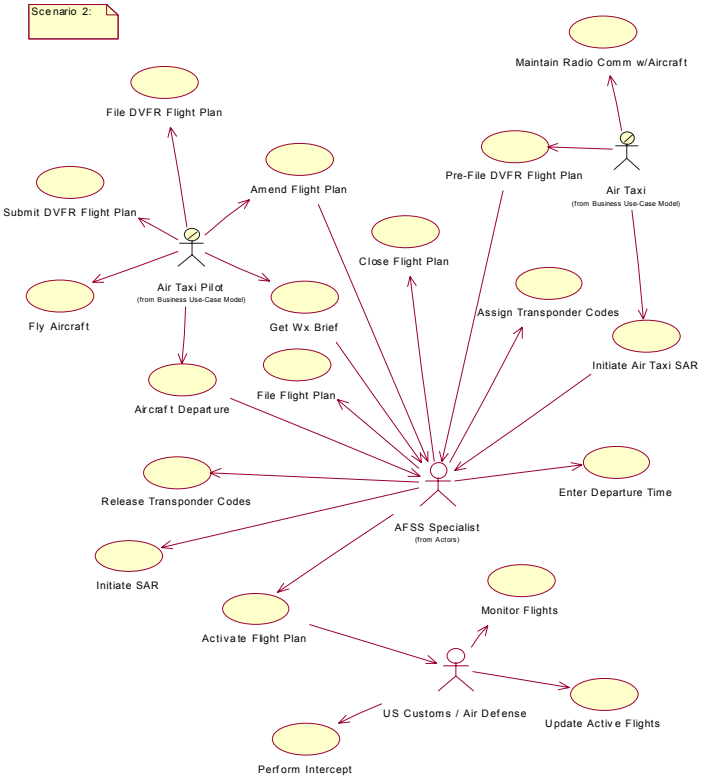



**Figure 4.4-42. Day Visual Flight Rules Pre-Filed Flight Plan Use-Case Diagram**

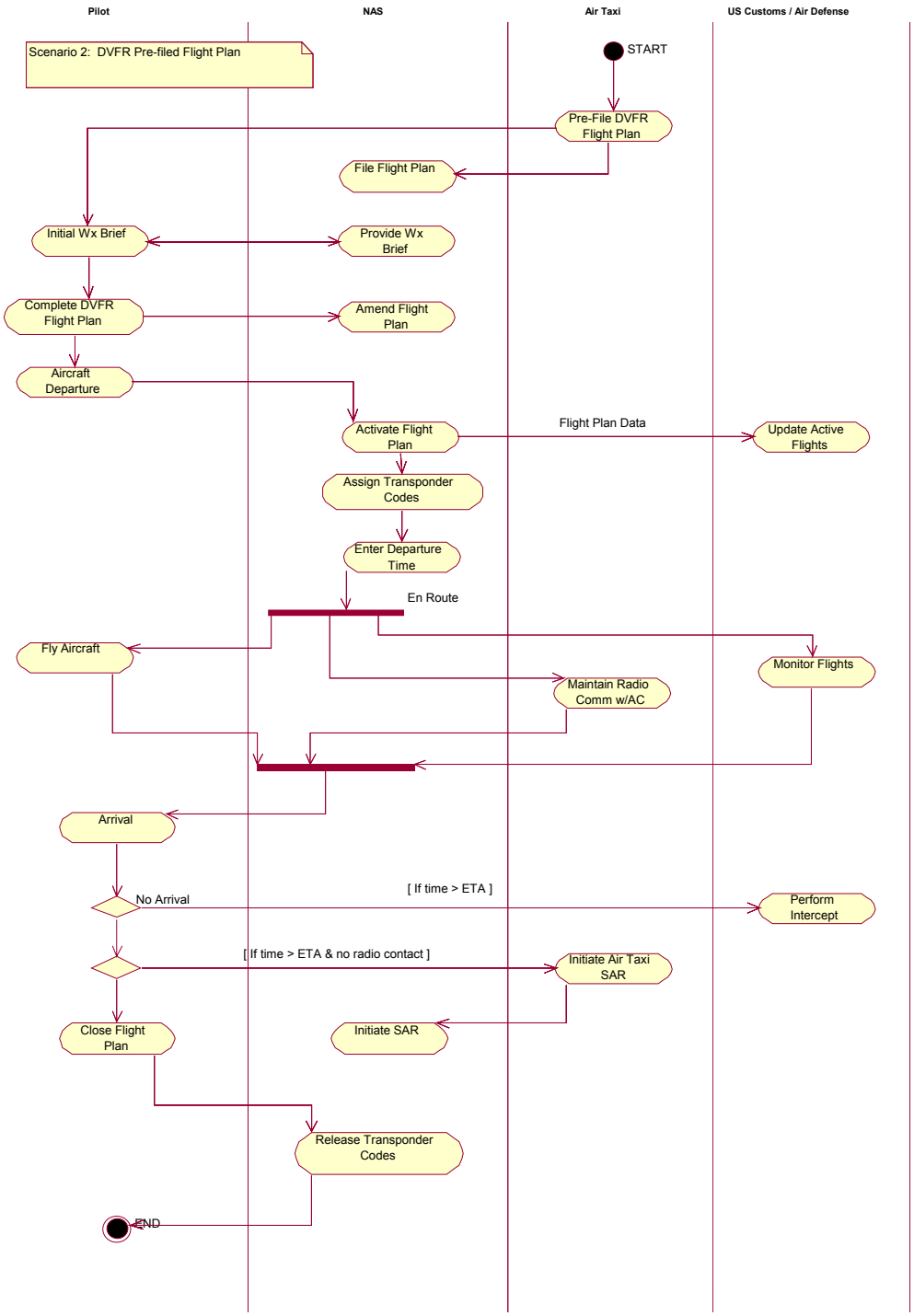



**Figure 4.4-43. Day Visual Flight Rules Pre-Filed Flight Plan Activity Diagram**

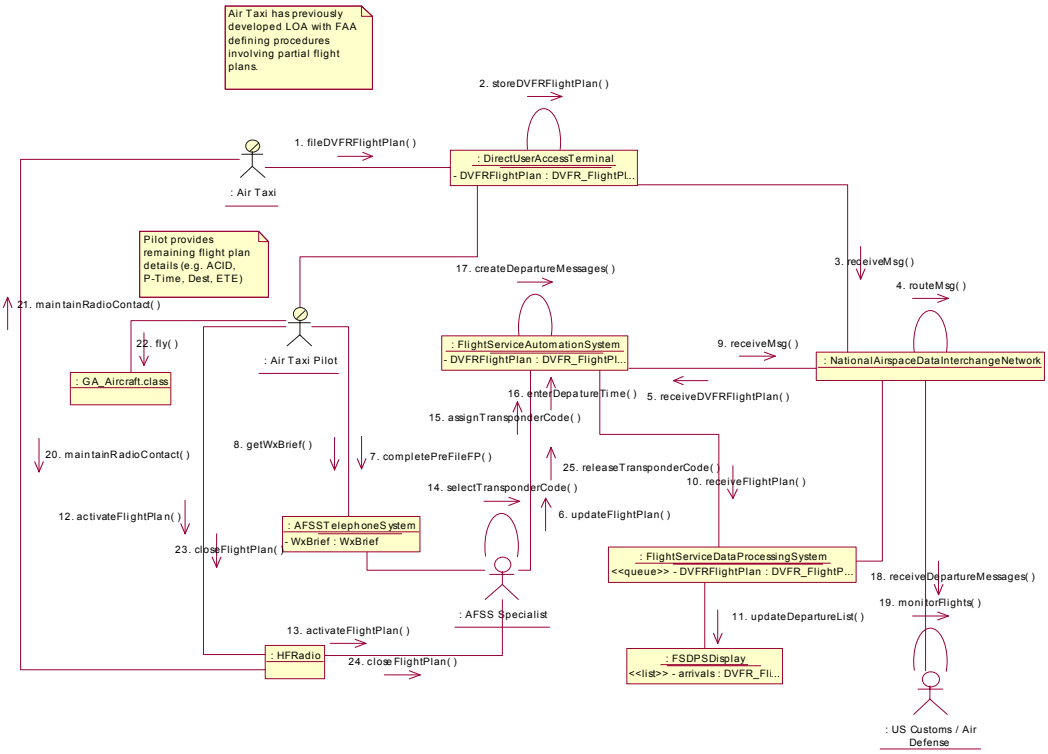



**Figure 4.4-44. Day Visual Flight Rules Pre-Filed Flight Plan Collaboration Diagram**

### 4.4.6. Functional Analysis Process Metrics

Candidate metrics used to measure the overall process and products of Functional Analysis include the following:

- Percent of analysis studies completed (schedule/progress)
- Depth of the functional hierarchy as a percentage versus the target depth
- Percent of performance requirements allocated at the lowest level of the functional hierarchy

Of the seven general measurement categories (see Table 4.1-1), the two that are applicable to Functional Analysis are Process Performance and Product Quality. In addition to the measures listed above, other candidate measures for Functional Analysis are provided in the table below. It is recommended that each effort tailor these measures and add other applicable project-specific measures to ensure the contribution of necessary information to the decisionmaking processes.

**Table 4.1-1. Candidate Measures for Functional Analysis***

| Schedule and Progress | Resources and Cost | Product Size and Stability | Product Quality | Process Performance | Technology Effectiveness | Customer Satisfaction |
|---|---|---|---|---|---|---|
| Achievement of specific milestone dates | Total effort compared to plan | Documentation of interfaces | Technical performance | Process productivity | Technology impact on product | Customer survey results |
| Test status | Resource utilization | Requirements | Defects | Process activity cycle time | Baseline changes | Performance rating |
| | | | Standards compliance | Defect containment | | |

*NOTE: The measures above are only general examples to indicate the type of information that might be included in the individual section measurement matrix.

### 4.4.7. References


1. Blanchard, B. "System Engineering Management. 2nd ed. New York, New York: John Wiley & Sons, Inc.
2. Blanchard, B., and W. Fabrycky. "Systems Engineering and Analysis." 2nd ed. Englewood Cliffs, New Jersey: Prentice Hall.
3. Defense Systems Management College. "Systems Engineering Fundamentals." Fort Belvoir, VA: Defense Systems Management College Press, 1999.
4. Department of Defense. Military standard system safety program requirements. MIL-STD-882, 1984.
5. Federal Aviation Administration. "NAS Modernization System Safety Management Program." Washington, DC, 2001 < http://fast.faa.gov/toolsets/index2.htm>
6. International Council on Systems Engineering. "Systems Engineering Handbook," Version 2.0, 2000
7. National Aeronautics and Space Administration. "NASA System Engineering Handbook," June 1995.
8. RTCA, Inc. "Guidelines for the Approval of the Provisions and Use of Air Traffic Services Supported by Data Communications." RTCA DO-264.
9. SAE. "Certification Considerations for Highly-Integrated or Complex Aircraft Systems." Aerospace Recommended Practice ARP-4754, 1996.
10. SAE. "Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment." Aerospace Recommended Practice ARP-4761, 1996.
11. Sage, Andrew B., and William B. Rouse, eds. "Handbook of Systems Engineering and Management." New York, New York: John Wiley & Sons, Inc., 1998.
12. Yourdon, Edward. "Modern Structured Analysis."

## 4.5 Synthesis

Synthesis simply put is design. Design is the creative process by which we develop solutions to requirements, thereby employing System Engineering (SE) to satisfy operational needs. In the Synthesis process, we first conceive and then later refine specific designs that will serve to satisfy operational needs.

The purpose of Synthesis is to define design solutions and identify systems that will satisfy the requirements baseline. Synthesis translates the requirements, as set in context by the Functional Architecture, into the design architecture, consisting of the Physical Architecture with its associated technical requirements. The resulting architecture provides an arrangement of system elements by designing their composition and interfaces, both internal and external. Additionally, the design architecture incorporates environmental, technical and other constraints.

Synthesis is seldom, if ever, a one-step process, but rather accomplished many times over the life of a project in response to many things, such as newly evolving technology, test data from the present or previous designs, changes in requirements from the user, changes in the price or availability of components, feedback from the field once a system is deployed, and so on. As with all SE functions, different objectives and activities exist within different phases of the acquisition process.

### 4.5.1 Introduction

The Synthesis process is only a portion of the overall SE discipline, with other processes occurring before, during and after. Synthesis also leverages the efforts conducted under various Specialty Engineering (Section 4.8) disciplines through concurrent engineering. Accordingly, Synthesis requires a number of inputs into the process in order to achieve the anticipated results, or outputs, of the process. See Figure 4.5-1.

Synthesis is conducted to translate the requirements (based on the Functional Architecture) into a Physical Architecture by defining the system elements. Those elements are then refined and integrated into the system's physical configuration, which satisfies the functional and performance requirements. This process relies heavily on prior establishment of clearly defined, documented, and validated requirements.

When entering the Synthesis process it is not assumed the entire requirements set associated with the functional area under consideration is achievable within the cost and schedule constraints. When entering the Synthesis process, it is assumed that all requirements associated with the functional area under consideration have been validated in accordance with Validation and Verification (Section 4.12). The engineers involved in the process are challenged to find the best possible solution that will optimize achievement of the requirements baseline for the functional area under consideration. This will require close and continuous coordination with Requirements Management (Section 4.3) and Functional Analysis (Section 4.4).

Success of the Synthesis or design process relies upon a structured and disciplined approach to achieving the desired outcomes. The Synthesis outputs will naturally emerge from taking the appropriate steps during the conduct of design. Conducted properly, Synthesis will define the build-to characteristics of the system or system elements. The Configuration Items (CI) are established and defined during Synthesis. At each level of the resulting design architecture, the requirements and interfaces must be verified. The Synthesis process must not only identify technically feasible and programmatically achievable design alternatives, but those alternatives must be well analyzed, documented, and finally placed under disciplined management.

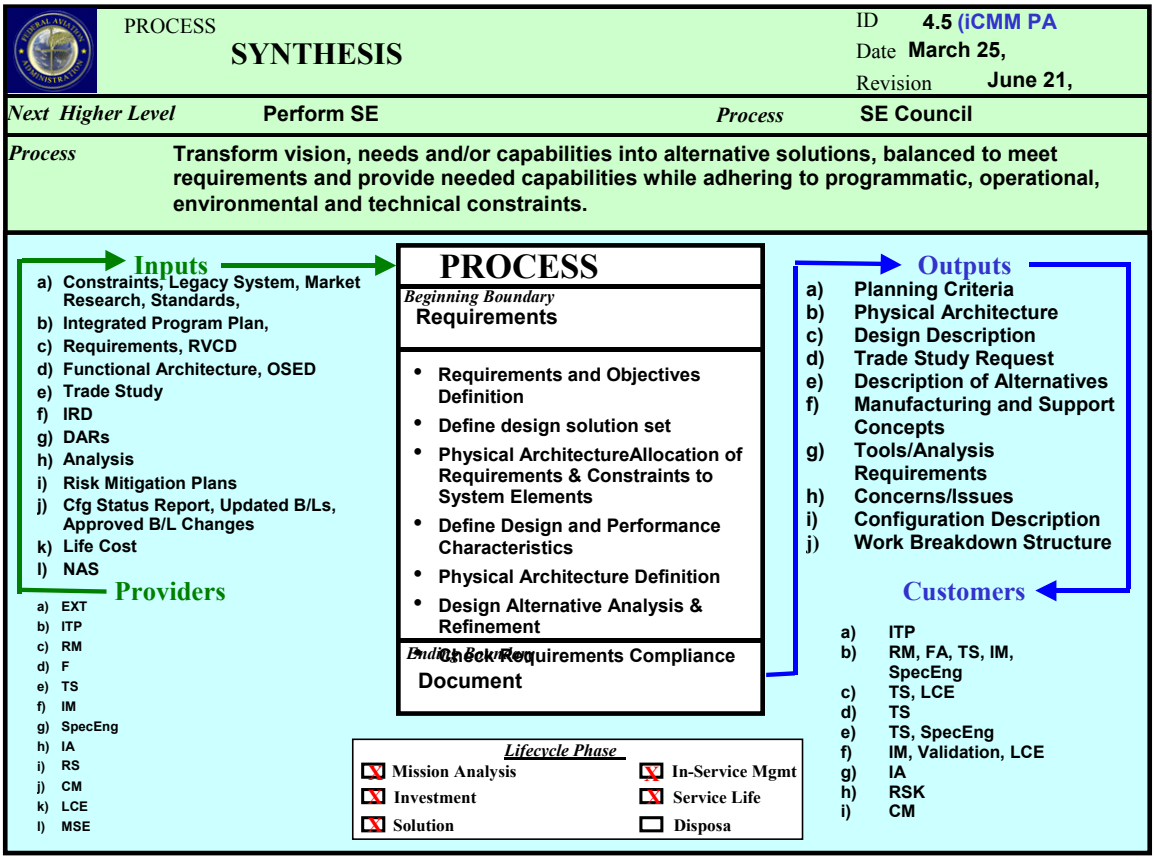



**Figure 4.5-1: The Synthesis Process; Inputs, Tasks, and Outputs**

### 4.5.2 Process Inputs

The Synthesis process starts at the conclusion of a number of preceding key SE steps, as illustrated in Figure 4.5-2. These SE processes result in a number of outputs that will serve as necessary inputs to Synthesis.

Like Synthesis, the processes preceding it are not necessarily one-step processes. Each may undergo a number of iterations through the given process before the output is ready for the next process to commence. Additionally, the Requirements Management (Section 4.3) and Functional Analysis (Section 4.4) processes are tightly coupled, and a few iterations through these processes will occur before readiness to proceed into Synthesis.

Once Synthesis commences, it will be as an iterative process, at times initiating iteration back through Requirements Management, known as the requirements verification loop. Synthesis might also at times initiate iteration back through Functional Analysis, known as the design loop. During these iterative loops through preceding processes, the requirements baseline and or Functional Architecture are constrained and refined to optimize the potential for viable design alternatives. This ensures that the Functional Architecture and requirements at lower levels of the Physical Architecture reflect the envisioned design.

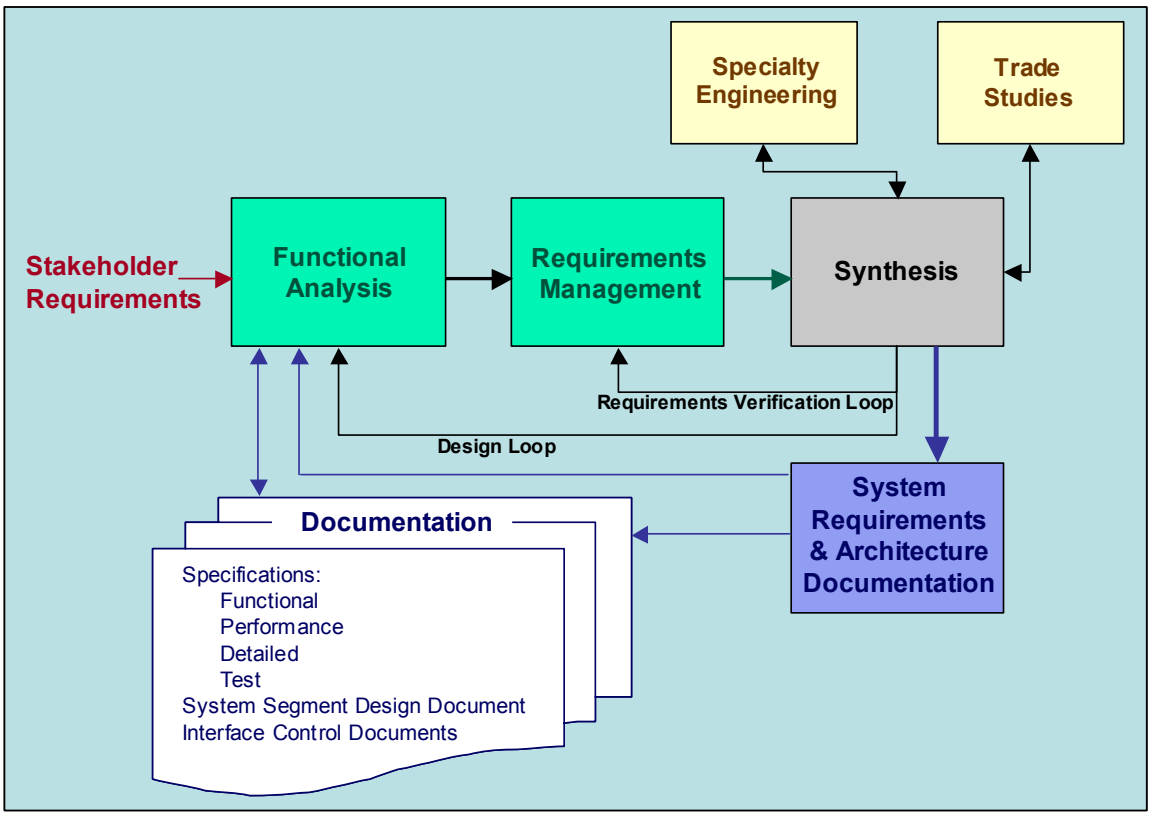



**Figure 4.5-2. Requirements and Architecture Definition**

### 4.5.2.1 Initial Inputs

The inputs resulting from the previously conducted SE processes are known as the initial inputs, as they serve to initiate Synthesis. They must be available prior to commencing system design.

#### 4.5.2.1.1 Functional Architecture

During Functional Analysis (Section 4.4), the high-level functions are decomposed to lower-level functional groups or areas that can be satisfied by system design alternatives. The Functional Architecture must describe the functional arrangements and sequencing of subfunctions resulting from decomposition. The Functional Architecture does not consider design solutions, but only tasks that must be performed by the solution(s). Synthesis, by contrast, will consider the grouped and decomposed functions, or functional areas, in light of technically feasible and achievable solutions.

Functional Analysis will provide the design group the appropriate area of the Functional Architecture at which to begin the design process. This Functional Architecture is translated into an established requirements set that documents the problem, or set of problems to be solved by Synthesis. The problem for the design group is to identify and define a system or systems that will adhere to the prescribed Functional Architecture while meeting stakeholder requirements.

#### 4.5.2.1.2 Requirements Baseline

The user needs and system functions are translated into a set of definitive, prioritized, and measurable requirements for which the design group must provide a solution or solution set.

The established baseline will dictate the tasks the system(s) under design must perform through functional requirements. The baseline will also dictate how well the system(s) must perform its tasks through documented performance requirements. And finally, the requirements baseline will ensure system functional and performance compliance through measurable verification requirements.

Not only will information be needed regarding what the system must perform, how well it will be performed, and how performance will be measured, but the baseline will also establish the system's limitations. The requirements baseline will contain the constraint requirements levied on potential solutions. Design constraints will further limit the ability of the system under design from reaching its desired level of achievement. System design usually faces limitations. Therefore, design constraints must be identified, documented, and managed so they do not serve to manage design by default. The constraints will determine the output of the system under design whether or not they are acknowledged.

During the Synthesis process, the limitations of engineering must themselves be considered. Often solutions are limited by "the laws of physics" or state of the art. The design group undertaking the Synthesis process will need to clearly understand technical as well as programmatic limitations in order to trade risk, schedule, and financial constraints in overcoming the various challenges to satisfying the requirements baseline.

#### 4.5.2.1.3 Legacy System Definitions

In the FAA, it is rare when a solution is introduced into a pristine environment (one without previous attempts to satisfy user needs). It is also rare that established needs do not evolve and change as the operational environment also evolves and changes. Consequently, it is important to understand the existing legacy system that currently seeks to satisfy documented needs.

Understanding must include knowledge of the legacy system functions, performance, and its shortfalls. Only then can the design solution provide an alternative that improves upon existing capabilities and compliance with evolving user needs. All documentation regarding system functional, performance, and constraint requirements is therefore a necessary input into the Synthesis process.

The design constraints imposed by the need to operate with existing interfacing systems must also be understood. Interface Control Documents will provide the information needed to ensure integration into the existing environment.

Finally, the new system must eventually operate in the existing support environment. Documentation regarding legacy system maintenance and support is therefore needed to ensure that the system is designed in a manner that will enable it to continue to perform the needed user tasks, at the needed level of performance once introduced into the support system.

#### 4.5.2.1.4 Integrated Program Plan

The Integrated Program Plan (IPP) is the document within the Acquisition Management System that provides the plans for the detailed actions and activities necessary to execute the program within the cost, schedule, and performance baselines. The IPP encompasses all elements of program implementation. This may include the acquisition of systems and equipment, construction or modification of facilities and the physical infrastructure, functional integration of planned capabilities within the existing infrastructure, and procurement of services.

To perform Synthesis, one must also know the schedule or budget constraints, provided as clearly documented input. If the program phase is such that an IPP exists, it will provide this needed information. If such a plan does not exist the design team will have to determine the cost and schedule constraints through interface with program management and other stakeholders.

#### 4.5.2.1.5 Operational Services and Environment Description (OSED)

The OSED provides operational, safety, performance, and interoperability requirements. (See Functional Analysis (Section 4.4).) This document will provide needed information for the Synthesis process. The OSED identifies the desired air traffic services and or capabilities and their operational environments, including documented operational functions, performance expectations, and selected technologies. It will define the customer needs so that more appropriate alternative selections are feasible during Synthesis.

#### 4.5.2.1.6 Preliminary Work Breakdown Structure (WBS)

A preliminary WBS is provided and initially guides Synthesis efforts. (See Integrated Technical Planning (Section 4.2).) It is then refined under Synthesis by incorporating the characteristics necessary to support the functional and selected Physical Architecture(s) of potential design alternatives. The WBS will define categories of work, work packages, and, ultimately, through Synthesis the identification of associated physical elements. The WBS is invaluable from the planning and management perspective as it establishes a top-down framework for allocating and computing costs. The WBS assists in tracking the status of engineering efforts, resource allocations, cost estimates, expenditures, and cost and technical performance.

During Synthesis, the WBS must be scrupulously maintained and finalized to show in a hierarchical manner all work elements needed to complete a given program or project. As solution Physical Architectures are defined, the physical elements are introduced into the WBS.

### 4.5.2.2 Other Inputs

Beyond the inputs available from SE processes occurring prior to Synthesis, there will be inputs gathered during Synthesis from sources both internal and external to the SE process.

#### 4.5.2.2.1 Market Research

Market research is conducted during Synthesis to gather data necessary to conduct the process and for various reasons. During various stages of the AMS cycle the role of market research in the Synthesis process will vary.

The first time through the Synthesis loop, when an initial requirements database has been established (iRD) and provided as input to the Synthesis process, market research will help to determine the available technologies or various systems that can meet all or part of the requirements baseline.

If multiple viable alternatives do not exist, the baseline and Functional Architecture will be modified for optimization of alternative solutions. This optimization can occur numerous times as needed. During the final Synthesis loop, iteration the requirements baseline is finalized (FRD) and market research is conducted in concert with the design team to identify vendors who will meet all requirements of the finalized baseline.

One final and important consideration for market research is to determine the market base for proposed design alternatives. A smaller potential market base for a system and/or its components will inevitably translate to an increase in cost risk and a greater potential for the market not to continue to produce the needed items for the needed timeframes as the demand

for the supply diminishes. Market research is therefore valuable in determining not only what is available in the market place but also in determining the extent of its availability and the likelihood that it will continue to be available for the required project/program lifecycle for which Synthesis will provide a solution.

#### 4.5.2.2.2 Risk Mitigation Plans

Risk Mitigation Plans, although invaluable, may or may not be available for a given iteration through the Synthesis loop. The initial time through the Synthesis loop a requirements baseline and Functional Architecture, refined to reflect what is achievable through viable design solutions, are not available. Therefore, the risks associated with potential design alternatives are undefined, and concerns and issues associated with those risks are not yet forwarded to risk management by the Synthesis team.

Subsequent iterations through the Synthesis loops will have incorporated those initial concerns and issues; however, and a risk mitigation plan will have been developed under risk management (in concert with the Synthesis process).

#### 4.5.2.2.3 Trade Study Reports

Trade Study reports are invaluable, whether available to the Synthesis process from previous related efforts or whether solicited through the course of the process. The Trade Study report will provide documented answers to many issues and concerns for the Synthesis process, such as feasibility of design alternative, state of technology to support the alternative, and so on.

Existing Trade Study reports may identify related technologies that Synthesis may consider for incorporation into design alternatives. The existing report provides valuable insight into what is feasible given the current state of the art.

When the Trade Study is conducted in concert with Synthesis, it is geared toward exploring and determining feasibility, associated risks, maturity of design, conformance to the requirements baseline and Functional Architecture, and adherence to the various constraints to the program/project. This input is solicited in the sense that the Synthesis process works in concert with the trade study process to determine objectives and needed outcomes for the Trade Study report. (See Trade Studies (Section 4.6).)

### 4.5.3 Process Steps

The activities of Synthesis involve selecting a preferred solution or arrangement from a set of alternatives and understanding associated cost, schedule, performance, and risk implications. Synthesis entails undertaking a number of necessary and distinct steps geared toward achieving measurable goals and objectives while striving to manage or overcome constraints. Alternative candidate designs are first conceptualized; candidate alternative solutions are then defined and refined in order to meet the established requirements baseline.

Engineering analysis is used, as necessary, to evaluate alternatives. Evaluation will identify, assess, and quantify risks and select proper risk mitigation approaches. The risk management plan is utilized to refine the various design alternatives and achieve a balance between risk and technical progress. Too much risk within a given alternative could result in an unachievable design at the end. Assuming too little risk within a given alternative could also result in a solution that cannot be reached within the schedule constraints established for the project/program. These two extremes are balanced against the requirements baseline and established Functional Architecture through the guidance provided in the Risk Mitigation Plan(s). (See Integrated Technical Planning (Section 4.2).)

The analysis of alternative solutions will also result in an understanding of cost, schedule, and performance impacts. As subsystem requirements are defined, the identification of the needs, requirements, and constraints for lifecycle processes is completed. The specific tasks that define Synthesis are identified in Figure 4.5-5.

Synthesis demands creativity in order to achieve success. The ability to discover new solutions, to look at the requirements from new perspectives, and to formulate new concepts from two or more previously held ideas challenges the design group during this process. In order for the design group to succeed, each individual member of the team must exercise awareness and sensitivity to problems associated with each proposed approach. They must exercise flexibility, originality, self-discipline, and persistence while maintaining adaptability, nonconformity, tolerance for ambiguity, self-confidence, and a healthy skepticism.

In addition to exercising necessary individual characteristics, the team must also be aware of necessary group characteristics and dynamics essential for the successful development of achievable yet satisfactory design alternatives.

A group of "*like-thinkers*" typically arrives at a mutually agreeable solution, or solution set, in less time and with less discourse than a diverse group with differing perspectives and priorities. The solution reached in this relatively pain-free manner will not have always considered and analyzed every facet of the approach and all problems associated with it. As a result, the solution may not in the end satisfy all the requirements and design constraints levied on the Synthesis process. The devil's advocate plays an important role in the group and is as equally important to achievement of the group's goals, as is the consummate politician.

Once a diverse and well-balanced group is achieved, the group can begin to develop design alternatives through a variety of methods. The group can use such methods as brainstorming, brainwriting, and dynamic confrontation. Whatever method or combination of methods is selected for this creative development of alternatives, care is to be exerted to ensure that no one individual is allowed to dominate the group and, therefore, its outcomes. Likewise, care must also be exerted to ensure that every member of the group is given ample opportunity to contribute to the group's efforts.

***Brainstorming***

*This technique involves both idea generation and idea reduction. First idea generation occurs by simply identifying as many solution ideas as possible. Later in idea reduction those potential solutions are ranked into groups, with a specific group encompassing those potential solutions considered most useful to the group.*

*This technique is frequently considered a powerful one as it often results in the most creative and effective solutions. These solutions may arise from a combination of seemingly unrelated ideas, generated early in the process. Brainstorming encourages creative and original thinking.*

***Brainwriting***

*This technique builds on the concept of brainstorming, as it is the same technique but simply replaced verbal communication with writing. Utilizing this technique, team members will write down a number of relevant ideas on a sheet of paper (usually limited to three ideas). The paper is then passed to another team member who then develops those ideas. New ideas and elements are added to the original concept(s) and the augmented pages are then passed onto another team member.*

*This process continues until each team member receives back the sheet of paper containing the original concepts they created. At this point the beginning phase is complete and a group leader collects all idea/solution sheets.*

*The next phase is then commenced with all the sheets being handed out to the entire group. The group then works to revise the ideas developed in the prior phase.*

*This technique alleviates one of the problems associated with brainstorming in that dominant members are cannot easily steer the efforts of the entire group.*

***Dynamic confrontation***

*This technique is an adversarial group process. The main idea in this technique is for team members to criticize every idea. A presentation is first made and then every element and assumption of that idea is intensely challenged. This technique tests out every idea thoroughly and forces all members to thoroughly think through and develop their ideas.*

#### 4.5.3.1 Ensure Availability of Needed Input—Step 1

The availability of data will depend on the status of the Synthesis process. If it is the first-time entry into Synthesis, or the first Synthesis loop, not all data will be available. However, as the Synthesis process continues, more data will become available from other SE disciplines. Table 4.5-1 summarizes the data that is required for the Synthesis process and its availability.

**Table 4.5-1: Needed Synthesis Data**

| Input | Delivering Process | SEM Reference | Availability |
|---|---|---|---|
| Requirements Baseline | Requirements Management | Section 4.3 | 1st and subsequent loops |
| Functional Architecture | Functional Analysis | Section 4.4 | 1st and subsequent loops |
| Legacy System Specifications | External to SE | N/A | 1st and subsequent loops |
| Legacy Interface Requirements | Identify, Define and Control Interfaces | Section 4.7 | 1st and subsequent loops |
| Draft IPP | Integrated Technical Planning | Section 4.2 | 1st Synthesis loop |
| Operational Services and Environment Description | Functional Analysis | Section 4.4 | 1st and subsequent loops |
| Preliminary WBS | Integrated Technical Planning | Section 4.2 | 1st Synthesis loop |
| Market Research | External to SE | N/A | May not be available 1st loop through Synthesis |
| Trade Study Report | Trade Studies | Section 4.6 | May not be available 1st loop through Synthesis |
| Risk Mitigation Plans | Risk Management | Section 4.8 | May not be available 1st loop through Synthesis |

#### 4.5.3.2 Requirements Review and Objectives Definition—Step 2

Synthesis begins with a review of the requirements baseline and the Functional Architecture in order to define what is to be performed and at what level of performance to meet stakeholder needs. Requirements Management will not dictate *how* the stakeholder needs will be met. The design Synthesis process will determine how to achieve stakeholder needs.

Establishing objectives assists in optimizing adherence to the requirements set within the technological and programmatic limits imposed on the design process. Objectives must be linked to stakeholder needs and system requirements. Objectives will take into consideration operational criteria, mission success, technical performance, cost, schedule, quality, risk, failure rate, maintainability, and supportability. Through definition and prioritization of all design solution objectives, the optimal solution is achieved that best satisfies the requirements set under consideration.

Often, devices will perform their functions at varying performance levels in differing environments. For instance, the system delay for a computer system gathering surveillance data from various sources and formulating a graphical representation of all existing air vehicles in a given space and presenting it to the controller on a display is vastly different at various locations and at various times during the day. Stakeholders would only state minimum NAS requirements for presentation of data to them from the source. The engineers involved in Synthesis must decide how they will meet those stated requirements in the various environments. A tailored system for each location might be provided, thus lowering the overall cost of up-front procurement, as computer systems with less processing power may be utilized in small airport areas. However, the training and support regarding multiple systems must also

be addressed in terms of added cost for multiple versions of the system. In this example, the engineers must look at the operating environment of the solution to determine what their objectives are for performance, for up front procurement cost, and the lifecycle costs of supporting the resulting system. These items represent three distinct objectives to be satisfied in selecting a design that will fulfill the stakeholder requirements.

Another facet to consider is that a single system design may not necessarily satisfy all of the requirements associated with the functional area under consideration. Multiple systems may be required to satisfy the entire requirements set.

Ideally, alternative solutions should satisfy all requirements, but it is useful to include solutions that challenge the requirements and lead to a better system concept. Various options are to be considered eventually in light of the objectives for the resulting system(s). Such alternatives include relaxing requirements of marginal utility that are costly to implement or extending requirements when added capability can be purchased cheaply while resulting in operational benefits.

#### 4.5.3.2.1 Performance Objectives

The performance objectives, although highly dependent on potential system solutions, must be clear, as they serve to define the main purpose of the system. The engineering team must not only define all terms that will measure how the system will perform, but the actual desired performance levels must also be stated. The accuracy, capacity, response time, throughput, and other similar requirements are reviewed and analyzed against feasible design possibilities. The threshold performance levels are clearly documented for the design under consideration. Most, if not all, of the performance requirements are contained in the requirements baseline provided under Requirements Management. However, the stated performance objectives that are to be achieved by any potential system or systems are clearly documented at the outset of Synthesis so that the tradeoff between these and other objectives may follow.

#### 4.5.3.2.2 Reliability Objectives

The reliability objectives must be defined in terms of the likelihood, or probability, that the resulting system will operate at its objective performance level for a defined period of time under normal operating conditions. In order to clearly define the reliability objectives, engineers must translate the environmental and operational data as contained in the OSED. Allocation of the RMA requirements in the requirements baseline is conducted in concert with the requirements process and Specialty Engineering in order to allocate the various reliability maintainability objectives to the various design alternative functional areas.

#### 4.5.3.2.3 Compatibility Objectives

The engineering team must define the objectives for the system to work, or interface, with both existing systems and those under agency development. Interface objectives are stated in terms of not only the data and physical interface, but also in terms of the working environment imposed by the existing systems or system elements with which the potential design alternative must interact. The objectives must address both backward compatibility with legacy systems and forward compatibility with known evolving technologies, protocols, and standards.

#### 4.5.3.2.4 Extensibility Objectives

The engineering team must define the objectives to allow for alternative design approach(es) to be able to adjust to a changing environment. For example the ability to process more flight data to adapt to a growth in air traffic must be clearly defined and documented. This is particularly important when it is known that the existing environment will evolve. As the environment evolves, so must any design alternative evolve to adapt to the new environment. Projections for

changes are documented along with the stated objectives for extensibility of the design alternative.

#### 4.5.3.2.5 Flexibility Objectives

Flexibility differs from extensibility, which means the ability to adapt to and accommodate growth needs. Flexibility is the ability of the design alternative to serve new or multiple uses. An example of flexibility is a multipurpose display that provides graphical display of flight plan data, surveillance data, or both simultaneously without need for modification.

#### 4.5.3.2.6 Cost Objectives

A limited budget is a never-ending facet of the Synthesis process. Thus, it is essential to clearly define the cost objectives for any potential design alternative at the outset. Care should be taken not to overemphasize cost of the item over all other objectives. The old adage, “You get what you pay for” is all too often true. Consequently, cost objectives are best stated as a range within which the design alternatives must reside. Cost objectives must include all facets of the potential design alternatives’ lifecycle. Restricting objectives merely to the initial cost of a design solution may not fairly consider other design alternatives that have higher initial cost, r but whose overall lifecycle costs are lower due to quality, reliability, and supportability characteristics. Therefore, the cost objectives shall be defined for all stages of the intended lifecycle.

#### 4.5.3.2.7 Schedule Objectives

What a design alternative will do, how well it will perform the function(s), and where it perform become irrelevant if the design alternative is not delivered to the user when needed. A design alternative delivered before its time is as potentially damaging to the effort as one delivered too late. Therefore, the schedule objectives for all facets of the design alternatives’ lifecycle must be defined clearly and comprehensively. The schedule objectives for test, operational introduction, full operational capability, service life, and so on are all documented.

#### 4.5.3.2.8 Objectives Tradeoffs and Hierarchy Definition

Rarely, if ever, are projects faced with unlimited time and financial resources. Tradeoffs and compromises are common during Synthesis in order to achieve the design objectives with an acceptable level of requirements compliance. It is essential to define the design objectives and rank their relative importance.

The prioritized set of objectives is to be well established and documented before design solutions are considered.

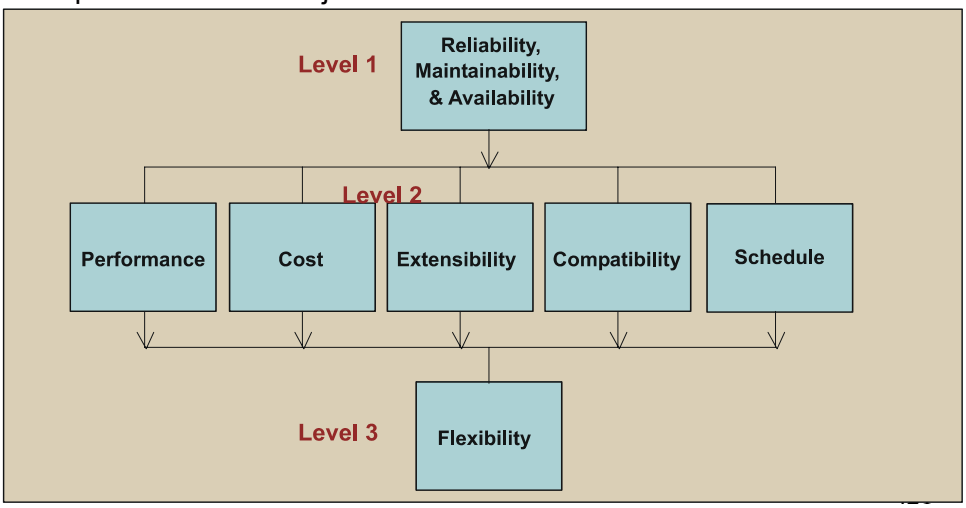



**Figure 4.5-3: Example Three-Level Objectives Hierarchy**

Objectives in both the above categories and additional categories to be considered under the program/project are first documented as a list. The list is expanded to include more categories as determined necessary in concert with program

management, Specialty Engineering, and stakeholders. The importance of each objective relative to the others is then determined for all objectives. Once all the relative priorities are established, priority levels are defined based upon the findings. This task, although not simple, is necessary, as the results are invaluable later when design alternative tradeoff analysis is performed.

Assume that each of the categories in Section 4.5.3.2 has one objective; there are then a total of seven resulting objectives. For the purposes of this example, we are examining a project that eliminates a reliability deficiency in an existing fielded system. In this particular example, RMA is therefore considered of higher importance than all other alternatives. Also, as the product introduced is only an interim solution to fulfill a shortfall, system flexibility is considered less important than all other factors. If all remaining objectives are considered to be of equal importance, we are then left with three priority levels (Figure 4.5-3)

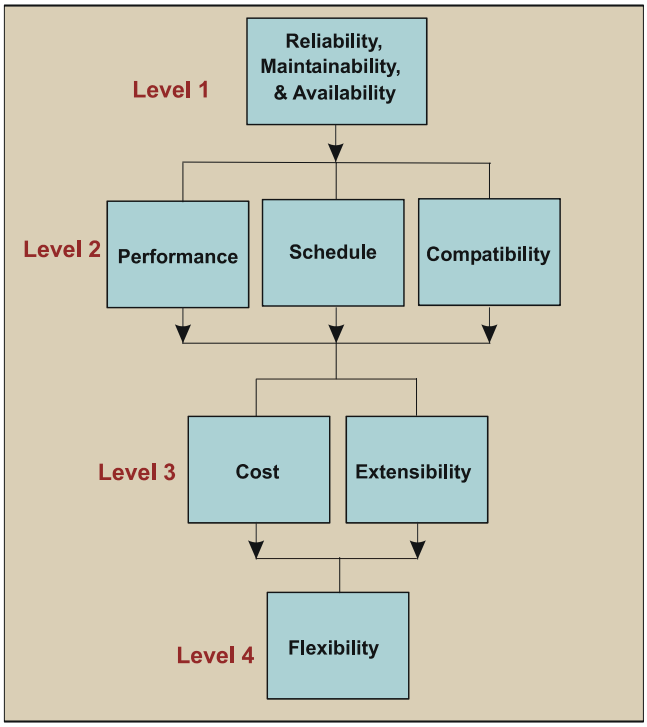



**Figure 4.5-4. Example Four-Level Hierarchy**

Establishing the objectives hierarchy is seldom this simple, however. The items in level two of the figure are rarely seen as equal in importance. This level may be further broken down into groups, with each group containing objectives of equal importance and with one group being considered to be more important than the other. This leaves four levels of priorities instead of three, and the hierarchy is established, complete with relative objective priorities, and priority-level definition (Figure 4.5-4).

### 4.5.3.3 Define Design Solution Set—Step 3

During this Synthesis step, grouping of needed functions into common functional areas is complete and the Functional Architecture is established. The design team must now begin partitioning desired functional elements into design elements. During review of various designs in terms of whether or not they will

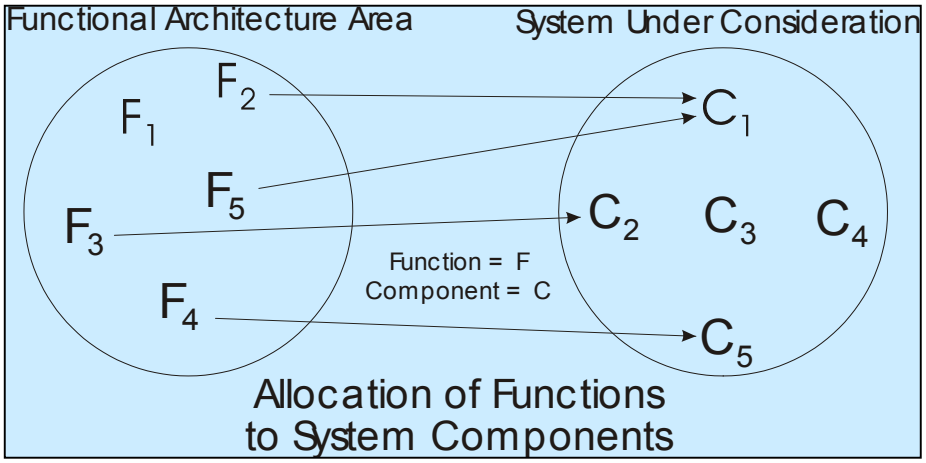



**Figure 4.5-5. Functional Partitioning to System Components**

perform the desire functions, each function in the Functional Architecture is mapped to a component of the system under review. Some components will perform only one function, whereas others may perform one or more functions (Figure 4.5-5).

This Synthesis process step boils down to generating alternative design solutions for the functional elements identified during Functional Analysis (Section 4.4) that perform the needed functions and adhere to the requirements for that functional area. The alternative solutions should be composed of one or a combination of more than one of the following: hardware, software, material, data, facility, people, and techniques.

There are a variety of tasks conducted to identify various design alternatives. The tasks may be performed sequentially or concurrently by various subteams. If the Synthesis group is small, the preferred choice is for all members to look at identifying alternatives sequentially. If the group is large enough and good communications exist among all members, the option to concurrently identify solutions by the various means described below is worth exploring. Both approaches require that the entire group conduct prior planning. Concurrent exploration of alternatives requires close coordination throughout identification of alternatives until all possibilities are identified; whereupon, the subteams will once again combine to complete this Synthesis step. Figure 4.5-6 illustrates the tasks feeding the Synthesis step that identifies the various design alternatives.

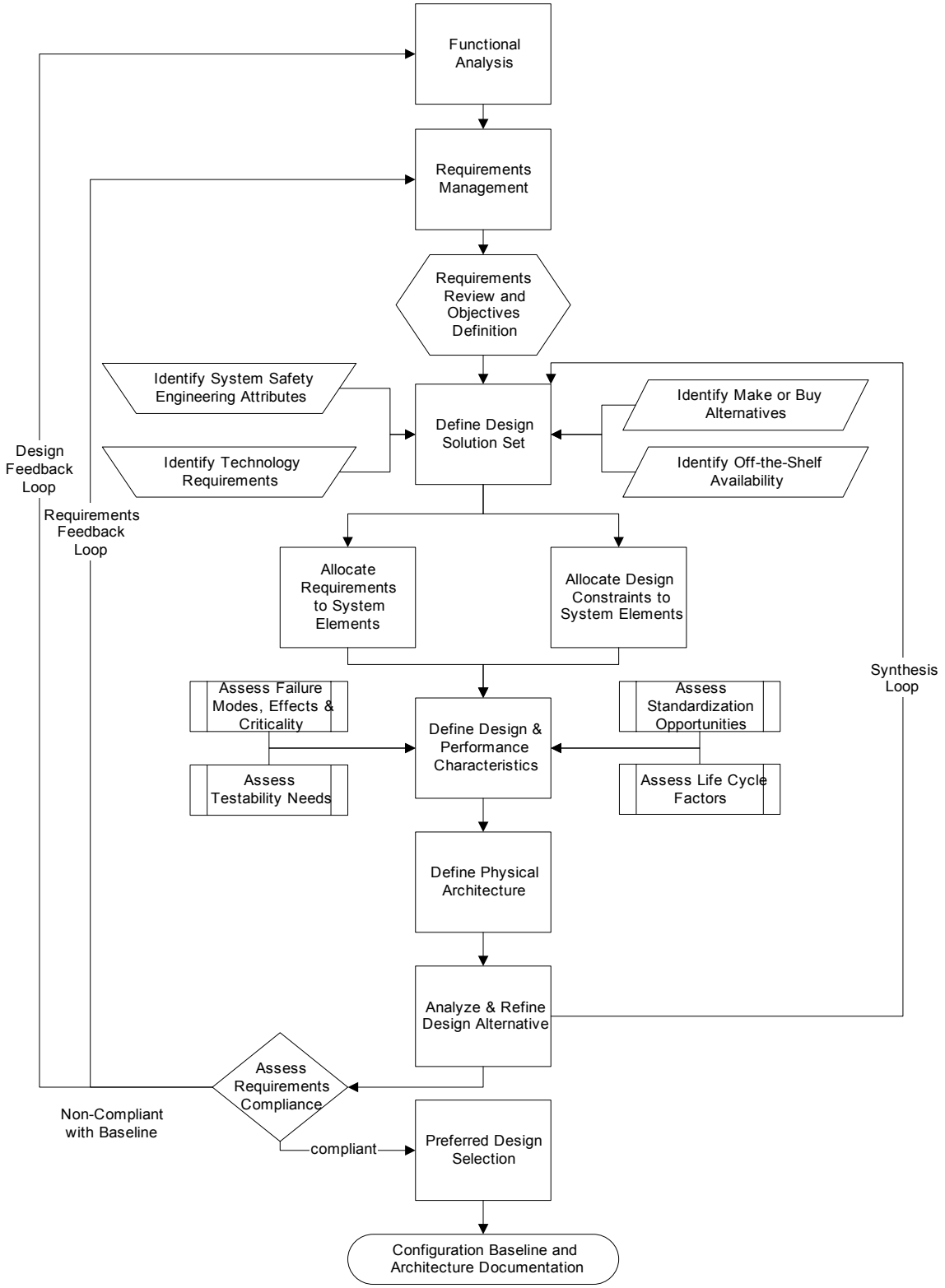



**Figure 4.5-6. The Synthesis Process Activities**

#### 4.5.3.3.1 Technology Assessment

This assessment addresses not only potential incorporation of existing technology into design solutions, but also looks at the risks and limitations imposed by and upon that technology. Each alternative under consideration is analyzed against the evolving technologies available in the marketplace. Available technologies are studied in terms of suitability for the design under consideration, potential enhancement to design performance, improvement to maintainability of the resulting system, cost-effectiveness, and maturity.

The need for a new technology that makes possible a performance or functional enhancement not possible with previous technologies must be carefully weighed against the risk imposed by that technology. The potential benefits of inserting the technology must outweigh the potential risks to cost, schedule, and performance imposed by that technology.

To continue consideration of the potential technology, insertion the impacts to the end user must be considered through a human factors analysis. The tasks, roles, and jobs assigned to humans are analyzed and assessed to determine whether or not the end users of the resulting system have the required knowledge, skills, and abilities. If the needed knowledge, skills, and abilities do not exist, then the cost and schedule risks of achieving them with the new technology are weighed against the benefits derived from the technology. Training and personnel pipelines are fully evaluated to ensure that they meet requirements.

#### 4.5.3.3.2 Identify Specialty Engineering Attributes

The design team must work in concert with specialty engineers to identify the characteristics of each potential alternative necessary to fulfill interdisciplinary needs.

The design team works in concert with safety engineers to analyze each alternative to identify potential hazards to the hardware/software components of the system, the humans involved in the system as users and support personnel, or the environment. The analysis must demonstrate that the design under consideration results in safe system operations. All aspects of the design, development, manufacture, test, operation and support of the potential design are included in the analysis.

The design team works in concert with human engineering to analyze each alternative for human factors suitability. Each alternative is analyzed with respect to the human user system interface. (See Specialty Engineering (Section 4.8).)

##### 4.5.3.3.2.1 System Safety Engineering

System hazards are identified and assessed for the design alternative. The hardware, software, operational, and ambient environments, procedures, and human elements of the design alternative are analyzed, and historical or test data is applied to estimate the risk (severity and probability) of each and all identified hazards. Controls are then designed in accordance with the safety order of precedence described in Section 4.8. All hazards and their associated controls are prioritized according to their risk criticality rating. The analysis results are used to direct further design efforts to characterize controls, safety features, redundancy, and graceful system degradation elements of the system.

#### 4.5.3.3.3 Off-the-Shelf Opportunities

Each design alternative is analyzed to determine if an off-the-shelf item exists that will fulfill the allocated requirements. Off-the-shelf solutions can include nondevelopmental hardware or software.

Once off-the-shelf solutions are identified, each must undergo assessment to ensure that a variety of factors are considered in determining suitability. The number of systems available off-the-shelf must be gauged against the number users need. The quantity needed must consider not only those needed initially by the user community, but also those needed to serve as replacements over the anticipated service life of the system.

Another facet of the suitability assessment process is consideration of the environment in which the prospective off-the-shelf item must eventually operate. The ability of the proposed item to adapt to the existing support structure is a necessary component in determining its suitability. If the item requires new equipment and/or training for support during its lifecycle, the benefits of the item must outweigh its cost and schedule impacts.

Finally, the manufacturer(s) of the off-the-shelf item must undergo assessment. Attributes such as product maturity, upward/downward compatibility, manufacturer track record, financial stability, and quality practices must be factored into the commercial product selection process. If the products or manufacturers fall short in any of the reviewed categories, they must be considered a risk. Refer to Appendix F of the FAA COTS Risk Mitigation Guide for a more detailed listing of COTS nontechnical selection factors.

#### 4.5.3.3.4 Make-or-Buy Alternatives

A cost analysis is performed for the design alternative(s) and used to support a make-or-buy decision. This analysis needs to address whether it is more cost-effective to produce the design element versus using an established supplier.

When cost, schedule, and risk are considered ,it is most beneficial to design and develop (a “make” decision) a peculiar system that satisfies all requirements of the functional area, and the team will proceed with this approach as a viable design alternative.

### 4.5.3.4 Identify Alternatives for Design Solution

Input from preceding processes and previous Synthesis steps identified not only potential alternatives, but also design constraints for potential solutions. This input is used to help the engineers determine if existing or newly developed items can accomplish the functional element under consideration.

Synthesis strives to identify viable design alternatives, refine those alternatives to fulfill the requirements baseline, and finally select the most balanced and beneficial design to introduce into the field. To accomplish this goal, all possible alternatives are first identified. These are reduced to reflect only those alternatives considered viable or worth pursuing.

#### 4.5.3.4.1 Populate the Solution Set

The design team identifies all possible design solutions that may serve to satisfy all or part of the requirements baseline. At first, it is merely an exercise in exploring all possibilities. Once the team has exhausted all possibilities, the design solution set is evaluated both as a group and individually. If the team has been able to identify only one possible design alternative, then the job is not complete. This is because no matter how large or difficult the requirements baseline and the functional area with which it is associated, there will always exist at least one possible design alternative: *do nothing*. The default is to continue the status quo and not

present new and/or innovative design solutions. Given the fact that a great effort went into previous SE processes (such as Requirements Management (Section 4.3) and Functional Analysis (Section 4.4)), it is unlikely that entrance into Synthesis would have occurred if all requirements in the functional area, with its associated requirements baseline, were satisfactorily met. Clearly, it is possible to identify an insufficient number of alternatives. The task is to develop additional alternatives that present better options.

The following methods can be used to develop new alternatives.

- **Change the characteristics of existing alternatives.** First, list all existing alternatives. For each alternative in the list, itemize its main characteristics. Generate a table with the rows representing the list of alternatives and the columns representing the main characteristics of all alternatives. In all likelihood, each of the potential alternatives will possess characteristics that are both similar and distinct from those of the other alternatives. The positive characteristics are identified. Missing characteristics needed by a design alternative and not represented by any potential solution should then be listed. Finally, more alternatives are then added to the list, as the characteristics within the previously listed alternatives are varied, enhancing the new alternatives with needed positive characteristics and eliminating as many negative characteristics as possible.

- **Go back to the objectives.** Focus on the most important objectives one at a time. List alternatives that will meet each of those top-level objectives. Then work down the objectives hierarchy developing more alternatives or refining existing alternatives that satisfy those additional objectives as you address each one.

- **Finally, step back and look at all the objectives and requirements set.** List alternatives that will maximize the number of objectives and requirements that can be met with the alternative.

If there still seems to be lack of viable alternatives, step through the various methods, introducing more creativity and ingenuity each time through. Eventually, a solution set will reach a stable point, and the group will be done identifying design alternatives.

Now that a significant number of design alternatives are identified, the group of alternatives is evaluated. First the design team must determine that a number of sound viable design alternatives exist that can satisfy all or most of the baselined requirements. It is possible to continue the Synthesis process with too many design alternatives, as the remaining steps will detail and document each alternative to a great degree. Therefore, proceeding with too many alternatives can waste valuable time and resources. One can argue that proceeding with one alternative is not sufficient. Likewise, one might also argue that proceeding with 10 alternatives that must be thoroughly defined and documented is an unnecessary excess; so reduction of the alternatives set to a manageable size or number of alternatives must occur.

#### 4.5.3.4.2 Reduce Solution Set to Manageable Number of Alternatives

When viable design solutions are identified, compromise of requirements considered absolutely necessary to satisfy the operational needs must not occur. Those requirements considered so important to satisfying the user needs that a system not meeting them is deemed unnecessary or unacceptable are to be considered "*threshold requirements.*" A potential design solution must satisfy threshold requirements for further consideration as a design alternative. Threshold requirement compromise, or tradeoff is not an option for consideration. A design alternative not meeting a threshold requirement that cannot be modified easily to meet the requirement(s) is eliminated and not considered further in Synthesis.

The objectives hierarchy is used next. If the remaining alternatives set contains potential solutions that do not meet the top-level objectives, and they cannot be easily or affordably modified to do so, then those alternatives are eliminated from the set of potential alternatives. As with requirements, some objectives are not subject to compromise, and alternatives not meeting the high-priority objectives, as defined earlier, should no longer be considered during Synthesis.

If potential solutions are only able to satisfy a portion of the functional area requirements or objectives, the design team considers various options in order to develop a set of viable design solutions. One or more of the solutions that nearly satisfy the team's objectives and/or requirements could be modified to achieve satisfactory results. The following options may be used to modify either the problem (functional area under consideration with its associated requirements) or the alternative design solutions:

- **Trade Study Request.** To determine if one or more of the options can be modified to fulfill the desired requirements and/or objectives, a detailed analysis, such as that conducted under Trade Studies (Section 4.6) is requested. Under the Trade Studies process, incorporation of new technologies and a variety of other means are investigated. Trade Studies If the results of the study render viable design alternatives, then Synthesis proceeds into the next step, requirements allocation. However, if it turns out that no alternative can meet all of the requirements in the functional area under consideration, the requirements and/or the functional area themselves are analyzed.

- **Initiate Requirements Feedback.** When the requirements baseline for the functional area under consideration cannot be satisfied through viable design alternatives, feedback to Requirements Management (Section 4.3) is initiated during Synthesis. If requirements are only partially met by all potential designs, the ability to meet the requirements set is analyzed concurrently by Synthesis and Requirements Management. Consideration is given to modifying requirements to lower and achievable levels. Full compliance is deferred until technological or other advances allow for full compliance with the original requirements. Requirements that cannot achieve even partial compliance in the various designs are addressed through the design loop.

- **Initiate Design Feedback**. Due to discovery of design issues, the Functional Analysis (Section 4.4) is reexamined, and the initial decomposition or performance allocations are reassessed. Design issues include identifying a promising physical solution or open-system opportunities that have different functional characteristics than those foreseen by the initial Functional Architecture requirements. Issues also include the inability of all design alternatives to fulfill the same functional area requirements, which may be addressed by repartitioning of the functional area. The functional area is subdivided so that allocation of those requirements to be satisfied by the alternative designs can be made down to perspective system elements. The remaining functional areas whose associated requirements will not be satisfied remain with the Functional Analysis (Section 4.3) process. The associated requirements are documented as unsatisfied in the Requirements Management (Section 4.3) process. The functional area(s) with the associated unsatisfied requirements are partitioned out of Synthesis, back to Functional Analysis for future Synthesis loop identification of potential solution(s).

Review and analysis of all remaining alternative solutions are conducted in concert with Specialty Engineering, risk management, lifecycle engineering, and integrated program planning

in order to determine adequacy and suitability of each remaining alternative. The alternatives are pared down to preferred system concept(s).

#### 4.5.3.5 Develop System Concepts

The previous Synthesis steps have resulted in a promising set of conceptual designs for systems satisfying the requirements baseline for the functional area under consideration. Each design concept must now be developed in more detail so that requirements and design constraints are assigned to the top-level elements of that system design.

##### 4.5.3.5.1 Allocation of Requirements to System Elements—Step 4

In prior steps the functional area and associated requirements were adjusted in concert with Functional Analysis and Requirements Management, respectively. As this Synthesis step is entered, the requirements to be satisfied by the design solution(s) are established, and this step furthers the design process by allocating the requirements to system elements.

These elements are the highest-level distinct elements of the system in the areas of hardware, software, and humans in the system. Each system element must perform at least one function within the functional area to be considered separately and distinctly in the traceability of requirements.

The design engineers proceed in allocating requirements to the selected system elements. All requirements that the system must satisfy are documented, and formal tracking of those requirements through the various design and acquisition phases of the system begins. Documentation includes information regarding the hardware, software or other components of the system to which each requirement is allocated.

##### 4.5.3.5.2 Allocation of Design Constraints to System Elements—Step 5

Design constraints that apply directly to system elements are identified. These constraints do not apply to the functions performed, but rather the elements: hardware, software, or people. Design constraints differ from constraint requirements in that they recognize existing limitations to design of a system, its interfacing systems, and its operational and physical environment. Such design constraints will include power, weight, data throughput rates, memory, and other resources. These constraints represent the inability to achieve a capability or level of performance due to such issues as technology, and space available for the system.

Design constraints are especially important in analyzing the design of potential replacements for existing systems. This is of particular interest to design engineers when major elements of the original system may be retained. The design constraints once allocated will clearly define which system elements remain, are added, or modified.

Those technology constraints identified during the prior technology assessment will be allocated to the system elements. Those constraints identified during review of Specialty Engineering attributes are allocated to ensure that inappropriate design characteristics are not introduced into the selected system. Finally, environmental constraints are allocated down to the system element level. Environmental constraints can be introduced by climatic conditions in which the total system will operate, by the facilities in which the system will be housed, or more globally by environmental hazards and constraints (such as Environmental Protection Agency regulations) imposed in the region(s) where the systems will be used.

#### 4.5.3.6 Define Design and Performance Characteristics—Step 6

With the system concepts now defined, the team must identify the design and performance characteristics of each alternative. Once defined, the design and desired performance characteristics are documented. The system(s) characterization is all-inclusive and addresses all facets of the system under design, including the associated human-engineering elements and lifecycle considerations or needs.

During this phase, there is strong benefit to practice concurrent engineering. The entire functionality of the system(s) under design is considered. When the design and performance characteristics are defined, the entire lifecycle of the potential system must be considered, from inception to disposal, in an integrated process. This requires involvement of all Specialty Engineering disciplines/processes (Section 4.8) in the Synthesis process. Thus, sound engineering decisions are made based on strong consideration of all phases and aspects of the system under design consideration.

##### 4.5.3.6.1 Assess failure modes, effects, and criticality

Failure modes, the effects, and the criticality of failure are assessed for the design alternative. The hardware, software, and human elements of the design alternative are analyzed and historical or test data is applied to estimate the probability of successful performance of each alternative. A failure modes and effects analysis (FMEA) should be used to identify the strengths and weaknesses of the design solution. (See Specialty Engineering (Section 4.8).) For critical failures, a criticality analysis is conducted to prioritize each alternative by its criticality rating. The analysis results are used to direct further design efforts to characterize redundancy and graceful system degradation elements of the system.

##### 4.5.3.6.2 Assess testability needs

The testability of the design is analyzed in relation to the operational or maintenance needs. The team determines the need for built-in test (BIT) and/or fault-isolation test (FIT) for each potential design alternative. For elements that are normally maintained by the users or field support engineers, test mechanisms are considered in the design and incorporated as necessary. Diagnostic operations to support lower-level maintenance actions are likewise incorporated into the design solution.

##### 4.5.3.6.3 Standardization Opportunities

The alternative is assessed for possible use of standardized end items that are technologically and economically feasible. Use of design elements that implement commercial and international standards is strongly considered.

##### 4.5.3.6.4 Lifecycle Factors Assessment

The design of each alternative is assessed to determine the degree to which quality factors (producibility, ease of distribution, usability, supportability, trainability, and disposability) have been included in the solution. Additionally, associated lifecycle process needs, requirements, and constraints are identified and defined for each design under consideration. (See Lifecycle Engineering (Section 4.13).)

#### 4.5.3.7 Physical Architecture Definition—Step 7

A Physical Architecture defines and describes the way in which the various Functional Architecture elements can be brought together to form physical entities. The physical entities must represent a viable design alternative and must provide one or more services that address user needs as translated by the requirements baseline. The Physical Architecture may involve

such physical entities as runways and various forms of equipment; such nonphysical entities as software; or a combination of the two.

The Physical Architecture identifies the physical subsystems, and architecture flows between subsystems that will implement the functions and provide the needed services/capabilities. The Physical Architecture further identifies the system inputs and outputs.

In constructing a Physical Architecture, the following definitions are used:

- **Physical Entities.** The classes of physical entities that will be used are:

  – Subsystems. Subsystems are the primary structural components of the Physical Architecture. They perform functions that "belong" together and whose interfaces require interoperability and compatibility.

  – Users. These are people who interact with the architecture implementation. They could either be those who use the system (such as the flying public or pilots in the NAS), or operators who use features of the system (such as air traffic controllers in the NAS). Each interface to a user involves human interaction with the system.

  – External Systems. These are organizations and agencies and/or their systems that will likely interact/interface with the system under design (such as DoD or National Weather Service to the NAS).

  – Environment. This is the physical world, such as pavement, air, obstacles, and so on.

- **Physical Interfaces.** These are mechanical, electrical, data, and other interfaces between system elements or subsystems. Physical interfaces also include all interfaces between the system and its outside world.

#### 4.5.3.7.1 Decomposition Into Physical Entities

The architecture can be viewed at several levels of detail. The architecture defines collections of subsystems while defining their interfaces. Consideration is given to a variety of engineering and programmatic disciplines along with stakeholder contributions, and all are incorporated into the Physical Architecture.

#### 4.5.3.7.2 Physical Interfaces Definition

The enterprise identifies and defines the physical interfaces among products, subsystems, humans, lifecycle processes, and external interfaces to higher-level systems or interacting systems. Physical interfaces that impact design include communication, data, support, test, control, display, connectivity, or resource replenishment characteristics of the interaction among subsystems, the products, humans, or other interfacing systems or a higher-level system. (See Interface Management (Section 4.7).)

### 4.5.3.8 Design Alternative Analysis and Refinement—Step 8

As a particular design alternative is refined, it is analyzed to determine how it satisfies the allocated functional and performance requirements, interface requirements, and design constraints and how it adds to the overall effectiveness of the system or higher-level system.

During analysis, specialty engineers work with design engineers to ensure that requirements such as reliability, availability, maintainability, supportability, safety, human factors, security, electromagnetic compatibility, and spectrum management are incorporated into the design. Additionally, lifecycle process requirements are identified and defined for each alternative system product solution and aggregate of solutions.

#### 4.5.3.8.1 Assess design capacity to evolve

The design alternative is analyzed with respect to its capacity to evolve or be reengineered, accommodate new technologies, enhance performance, increase functionality, or incorporate other cost-effective or competitive improvements once the system is in production or in the field. Limitations that may preclude the ability of a system to evolve should be identified and the approach analyzed and refined to resolve any limitations. The supportability of an evolving system may require the support process to evolve along with the product. This consideration may significantly affect support funding and training requirements.

#### 4.5.3.8.2 Develop models and prototypes

Models and/or prototypes are developed to assist in:

- Identifying and reducing risks associated with integrating available and emerging technologies

- Verifying that the design solution (made up of hardware, software, material, humans, facilities, techniques, data, and/or service) meets allocated functional and performance requirements, interface requirements, workload limitations, and constraints

- Verifying that the design solution satisfies Functional Architecture and requirements baseline requirements

The models, data files, and supporting documentation should be maintained, and each version of a model or data file that impacts requirements, designs, or decisions should be saved in the integrated database. Models may be digital, partial, or complete and may be hardware, software, or a combination of both, or may include human models or human-in-the-loop simulations or mockups for usability testing and workload measurement. (See Trade Studies (Section 4.6).)

### 4.5.3.9 Check Requirements Compliance—Step 9

Compliance with the requirements baseline for the functional area is reviewed and analyzed. For each alternative, solution level of compliance to all requirements is documented. If full compliance is not reached by any of the alternatives, and they all fail to meet the same requirements, the design loop is initiated. If some, but not all, of the alternatives fail to fully meet all of the requirements, and compliance varies among approaches, the requirements feedback loop is initiated for each design. This is not to be confused with Verification (Section 4.12).

#### 4.5.3.9.1 Design Loop

The design loop involves revisiting the Functional Architecture to verify that the Physical Architecture developed is consistent with the functional and performance requirements. It is a mapping between the functional and Physical Architectures. During design Synthesis, reevaluation of the Functional Analysis may be caused by discovery of design issues that require reexamination of the initial decomposition, performance allocation, or even the higher-level requirements. These issues might include identification of a promising physical solution or

open-system opportunities that have different functional characteristics than those foreseen by the initial Functional Architecture requirements.

#### 4.5.3.9.2 Requirements Feedback Loop

The system design is audited to determine compliance with the requirements set. Audits are performed at various levels, from the top-level Physical Architecture down through each hierarchy level to the lowest-level system element or configuration item. Compliance with the requirements is assessed through both informal and formal reviews. The audit results are then fed back to earlier Synthesis steps as needed. The audit results may call for requirements changes at varying levels, or they may lead to design changes to ensure compliance.

### 4.5.3.10 Select Preferred Design Solution

The alternative solution is selected using all prior analysis conducted in Synthesis or in conjunction with Requirements Management (Section 4.3), Functional Analysis (Section 4.4), Trade Studies (Section 4.6), Specialty Engineering (Section 4.8), and Risk Management (Section 4.10). The selected solution shall be the one that offers the most balanced design. Upon being selected, the design is detailed and finalized. The designation and description of interfaces (internal and external) among design elements are finalized. The design is baselined and placed under formal configuration management processes.

## 4.5.4 Process Outputs

### 4.5.4.1 Requirements Compliance Matrix

All requirements have been mapped to the system elements. As the mapping occurred during Synthesis, a matrix was developed containing all requirements, the subsystem or element to which they were assigned, and the level of adherence to the requirements achieved by the component. The matrix is designed for each level of the Physical Architecture, and all performance, functional, and constraint requirements are listed in the matrix to reflect each level of the architecture. Compliance levels are determined using system/cost-effectiveness analysis, simulations, demonstrations, inspection, and or testing.

### 4.5.4.2 Alternative Description(s)

#### 4.5.4.2.1 Concept Description Sheets

A separate description for each of the alternatives developed and refined during Synthesis is documented. For the selected or preferred design, more detail is provided such that other SE processes can best utilize the information. A complete description of the system, the system operational use, and characteristics is documented in the description sheets.

#### 4.5.4.2.2 Architecture Block Diagrams

The hierarchical relationship of all system elements is to be documented in an Architecture Block Diagram (ABD). The ABD includes hardware and software elements and their hierarchy, documentation and data, facilities, test equipment, and support.

An external ABD is also to be developed to depict the external elements that affect the selected system. Like the system ABD, the external ABD should include all hardware, software, facilities, personnel, data, and services having a significant effect on the selected system.

#### 4.5.4.2.3 Schematic Block Diagrams

Schematic Block Diagrams (SBD) illustrate the physical partitioning and interfaces for each candidate hardware and software design solution determined to be viable. SBDs should not be developed for every conceivable design, but only for those that are worthy of detailed evaluation.

#### 4.5.4.2.4 Interface Drawings

Drawings are developed for all system physical element interactions. Additionally, all interactions to external physical elements are also documented in drawings. The drawings provide the visualization of interfaces and are the basis by which interface specifications and control documents are developed later under Interface Management (Section 4.7).

### 4.5.4.3 Physical Architecture

For the alternative solution, the system elements are identified along with their arrangement and the interactions between them. A description of the salient features of the overall solution is developed as well as descriptions for the system elements and their relationships establishing a potential System Architecture baseline. The descriptions are diagrams, schematics, concept drawings, tabular data, and narrative.

The design architecture is established at a level appropriate to document the design solution and interfaces. It includes the requirements traceability and allocation matrices, which capture the allocation of functional and performance requirements among the system elements. Design architecture definitions should be stored in the integrated database along with tradeoff analysis results, design rationale, and key decisions to provide traceability of requirements up and down the architecture. Verification of the design architecture should be accomplished to demonstrate that the architecture satisfies both the validated requirements baseline and the verified Functional Architecture.

### 4.5.4.4 Integrated data package

The drawings, schematics, software documentation, manual procedures, etc., are developed as necessary to document the selected design elements in an integrated data package.

#### 4.5.4.4.1 Configuration Item Documentation

Each of the system elements are identified during the Synthesis process. This includes all hardware configuration items (HWCI) and computer software configuration items (CSCI). Documentation of each HWCI and CSCI occurs at identification of the item in summary or preliminary fashion. Once a final design alternative is selected detailed documentation for each HWCI and CSCI of the selected system is developed thus establishing a configuration baseline for the system. (See Configuration Management (Section 4.11).)

#### 4.5.4.4.2 Specification Inputs

During Synthesis, compliance with the requirements baseline was assessed. This analysis sometimes results in recommendations for requirements modification or elimination. Any proposed modifications or deletions are documented and forwarded to Requirements Management (Section 4.3).

#### 4.5.4.4.3 Refined Work Breakdown Structure

The selected design's Physical Architecture is used to refine the Work Breakdown Structure (WBS) by translating the decomposition into a WBS format. The refined WBS provides

enhanced work planning, cost/schedule tracking, and control by extending the existing WBS to account for the system elements identified during Synthesis.

#### 4.5.4.5 Initiate evolutionary development

An evolutionary development is initiated, if necessary, for any design element for which a lesser technology solution was selected over a higher-risk technology, and for which the capacity to evolve was designed into the element and interfacing elements. (See Trade Studies (Section 4.6).)

### 4.5.5 Metrics

Performance of the Synthesis process itself shall be measured on a regular basis and recorded in the metrics library on a monthly basis. The engineering team, minimally, is to use the following metrics to evaluate performance:

1. Trade Study Satisfaction Assessment (see Trade Studies (Section 4.6))
2. For approved engineering problem reports:
    a. Quantity, by type of problem report
    b. Cycle-time from disposition to incorporation of change into released engineering documents, by type of report
3. Technical Performance Measurements: objective versus achieved values
4. Number of approved engineering changes: by product, type, and stage
5. Documents/drawings submitted for engineering release:
    a. Unacceptable submittals
    b. Total submittals
6. Number of technical action items identified during reviews and audits
7. Design efficiency metrics, such as weight, required power, and envelope dimensions (volume)
8. Cost and schedule variance for the completion of Synthesis steps
9. System requirements not met
10. Number or percent of system requirements verified by system analyses
11. Number of TBDs in system architecture or design
12. Number of interface issues not resolved
13. Percent of identified system elements that have been defined

### 4.5.6 Tools

#### 4.5.6.1 Schematic Block Diagrams

Along with the definition of design alternatives, it is important to establish the relationships between alternatives at each level of design activity. A medium for accomplishing this is by using SBDs.

A simplified SBD shows the components that may comprise an element and the data that may flow between them. An expanded version is usually developed that displays the detailed functions performed within each component and their interrelationships. For complex systems, this may then be developed into a logic diagram for auditing the schematics produced by design engineers. This audit is a critical SE function. Interface information should also be embedded into the SBDs, as appropriate. The interface data will form the basis for the interface specifications to be developed at multiple levels of the system hierarchy. An $N^2$ diagram is very useful for developing and auditing interfaces at all levels.

If software is an element of the design, it must be determined whether a given function will be accomplished in hardware or software. Computer Software Elements (CSE) should be defined during this step of the process and embedded within the SBDs. Experience shows that it is helpful to first define the top-level hardware CE in which a given software function will reside before defining which candidate CSEs will accomplish the function. Additionally, as part of Section 4.5.3.7 of the Synthesis process (Physical Architecture Definition), it is recommended that a given function be tracked to determine whether it has been allocated to a software alternative or a hardware alternative. Determining the appropriate level of the system hierarchy for defining CSEs is largely project dependent.

The products of this step of the SE process are a set of viable system alternatives responsive to the design goals and a series of SBDs depicting how the alternatives interrelate.

#### 4.5.6.2 Computer-Aided Design

Modern computing hardware and software is used to convert the initial idea for a system into a detailed engineering design. The evolution involves creating geometric system models that are later manipulated, analyzed, and refined.

### 4.5.7 References

1. Defense Systems Management College. "Systems Engineering Fundamentals." Fort Belvoir, VA: Defense Systems Management College Press, 1999
2. EIA Interim Standard, Systems Engineering Capability Model, EIA/IS-731.1, December 1998
3. EIA Standard, Processes for Engineering a System, EIA-632, January 7, 1999
4. Guidelines for Approval of the Provision and Use of Air Traffic Services Supported by Data Communications, RTCA/DO-264, Prepared by SC-189, December 14, 2000
5. Handbook of Systems Engineering and Management, Andrew P. Sage and William B. Rouse, ©1999
6. IEEE Standard for Application and Management of the Systems Engineering Process, IEEE Std-1220-1998
7. International Council on Systems Engineering. "Systems Engineering Handbook," Version 2.0, 2000
8. ITS Physical Architecture Physical Architecture, Lockheed Martin Federal Systems, Odetics Intelligent Transportation Systems Division, September 1998
9. KAREN Physical Architecture Physical Architecture, Version 2, Siemens Traffic Controls Limited, October 1999
10. Systems Engineering Management, Second Edition, Benjamin S. Blanchard, ©1998
11. Systems Engineering Guidebook, A Process for Developing Systems and Products, James N. Martin, ©1997

## 4.6 Trade Studies

Trade Studies is the System Engineering (SE) element used by multidisciplinary teams to identify the most balanced technical solutions among a set of proposed viable solutions. It is a key tool in developing designs that meet stakeholder requirements in the most cost-efficient manner possible. The application of Trade Studies prevents program/project management from committing too early to a design that may not be cost-effective or meets all system requirements. Through Trade Studies, desirable and practical alternatives that better combine cost and effectiveness may be identified, resulting in beneficial selections among the alternatives. Figure 4.6-1 depicts the Trade Studies Process-Based Management chart.

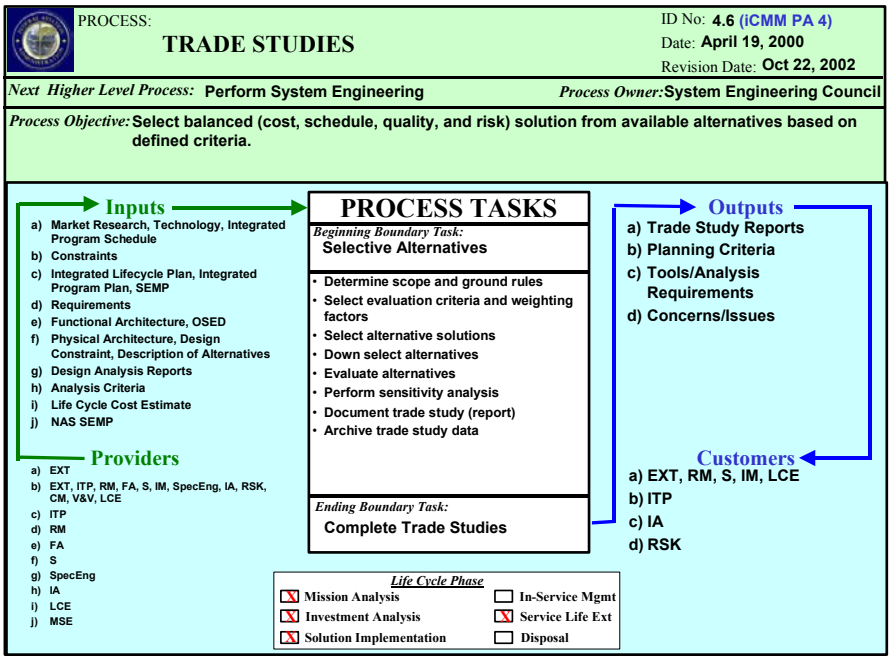



**Figure 4.6-1. Trade Studies Process-Based Management Chart**

Conducting Trade Studies involves evaluating two or more alternatives to select a preferred option. The Trade Studies process balances such considerations as cost, reliability, testability, supportability, survivability, compatibility, and producibility during each phase of the product development cycle.

A disciplined Trade Studies process is required to fairly evaluate alternative concepts and designs. The process requires that any affected discipline participate in the program/project to the extent needed to arrive at the best-balanced requirements solution. Typically, a Trade Studies leader, who is not a stakeholder in any of the proposed solutions, helps to focus and coordinate the flow of information that occurs during the Trade Studies process.

Trade Studies may be formal and informal, with different emphases, depending on when in the program lifecycle they are conducted. It is appropriate to develop a Trade Studies plan (Integrated Technical Planning (Section 4.2)) for each major problem or issue for which alternatives are being considered.

This section describes the Trade Studies process as a formal decisionmaking methodology used to select among alternative concepts, designs, products, or approaches that satisfy the system implementation and to resolve any conflicts that arise during the system's lifecycle.

### 4.6.1 Introduction to Trade Studies

Trade Studies are conducted within and across disciplines to support decisions at any phase of the program's lifecycle. The process quantifies and/or qualifies the consequences of selecting various system alternatives in terms of metrics that are traceable to customer requirements and are declared by project management to be project objectives. They support the allocation of performance requirements and the determination of design constraints and are used in evaluating alternative functional architectures obtained from Functional Analysis (Section 4.4). In general, the results of the Trade Studies process may be quantitative or qualitative in nature.

Trade Studies may be performed at any step in the system's lifecycle, but the process begins at the Mission Analysis (MA) phase and continues through first article production. For example, the major goal of the Investment Analysis (IA) phase is to define a set of system requirements that meet the goals and objectives of a mission or a system at an affordable cost and with an acceptable level of risk. During this phase, Trade Studies may be used to select among competing sets of requirements that define alternative system concepts. In a similar manner, the Trade Studies process is used to assist SE.

The following list summarizes the use and emphasis of Trade Studies in the program's lifecycle:

MA phase:

- Define mission requirements
- Resolve conflicting high-level customer requirements
- Evaluate alternative high-level requirements to meet mission needs

IA phase:

- Compare technologies and approaches
- Evaluate concepts to meet high-level requirements
- Select alternative system configurations for further study
- Select concept for preliminary design development and conceptual layouts
- Support Functional Analysis (Section 4.4) and allocation of performance requirements (alternative architectures)
- Establish system configuration
- Support decision for new product development versus nondevelopment products
- Establish system, subsystem, and component configurations
- Select testing methods
- Determine installation locations; check for fit and compatible environment
- Detail design
- Define a best-value design solution that satisfies all system requirements

- Support detailed design analysis
- Compare manufacturing processes
- Determine best order of assembly

Solution Implementation (SI) phase:

- First article, full-scale development
- Resolve unexpected manufacturing issues, such as changing the order of assembly or revising a manufacturing process
- Select alternative designs, solutions (operations, maintenance, integrated logistic), procedures, and configurations

#### 4.6.1.1 Trade Studies Objectives

Trade Studies are conducted at the program's different lifecycle stages to discover the best-value solution, best value to the government, and best value to a set of requirements from technical, cost, or schedules points of view. Trade Studies, also referred to as tradeoff studies or selection studies, are performed for a variety of purposes, including to:

- Choose among alternative design and implementation strategies and solutions based on architecture, performance, and cost in order to meet stakeholder requirements
- Recommend commercial-off-the-shelf (COTS) products for acquisition
- Perform make versus buy analyses, or buy versus lease analyses (Office of Management and Budget (OMB) Circular a.76, Outsourcing Decision)
- Recommend a supplier for services
- Document and justify the selection of a solution for a system requirement
- Reduce risk

Trade Studies provide an objective determination of comparative metrics for various system options. An essential aspect of the analyses performed for these studies is that consistent, configuration-controlled parameters be used in the computations to ensure comparison of likely system solutions.

#### 4.6.1.2 Participants

All elements of the project organization are responsible for Trade Studies. The process requires the participation of various interdisciplinary skills in an integrated manner with the objective of producing an optimum system design.

Design, manufacturing, test, operations, and product support perform lower-level Trade Studies that involve subsystems, components, subcomponents, and software. In the event of utilization of system-level resources contention, program/project management coordinates with the stakeholder organizations to resolve issues and establish priorities. It is recommended that Trade Studies affecting hardware and software account for system issues related to software, operations, procedures, training, and other nonmaterial-related solutions.

To determine impacts across interfaces, it is recommended that SE integrate the Trade Studies performed by various groups.

### 4.6.2 Inputs to Trade Studies

Inputs to the Trade Studies process may be divided into two categories: stakeholders and project. The stakeholder inputs include the operations concept, program requirements, and system requirements. The project inputs include design analysis report (DAR), Functional Architecture (Section 4.4), DAR (Section 4.8), results from Validation and Verification (Section 4.12), and Lifecycle Cost Estimates from Lifecycle Engineering (Section 4.13).

The Trade Studies process presupposes that alternatives have been identified that are evaluated as specified by the process objective. To complete this task:

- Requirements, Constraints, expectations, assumptions, goals, and regulations shall be clearly understood
- Design options, including Baseline and other criteria, shall be provided or developed
- Relevant plans and documents shall be provided

### 4.6.3 Trade Studies Process Tasks

The methodology to evaluate system alternatives is described in the following paragraphs. The Trade Studies process consists of the following tasks:

- Determine scope and ground rules
- Define evaluation criteria and weighting factors
- Select alternative solutions (brainstorm possible solutions), if not provided
- Down-select alternatives
- Evaluate alternatives
- Perform sensitivity analysis
- Review results and form conclusions
- Document the Trade Studies

These steps seldom are performed sequentially. Certain steps, such as definition of evaluation criteria, may be repeated several times as alternatives are defined and evaluated. Figure 4.6-2 depicts the overall Trade Studies process.

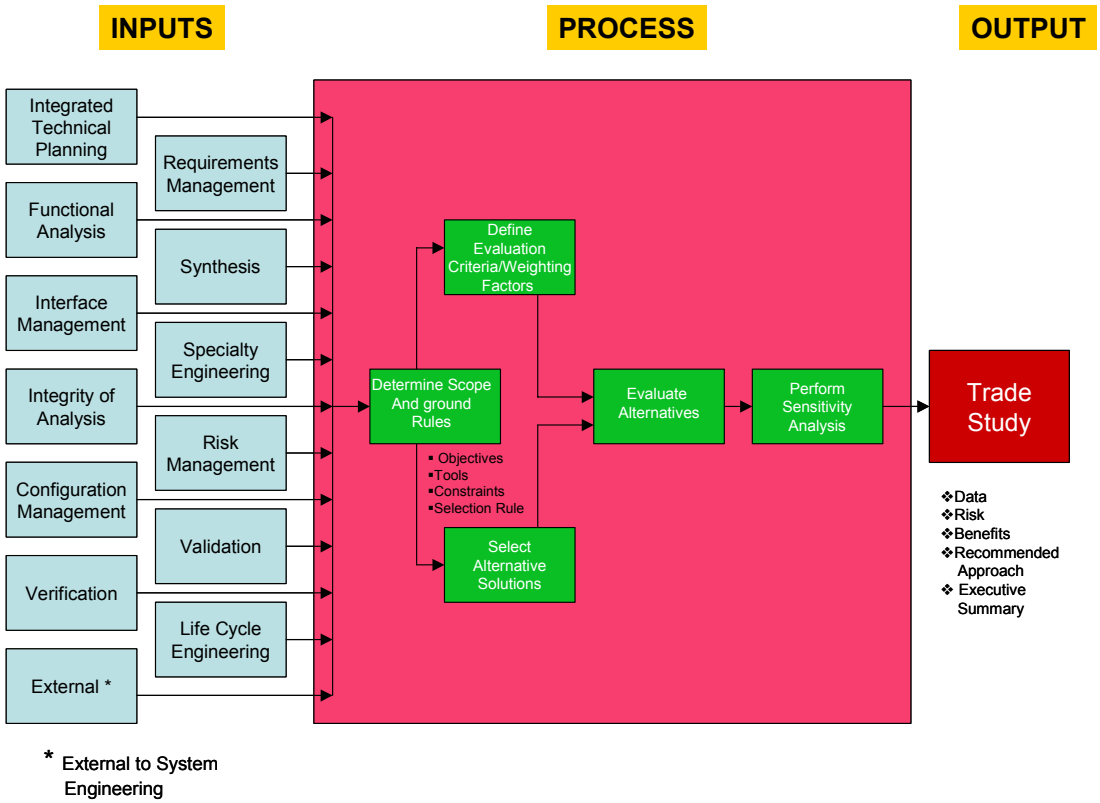



**Figure 4.6-2. Trade Study Process**

### 4.6.3.1 Task 1: Determine Scope and Ground Rules

To complete Task 1, perform/consider the following checklist of actions/issues:

- Determine the specific goals of the Trade Studies and the Requirements to be met before establishing the scope and methods of the study:
  - Consider all viewpoints of stakeholders (e.g., users, developers, managers, and operations and maintenance personnel) to accomplish this goal
  - Ensure that input is obtained from all customers associated with the study and that the stakeholders' viewpoints are clearly understood and documented
  - Understand and resolve differences between competing viewpoints and any underlying biases before continuing the evaluation process
- Use the methodology described in Requirements Management (Section 4.3) to define and analyze the Requirements for the Trade Studies:
  - Select Requirements to bound the Trade Studies into four major categories: functional, performance, operational, and programmatic
  - Base the Requirements on the goals established for the study and adjust the level of detail of the Requirements to the scope of the particular study

  - Ensure that the Requirements, which are used as a basis for criteria against which alternatives are evaluated, are accurate, unambiguous, verifiable, complete, and appropriate
  - Obtain the customer's approval on the goals and Requirements for the tradeoff study
- Define the system's goals and objectives and identify the Constraints to satisfy:
  - Recall that in the early phases of the system's lifecycle, the goals, objectives, and Constraints are usually stated in the operational terms; when the system architecture and design have been determined or established, the goals and objectives are usually stated as performance requirements
- Spend time up front clearly defining the problem and jointly coordinating with the respective internal and external customers the key Requirements that any solution needs to meet. Achieving consensus with affected team leaders regarding the real problem to be resolved saves significant time in the overall process.
- Establish a multidisciplinary team that is able to support the analysis effort from start to finish. Having expertise within each discipline ensures that alternatives are thoroughly evaluated, leading to the most accurate assessment results. Available budget and time control most studies; therefore, when equipped with this information, team members realize how far they may pursue alternatives.
- Develop an attainable schedule as well as identify major Trade Studies milestones. It is recommended that the degree to which excursions from the baseline concept are allowed also be defined. A study lacking clear boundaries easily grows far beyond the available resources.

It is recommended that the Trade Studies team leader coordinate items that influence subsystems and assess the impact on his/her area. It is also recommended that subcontractors, as well as those on the Trade Studies team, consider and identify previously developed hardware and software components, non-developmental items (NDI), and COTS hardware and software as candidates for utilization in the Trade Studies. Additional items for the team to consider and identify are common components in different parts of development to share across development groups or across configuration items.

Before the Trade Studies process is conducted, the decisionmaking body responsible for the affected baseline shall approve the Trade Studies plan.

#### 4.6.3.2 Task 2: Define Evaluation Criteria and Weighting Factors

The definitions of measures and measurement methods for system effectiveness, system performance, and system cost are related to the definition of goals and objectives and Functional Analysis (Section 4.4) performance. These measurements are the decision criteria. Each quantitative measure shall have a defined measurement or computational method. This task initiates the analytical portion of the Trade Studies process, as it involves using quantitative methods.

The definition of evaluation criteria requires considerable engineering judgment and interaction with the stakeholder to establish the appropriate criteria, associated weights, and scoring methods. For example, supporting missions with tight schedules requires heavy weighting of schedule risk, while supporting missions with more flexible schedules generally emphasizes low

cost while accepting higher schedule risks. Sufficient comments shall be provided for each evaluation criterion to ensure evaluator and stakeholder comprehension. Stakeholder approval shall be obtained before proceeding to the next task.

The technical requirements that potential solutions need to achieve serve as the criteria against which alternative concepts are measured. The selected criteria may include limits of minimum acceptable values and desirable attributes that permit judging of candidates against each other. Trade Studies leaders are encouraged to use Quality Function Deployment (QFD) to help to define the evaluation criteria and weighting factors applicable to the Trade Studies. These criteria are defined based on the technical requirements that determine if a design is acceptable to the Stakeholder Needs.

Evaluation criteria are more meaningful if they represent measurable characteristics, which is not always possible. It is recommended that criteria on cost and programmatic risk be included. Alternatives may be evaluated based on projected fixed and variable cost using risk factors, when applicable, to derive expected costs. It is also recommended that elements not directly related to cost (e.g., weight, production cycle time) have criteria established to associate cost with changes in the elements. Trade Studies shall address these criteria.

An experienced, multidisciplinary team shall brainstorm a list of additional criteria suitable for the study's intent if all feasible alternatives are to be identified and thoroughly evaluated. Each criterion shall be described to a level of detail such that its intent is clear to all team members. This detail ensures that all participants are well aware of specified and derived Requirements affecting evaluation.

When a particular study is planned, the effort and cost of that study shall be balanced against the impact (e.g., cost, schedule, and technical risks) on the study's scope and methodology. An overly ambitious and costly study among low-impact alternatives is as serious as the failure to adequately evaluate high-impact alternatives. For a simple evaluation of several low-impact alternatives, subjective evaluation and consensus may be sufficient. For complex studies with higher impact, the following is recommended:

- Define evaluation criteria based on the Requirements analysis.
- Determine relative weights for the evaluation criteria based on the Requirements analysis.
- Prepare a scoring matrix that assigns a row for each evaluation criterion and a column for each alternative to be evaluated, with comment fields for each criterion.
- Define a method for assigning a score to each element in the scoring matrix.
- Assign a score for each criterion for each alternative:
  - Select scores in such a manner that the higher the score, the more favorable the evaluation; use an odd number of integers so that the middle score represents an average rating
  - Use small integers, typically 0 to 5, to represent scores; a range of 0 to 2 may be adequate; a range in excess of 0 to 10 is not recommended
  - Determine a method of recording items that is unable to be scored; define the scoring method to be used; recording a blank for unknown information often is useful

- Prepare a weighted score matrix that assigns a row for each evaluation criterion and its weight and assigns a column for each alternative to be evaluated. The weighted score recorded for each element in the matrix is the product of the weight for that criterion and the corresponding score in the scoring matrix.

Figure 4.6-3 is a sample decision analysis matrix.

| Decision Factors / Alternatives | Decision Factor 1 | | Decision Factor 2 | | ... | Decision Factor n | | Total Weighted Score |
|---|---|---|---|---|---|---|---|---|
| | Weight = 1 | | Weight = 1.5 | | ... | Weight = 2.5 | | |
| | Score | Weighted Score | Score | Weighted Score | ... | Score | Weighted Score | |
| Alternative 1 | 7 | 21 | 4 | 6 | ... | 8 | 20 | 47 |
| Alternative 2 | 8 | 24 | 10 | 15 | ... | 10 | 25 | 64 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ... | ⋮ | ⋮ | ⋮ |
| Alternative n | 10 | 30 | 5 | 7.5 | ... | 12 | 30 | 67.5 |
| Alternative n+1 | ⋮ | ⋮ | ⋮ | ⋮ | ... | ⋮ | ⋮ | ⋮ |

**Figure 4.6-3. Decision Analysis Matrix**

Stakeholder approval of the proposed evaluation method shall be obtained.

Stakeholders and internal technical experts are used to establish meaningful evaluation criteria. Criteria for which all potential alternatives are equal in value are not used in the evaluation because they do not add value to the process.

Defining evaluation criteria often requires several iterations before the final criteria are determined. Evaluation criteria are defined based on the analysis of Requirements. Bias shall be avoided when evaluation criteria are established (e.g., acceptance of an existing system or product as the de facto standard for evaluation). The following evaluation criteria are applicable to a wide range of Trade Studies:

- Development cost
- Lifecycle cost
- Requirements compliance
  - Functional
  - Performance
  - Operational
  - Programmatic
- Technical risk (Maturity)

1. Reliability, Maintainability, Availability (RMA)
2. System Safety
3. Quality
4. Human Factors
5. Electromagnetic Environmental Effects (E3)
6. Hazardous Materials

- Budget risk
- Schedule risk
- Operational complexity
- Vendor assessment
- System maturity
- Development support tools
- Test support tools
- Development team familiarity with candidate hardware and software
- Quality of logistics support

Evaluation criteria that apply specifically to the Trade Studies shall be selected, adding additional relevant criteria, such as security, as needed. For each evaluation criterion, established threshold values that may be used to evaluate the alternatives on a pass/fail basis shall be identified. An example criterion is: “The system MTBF shall be 10,000 hours or greater." For the remaining criteria, a weight and scoring range shall be assigned for use with the weighted matrix evaluation method.

Criteria are ranked and grouped into three categories so that the assigned weights reflect their criticality. The most critical criteria are assigned large weights and flagged so that any alternative with low scores for these criteria influence any subsequent analysis. Mid-critical and noncritical criteria are assigned smaller weights; it is recommended that noncritical criteria have a negligible effect in further analysis.

#### 4.6.3.3 Task 3: Select Alternative Solutions

Once the evaluation method is established, all available resources are used to develop viable alternatives and solutions. Trade publications, prospective bidders for service contracts, technical staff, stakeholders, and managers, as appropriate, are helpful resources in developing a set of alternatives that may potentially achieve the goals and objectives of the system (e.g., architectures, designs, COTS products).

Based on defined ground rules, the alternative development phase is intended to evaluate multiple alternatives and narrow the prospects for extensive evaluation. The importance of creativity is especially emphasized, as this task may or may not affect the alternative design solutions previously submitted.

The evaluation criteria and detailed Requirements shall be used to synthesize alternative solutions. In defining alternative approaches, developing the alternatives often requires

lower-level Trade Studies, which enables a hierarchy of design alternatives. A trade tree that reflects the complete hierarchy of trades to address when performing the top-level Trade Studies shall be drawn. The trade tree shall contain a number of high-level system architectures, which prevents focusing on a single architecture. To eliminate undesirable alternatives, for each trade item in the trade tree, the tasks in the sections above shall be repeated until a complete trade tree is generated, and the objectives, Requirements, evaluation method, and evaluation criteria are defined. Top-level objectives and Requirements are allocated to successively lower levels of Trade Studies in the trade tree. The allocated objectives and Requirements are used to define the evaluation methodology and criteria, and evaluation is performed, as described in the following paragraphs.

Each design alternative shall be thoroughly assessed. Potential design approaches for each Requirement shall be reviewed against potential approaches for other Requirements in order to identify possible interactions. It is recommended that interactions that might affect the cost of, or make one solution feasible, be documented and handled as linked decisions throughout the Trade Studies process.

Often, risk is the deciding factor in candidate selection. A complete technical analysis identifies and quantifies technical risks and develops contingency alternatives. Therefore, the technical and schedule risks associated with each candidate system are identified, and the probable gain and loss for each risk are analyzed. Also, an acceptable level of risk for a given gain is defined, and efforts are undertaken to minimize new, unproven, complex, or unusual Requirements for hardware, software, and firmware. The use of untried elements is minimized by recommending proven substitutes whenever possible.

A technical analysis of schedule risk areas is performed, and all long-leadtime items, which are the schedule drivers, are identified. How the design affects the development schedule is discovered, and all system elements and resources that may be available when needed are determined. All single-source items that may be potential risks are identified, and a recommended level of schedule contingency, as appropriate, is defined.

Expected operational scenarios for each candidate system to assess the interactions of the design alternatives are defined. Also, the expected system growth over the planned system life is determined to assess system design flexibility and expandability. Because system sizing is based on the anticipated workload, every effort to ensure an accurate workload forecast is made, as improperly sized systems result in unnecessary cost and/or insufficient capacity. Human workload and scenario definitions are used as drivers to assess performance, utilization, and capacity of the system under anticipated operational conditions. (Specialty Engineering (Section 4.8) provides guidance on this topic.)

Once a set of possible alternatives has been selected, the next task is to collect data on each to support the evaluation of the measures by the selected method. The data collection, directed by the Trade Studies leader, emphasizes the role of the disciplines, such as reliability, maintainability, integrated logistics, producibility, software, testing, operations, and costing. Figure 4.6-4 is an example of a Trade Studies table.

| Criteria | Alternative 1 | Alternative 2 | - - - | Alternative N |
|---|---|---|---|---|
| Cost<br>▪Initial<br>▪Recurring | High<br>low | Medium<br>Low | - - - | Low<br>Low |
| Performance | Low | High | - - - | New Design |
| Reliability<br>Maintainability<br>Availability | Medium | High | - - - | Medium |
| Risk<br>▪Cost<br>▪Technical | Low (20%)<br>Low (25%) | Low (10%)<br>Low (20%) | - - - | Low (10%)<br>Medium (35%) |

**Figure 4.6-4. Example Trade Study Table**

#### 4.6.3.4 Task 4: Down-Select Alternatives

When numerous possible alternatives are identified, a detailed analysis of each one may not be cost-effective; therefore, down-selection of candidates is recommended. Identifying high-risk candidates and candidates with questionable feasibility or high lifecycle cost helps to reduce the number of alternatives to be studied. Screening the alternatives against the selection criteria eliminates these candidates. If one of a closely grouped set of alternatives is down-selected, it is recommended that all alternatives in that group be down-selected. Any relationship that is not the same for each down-selected alternative and the baseline becomes part of the detailed Trade Studies. Each alternative is defined to an appropriate level of detail to differentiate the alternative with respect to the technical requirements, which typically include layouts, tooling concepts, cost studies, and other detailed analysis. When only the down-select Requirements are the focus, the effort is simplified to only those Requirements that are different among the design alternatives and the baseline.

The down-selected alternatives are provided to all disciplines involved to ensure that each has the opportunity to evaluate the impacts. This process provides discrete impacts for each area used to select the preferred alternative. It is recommended that this process be performed in parallel with each discipline preparing its inputs simultaneously.

#### 4.6.3.5 Task 5: Evaluate Alternatives

The next task in the Trade Studies process is to quantify the outcome variables by computing estimates of system effectiveness, underlying system performance or technical attributes, and system cost. If the needed data has been collected and the measurement methods (e.g., models) are in place, this step, in theory, is mechanical. In practice, considerable skill often is needed to obtain meaningful results.

Recommended Task 4 actions include the following:

- Perform a detailed evaluation of all approved viable alternatives. An individual or a small group may perform this evaluation. Record any problems or questions. If a weighted matrix method is used, finish scoring without reference to weights or flags.

- Evaluate the alternative approaches relative to the evaluation criteria when performing the Trade Studies process.
- Identify any alternatives with high-weighted scores that narrowly failed the pass/fail criteria. Discuss these alternatives with the stakeholder.
- Evaluate cost factors separately from the remaining evaluation criteria throughout the process. In some cases, none of the alternatives may satisfy all pass/fail criteria. In such cases, relax one or more pass/fail criteria, investigate additional alternatives, or report to the stakeholder that no entirely acceptable alternative has been found.

Ideally, all input values are precisely known, and models perfectly predict outcome variables. Since this case is not typical, it is recommended that the Trade Studies leader supplement point estimates of the outcome variables for each alternative with computed or estimated uncertainty ranges. For each uncertain key input, it is recommended that a range of values be estimated. Using this range of input values, the sensitivity of the outcome variables may be gauged, and their uncertainty ranges calculated. Figure 4.6-4 is an example of a Trade Studies table.

The baseline reference method, relative rank method, and cost assessment method are several methods used to evaluate alternatives and are discussed in the following paragraphs.

#### 4.6.3.5.1 Baseline Reference Method

The baseline reference method requires a baseline or legacy design and a set of associated databases on the use of that design. Alternatives are evaluated against the baseline design or other reference using the selected evaluation criteria. If an alternative is clearly better than the baseline, it is marked as a plus (+); clearly worse than the baseline (-); same as baseline (S); and unacceptable as the baseline (U). This evaluation requires a team effort of all disciplines participating in the study, with team agreement for each rating. It is recommended that notes be maintained as to why ratings are given for each relationship. Using numbers or ++/-- may expand the sensitivity of the +/- system. However, doing so slows the evaluation process and places dangerous emphasis on the matrix as a tool that delivers answers more definitive than the process warrants. When making the +/- decision, the magnitude of the difference shall be considered; however, the process of marking an only marginally better feature as + compared to the baseline shall be avoided.

Generally, alternatives with a U relationship are eliminated, or the U condition is removed; however, there are exceptions to this rule. An exception may be when the Trade Studies process is conducted to determine whether there are sufficient benefits from an alternative to justify a request for a specification change. Also, an alternative in a study may present itself that significantly improves the overall system performance but requires a specification change. It is recommended that common sense be used when U relationships are evaluated and that the users' needs be considered.

Once relationships are defined for each alternative and technical requirement, the overall value of merit of the alternative is calculated. A value of +1 is assigned to each (+) rating, and a -1 to each (-) rating. A relative weight may also be assigned to each evaluation criterion if not all criteria are considered equal. QFD may help to determine this importance weighting.

It is recommended that the following actions be taken when the baseline reference method is used:

- Multiply the importance weighting and the evaluation and then perform the sum calculation for each alternative. No calculation is performed for the same evaluations because this method of evaluation is relative to the baseline. The overall importance rating is a figure of merit for each alternative. The higher the importance rating, the better the alternative for the given design requirement. However, this guide is only relative. Do not differentiate alternatives by closely grouped importance ratings. If, for example, three concepts fall in a range of 10-20 and the other is -30, the alternatives in the group 10-20 are basically equivalent.
- Review each alternative to gain an overall understanding of the meaning of the final importance rating. It is recommended that the team review all the alternatives with negative relationships and develop supplemental alternatives that eliminate these negatives, resulting in additional viable alternatives. Some of these alternatives use portions of the previously developed alternatives. The development and evaluation of subsequent alternatives shall follow the procedures used for initial alternative development. When supplemental alternatives are developed, low sensitivity of the +/- system is avoided. Developing supplemental alternatives is critical to a successful Trade Studies. A "zero change" option normally is included for comparison.

#### 4.6.3.5.2 Relative Rank Method

The relative rank method uses the Kepner and Tregoe technique to evaluate alternatives. This technique evaluates each alternative against the selected criteria and establishes a ranking for each criterion. Weighting of the criteria is defined by category, while the trade options are graded in their appropriate columns according to the scaling factors over the range 0 to 4. The average ranking within each category is multiplied by the criteria weighting to determine a score. Scores are summed across the criteria for a total.

#### 4.6.3.5.3 Cost Assessment Method

The cost assessment method is similar to the baseline reference method, with the exception that the alternatives are reduced to rough order of magnitude (ROM) estimates of fixed and variable costs. Elements that do not reduce directly to cost (e.g., weight, production cycle time) are translated to cost using common criteria described in “Task 2: Define Evaluation Criteria and Weighting Factors” (Paragraph 4.6.3.2). If risks are present, risk projections are used to calculate an expected value.

As cost is a major factor in selecting among candidate systems during system design, development, implementation, and operational costs shall be considered when the lifecycle costs of candidate system configurations are evaluated. A refinement of earlier ROM cost estimates is required to complete the information needed to select the system configuration. It is recommended that the estimate include estimates submitted by major subcontractors and vendors and contain sufficient cost detail to answer client questions.

In addition, it is recommended that the following actions be taken when the cost assessment method is used:

- Determine the relative complexity and risk of each candidate system configuration.
- Identify how each candidate system configuration proposes to handle stringent system requirements, such as response time, transaction processing time, and throughput.

- Analyze how each candidate configuration meets special system requirements for a high level of reliability and availability or for quick recovery or automatic failover.
- Highlight key factors that result in lower cost and risk. Discuss the factors with the stakeholder, including the option of analyzing a more simple system that addresses only the most critical requirements set. This type of analysis gives the stakeholder a minimum system cost benchmark to assess cost of the candidate system and functionality of each requirement.
- Include the tradeoffs among hardware, software, and manual operations as part of the cost analysis, and identify the most sensitive cost drivers of each candidate system. If the system has security requirements, also consider security cost drivers.

#### 4.6.3.6 Task 6: Perform Sensitivity Analysis

Sensitivity analysis is used when the solutions are nearly equivalent in scoring and, in some cases, may be required even if the scoring is equivalent.

Recommended Task 5 actions include the following:

- If using a weighted matrix evaluation method, analyze all alternatives to determine if the differences between the scores are truly significant and if minor variations in the raw scores and weights might affect the selection. Reference any questions or problems noted by evaluators. For each compliant alternative, including any solution that is compliant based on redefined pass/fail criteria, determine if any weighted score or total for a group of related weighted scores is sensitive to variation of weights or scores.
- Evaluate the effect on weighted scores of varying weights. If some weights are determined by compromise, the range of reasonable values discussed during the definition of evaluation criteria (Paragraph 4.6.3.2) provides useful guidance for such variation.
- Evaluate the sensitivity of weighted scores to variation of scores. If a number of evaluators have evaluated the alternatives against a given criterion, the range of scores recorded provides useful guidance for such variation.
- Record the ranges of scores and weights evaluated for each criterion. Compute the upper and lower bound for weighted scores (and groups of weighted scores). Document the data in a matrix corresponding to the score and weighted score matrices.
- By inspection or use of a suitable statistical test, determine if any of the variations are large enough to require special attention (i.e., more detailed investigation to ensure the accuracy of the evaluation).
- Evaluate the effect on weighted score totals, including or excluding criteria flagged as noncritical.

Typical outcomes of the sensitivity analysis and review of results include the following:

- **Case 1:** One alternative emerges as the optimal choice if it meets all critical requirements, has the highest weighted score (with a range that does not overlap the range of another alternative), and has the lowest cost.
- **Case 2:** A cluster of alternatives is acceptable (i.e., each alternative in the cluster satisfies all critical requirements, its weighted scores have overlapping ranges, and its cost is competitive).

- **Case 3:** No single, entirely satisfactory alternative is found.

Case 1 is the most straightforward for the stakeholder. Case 2 may be resolved by reviewing evaluation results with the stakeholders. If a weighted matrix evaluation method is used, inspecting the score and weighted score matrices may reveal patterns that are helpful and clear in the decisionmaking process. A review of weights and criteria may indicate that weights may be modified, which may resolve the overlap. Additional factors may be identified as criteria to resolve the overlap. If the overlap of weighted scores persists, the lowest-cost alternative may be selected. Case 3 is the most difficult to resolve. A review of evaluation criteria, especially pass/fail and critical criteria, may indicate that no satisfactory alternative has been identified by the study. In this case, engineering judgment and discussions with the stakeholder shall be used to define additional alternatives or to accept a less than optimal alternative.

Figure 4.6-5 depicts typical utility curves used for sensitivity analysis.

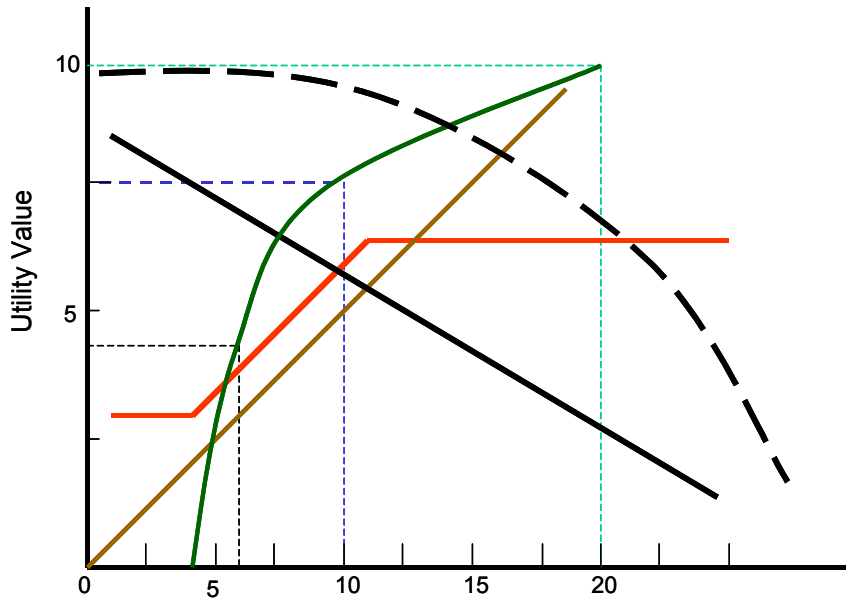



**Figure 4.6-5. Example Utility Curves**

### 4.6.3.7 Task 7: Review Result and Form Conclusions

This part of the Trade Studies process typically presents one alternative that balances the Requirements and a "zero change" option for comparison. While the defined decision authority makes the final decision, a recommendation by the Trade Studies team is essential. All results shall be reviewed, any necessary additional data obtained, and evaluations and preliminary conclusions revised as needed. Any or all parts of the study may be repeated.

If the evaluation's intent is to select a product or service, it may be useful to review preliminary conclusions with vendors to ensure that no misunderstandings have occurred. Delaying such reviews until this phase of the evaluation avoids much of the risk of biasing the overall process.

When the evaluation is completed and deemed reliable, cost estimates for each alternative shall be prepared. Weighted scores for evaluation criteria are related to benefits associated with the

evaluated alternative. The cost of an alternative divided by the total score for that alternative is a measure of the cost/benefits of that alternative.

At this point, the alternatives are now ordered based on the technical requirements and quantified impacts. It is recommended that changes from the baseline design technical performance and the decision criteria used during this evaluation be highlighted.

### 4.6.4 Outputs of Trade Studies

The outputs of the Trade Studies process are a report with an executive summary and a design/manufacturing decision document.

#### 4.6.4.1 Trade Study Report

A Trade Study Report is prepared for each study. The report documents the study results and provides traceability to decisions made during the program's lifecycle. The report provides the traceability needed to substantiate design and configuration changes to the baseline design and also documents the decisionmaking process that selected one alternative over another. Additionally, it describes the effects of selecting a particular alternative among trades and clearly notes affected areas that were included in the Trade Studies assumptions, as well as affected areas that were not included in the associated trade. Once the report is completed, the Trade Studies leader is expected to coordinate the report with all affected team leaders before submitting it for approval and signature.

The Trade Study Report is prepared using a format appropriate for documenting and communicating the results, conclusions, risks, benefits, and recommendations to the decisionmaker. It is recommended that the format be standardized wherever possible to satisfy individual program needs. At a minimum, it is recommended that the following be included, but not limited to:

- Clear problem statement
- Identification of affected Requirements
- Ground rules and assumptions
- Decision criteria
- Resource requirements statement to accomplish the study
- Schedule to accomplish (proposed and actual)
- Evaluation of all potential solutions and screening matrix
- Comprehensive array of feasible alternatives
- Comparisons of alternatives using decision criteria
- Technical recommendation of the Trade Studies team
- Documentation of any decisions leading to the final technical recommendation

The following is a suggested report format. Each project may enhance the standard outline as needed by adding subsections and separately numbered items to the sections. Each project may also add sections and subsections for special topics and delete sections and subsections that are not applicable.

Table of Contents

1 Introduction
1.1 Scope
1.2 Applicable Documents
1.3 Definitions
2 Study Summary
3 Requirements Summary
4 Evaluation Criteria
4.1 Evaluation Criteria and Scoring Method
4.2 Evaluation Criteria and Changes During Study
5 Alternative Solutions
6 Results
6.1 Evaluation Approach, Scores, and Analysis
6.2 Conclusions
Appendices
List of Acronyms
References

#### 4.6.4.2 Design/Manufacture

Once the Trade Study Report is approved, the design decision/manufacturing document is produced, outlining the impacts and actions necessary to implement the alternative recommended in the Trade Studies into the baseline configuration. In general, this document describes the rationale required to substantiate the change. The report then becomes an attachment to the design decision/manufacturing document and serves as the technical basis for the option to be implemented. The design decision document is submitted to the appropriate control authority to authorize implementation into the baseline configuration. The control authority is also required to maintain the report and the design/manufacturing decision document for the program's lifecycle.

### 4.6.5 Trade Studies Tools

#### 4.6.5.1 Quality Function Deployment

QFD is a methodology used to ensure that the stakeholders' operational needs and requirements are gathered, interpreted, and deployed in developing a product or service. The primary objective of QFD is to eliminate three major problems: difficulty in gathering and interpreting stakeholder's requirements; loss of information; and different individuals and functions using varying interpretations of the same requirements. QFD provides a Trade Studies tool that screens alternatives using weighted selection criteria. QFD is recommended for use whenever:

- Stakeholder requirements are vague, ambiguous, or self-contradictory
- Multiple disciplines are involved in the collection and interpretation of the requirements
- Multiple solutions are feasible with no clear choice
- Lack of an obvious feasible solution
- Cost and/or risk appear to be unacceptably high

QFD (see http://www.shef.acu.uk/~ibberson/qfd.html) requires teamwork among the multiple disciplines that make up a program/project team to address requirements from multiple perspectives. It is recommended that QFD involve the customer, representatives from the product development and support functions, and suppliers. It is also recommended that a team attempting to conduct a QFD exercise for the first time receive training before the start of the QFD exercise and support from an experienced product-oriented QFD expert.

#### 4.6.5.2 Modeling and Simulation

Models and simulations are standard engineering tools that represent the key features of a system and the interactions of those features with each other and the outside environment. The defining feature of any model is its purpose. In general, a model represents how the system operates in its environment. An excellent guideline to follow is to select the least complex model that provides the most visibility into the problem.

### 4.6.6 Trade Studies Process Metrics

**Quality** may be measured by the degree to which the project objectives are satisfied, as noted in “Trade Studies Objectives” (Paragraph 4.6.1.1); objectives are satisfied when they may be numerically quantified (e.g., increase of payload capability). For imprecise objectives, project management may decide on a different type of assessment (e.g., yellow/red/green).

**Timeliness** may be measured by compliance with the schedule. It may be measured by when the decision support provided by the studies is available for the decision to be made.

**Resources** consumed to reach the required decision support level may identify efficiency, which may include labor hours, computer usage, and schedule time.

**Cycle time** may measure the duration from the creation of system alternatives to the delivery of the decision support products discussed in “Outputs of Trade Studies” (Paragraph 4.6.4).

**Process performance** is measured and recorded on a regular basis. Process users (teams or equivalent functions) accumulate the following metrics, at minimum, to evaluate the performance of this process:

- Percentage of studies performed in which none of the alternatives emerged conclusively as the best solution, thereby driving a decision based on other factors
- Percentage of studies in which the recommended alternative was not subsequently selected
- Percentage of planned discipline viewpoints, as defined by the study scope, that actively participated in conducting the Trade Studies

The decisionmaker completes satisfaction assessment.

### 4.6.7 References

1. Blanchard, B. "System Engineering Management." 2nd Ed. New York: John Wiley & Sons, Inc., N.Y.
2. Blanchard, B., and W. Fabrycky. "Systems Engineering and Analysis." 2nd Ed. Englewood Cliffs, New Jersey: Prentice Hall.
3. Defense Systems Management College. "Systems Engineering Fundamentals." Fort Belvoir, Virginia: Defense Systems Management College Press, 1999.
4. International Council on Systems Engineering. "Systems Engineering Handbook." Version 2.0. 2000.
5. National Aeronautics and Space Administration. "NASA System Engineering Handbook." June 1995.

# 4.7 Interface Management (Satisfies criteria of EIA/IS731 FA 1.5 and iCMM PA 7)

## 4.7.1 Introduction to Interface Management

Interface Management, which includes identification, definition, and control of interfaces, is an element of System Engineering (SE) that helps to ensure that all the pieces of the system work together to achieve the system's goals and continue to operate together as changes are made during the system's lifecycle. Precise interface definition early in the program is crucial to a successful and timely development. As the total system is decomposed into functional areas, interfaces (functional and/or physical) between the areas are identified. These interfaces are typically characterized by functional data parameters with associated data requirements or mechanical, electrical, and space requirements. Functional and physical interface requirements are contained in the appropriate performance specifications. The Interface Management process enters the Acquisition Management System (AMS) process at the end of the first phase of Investment Analysis and continues through In-Service Management. The essential elements of the Interface Management process are illustrated in Figure 4.7-1, which lists the key inputs necessary to initiate the task, providers, process tasks, outputs required, and customers of process outputs. The beginning and ending boundary task and the intermediate tasks are detailed later in the section.

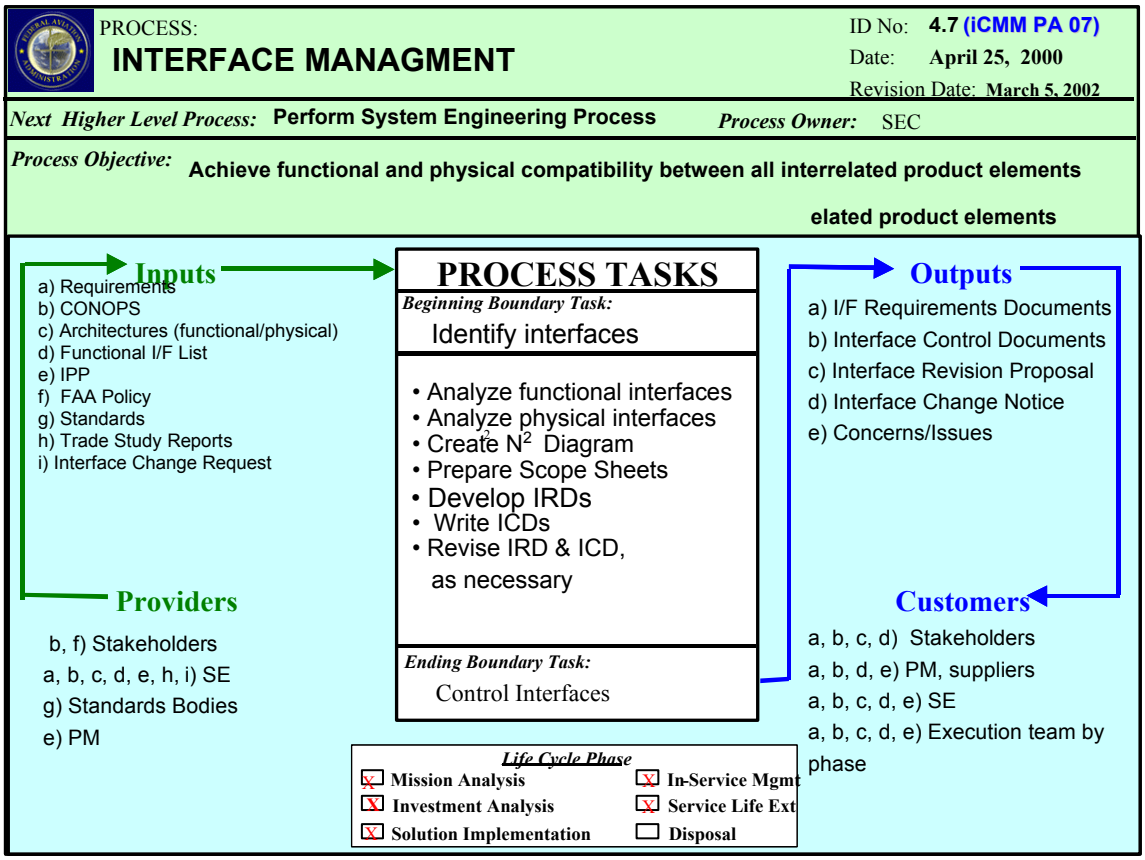



**Figure 4.7-1. Interface Management Process-Based Management Chart**

#### 4.7.1.1 Interface Management Objectives

The objective of Interface Management is to identify, describe, and define interface requirements to ensure compatibility between interrelated systems and between system elements, as well as provide an authoritative means of controlling the interface design.

The Interface Requirements Document (IRD) controls interface requirements, and the Interface Control Document (ICD) controls interface design. These documents:

- Define and illustrate physical and functional characteristics in sufficient detail to ensure compatibility of the interface so that this compatibility shall be determined from the information in the IRD/ICD alone
- Identify the necessary interface data and monitor submission of this data
- Control the interface requirements and design to prevent any changes to characteristics that might affect compatibility with other systems and equipment
- Communicate coordinated interface requirements/design decisions and interface requirements/design changes to program participants

#### 4.7.1.2 Types of Interfaces

An interface is any external or internal boundary between one element and another that is physical or functional. Internal interfaces are within the defined system's boundary. External interfaces are with elements outside the defined system's boundary. The external/internal interface distinction relates to the level of ownership and the verification of the requirements associated with each interface. Examples of interface types that may be encountered appear in Table 4.7-1. The 5M and SHELL Models (Figures 4.7-2 and 4.7-3, respectively) depict the types of interface elements that are recommended for consideration within most systems. Each element of the system shall be described functionally and physically. A functional description describes what the system is intended to do. It includes subsystem functions as they relate to and support the system function. (Functional Analysis (Section 4.4) provides more information on this topic.) A physical description provides information on the composition and organization of the tangible system elements. The level of detail varies with the system's size and complexity, with the end objective being adequate understanding of the system configuration and operation. (Synthesis (Section 4.5) provides more information on synthesis alternatives.)

**Table 4.7-1. Examples of Interface Types**

| Interface Type | Interface Subtype | Examples |
|---|---|---|
| Functional | Mechanical | Vehicle operator increasing speed<br>A computer sending a document to printer |
| Physical | Mechanical | Transmission of torque via a drive shaft<br>Connection between computer communication port and the printer cable |
| Functional | Control | A control signal sent from a flight control computer through a cable to an actuator (two interfaces)<br>A human operator selecting a flight management system |

| Interface Type | Interface Subtype | Examples |
|---|---|---|
| | | mode |
| Physical | Control | The connection between the flight control computer and the cabling<br>A human operator's fingers adjusting a flight management system mode switch |
| Functional | Aerodynamic | Pilot notification of a stall<br>Vortices impacting on an aircraft |
| Physical | Aerodynamic | A stall indicator on a wing<br>A fairing designed to prevent vortices from impacting a control surface on an aircraft |
| Functional | Environmental<br>(Natural or Induced) | Maximum/minimum temperature of radar electronics<br>The amount of rain/snow that makes a sensor reading anomalous |
| Physical | Environmental<br>(Natural or Induced) | Increased volume of mercury in thermometer reaching new markers on temperature scale<br>Wind impacting radar antenna surface |
| Functional | Noise | Minimum decibels required for an alert to be heard |
| Physical | Noise | Sound waves impacting on person's ear drum |
| Functional | Space | Space required to perform maintenance |
| Physical | Space | Inserting hardware into existing rack |
| Functional | Data | A cockpit visual display to a pilot<br>Weather Message Switching Center Replacement (WMSCR) to Weather and Radar Processor (WARP) data transfer |
| Physical | Data | Light from cockpit visual display impacting on pilot's retina<br>Weather data bits moving from communications cable to communications port on WARP |
| Functional | Electrical | Energy from a direct current (DC) power bus supplied to an anticollision light<br>A fan plugged into an alternating current (AC) outlet for current<br>An electrical circuit opening a solenoid<br>Shielding and grounding for coaxial cables |
| Physical | Electrical | Energy from a DC power bus supplied to the cabling connected to the anticollision light<br>Electrical current moving from AC outlet to fan wire<br>Current flowing through wiring<br>Shielding material wrapped around copper wiring |

| Interface Type | Interface Subtype | Examples |
|---|---|---|
| Functional | Hydraulic | Pressurized fluid supplying power to a flight control actuator<br>A fuel system pulling fuel from a tank to the engine |
| Physical | Hydraulic | Pressurized fluid in a hydraulic line<br>Connection of fuel line to fuel tank |
| Functional | Pneumatic | An adiabatic expansion cooling unit supplying cold air to an avionics bay<br>An air compressor supplying pressurized air to an engine air turbine starter |
| Physical | Pneumatic | Pressurized air in an aircraft |
| Functional | Electromagnetic | Radio frequency (RF) signals from a Very High Frequency Omni directional Range (VOR)<br>A radar transmission |
| Physical | Electromagnetic | RF signals from a VOR vibrating radio receiver<br>Radio waves emitted from radio transmitter |
| Functional | Heating, Ventilating, and Air-Conditioning (HVAC) | Amount of heating and cooling required for a facility<br>Circuit protective devices for equipment racks |
| Physical | HVAC | Thermocouple contacting sensor<br>Circuit breaker connection to power line |

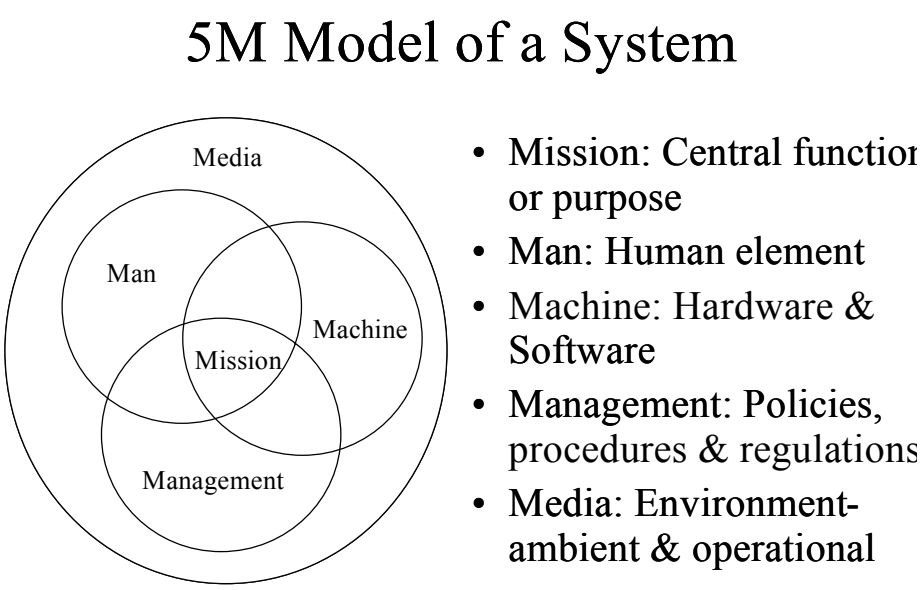



**Figure 4.7-2. Depiction of 5M Interface Model**

The following is a description of the 5M Interface Model:

- **Mission:** the purpose or central function of the system that brings together the other elements.
- **Man:** the human element of a system. If a system requires humans for operation, maintenance, or installation, this element shall be considered in the system description.
- **Machine:** the hardware and software (including firmware) element of a system.
- **Management:** the procedures, policy, and regulations involved in operating, maintaining, installing, and decommissioning a system.
- **Media:** the environment in which a system shall be operated, maintained, and installed. This environment includes ambient and operational conditions. Ambient conditions are physical conditions involving temperature, humidity, lightning, electromagnetic effects, radiation, precipitation, and vibration. Operational environment consists of the conditions in which the mission or function is planned and carried out. Operational conditions are human-created conditions involving operations such as air traffic density, communication congestion, workload, and Instrument Flight Rules (IFR) versus Visual Flight Rules (VFR). Part of the operational environment may be described by the type of operation (e.g., air traffic control, air carrier, general aviation); phase (e.g., ground taxiing, takeoff, approach, en route, transoceanic, landing); or rules governing the operation (e.g., IFR, VFR).

In the SHELL Model, the match or mismatch of the blocks (interface) is just as important as the characteristics described by the blocks themselves. These blocks may be rearranged to describe the system as required. A connection between two blocks indicates an interface between the elements.

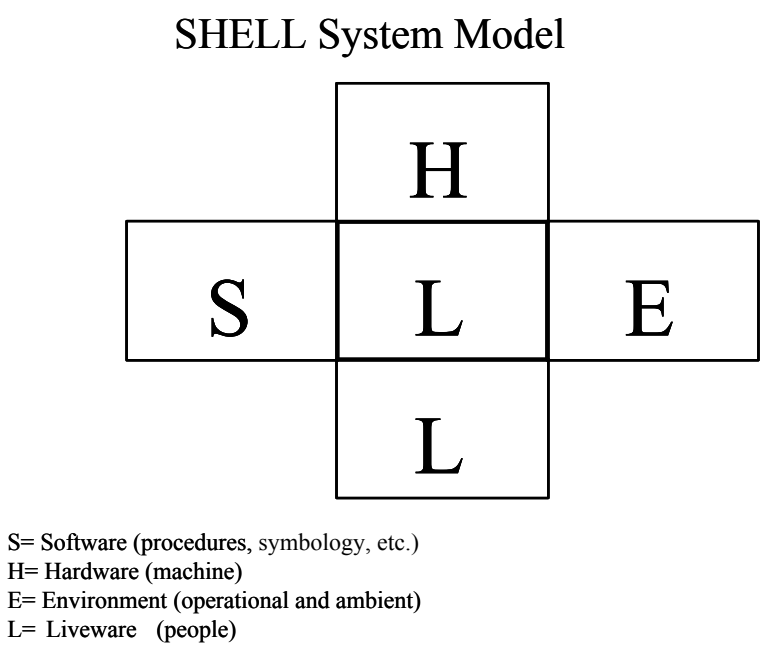



**Figure 4.7-3. Depiction of SHELL Interface Model**

#### 4.7.1.2.1 Functional Interfaces

Functional interfaces define the purpose of the interface. Each interface has at least two associated functions, and, because all performance requirements are traceable to functions, there shall be at least two associated interface requirements. This concept is illustrated in Figure 4.7-4, where Side A delivers some quantity (e.g., electrical power) to Side B; at the same time, Side B receives that quantity from Side A. The two implied requirements are:

- Side A shall generate the quantity
- Side B shall provide a compatible response to the quantity that Side A delivered

Interface requirements shall be expressed in verifiable terms. For example, as expressed in strict requirements terminology, "the [Side A] subsystem shall deliver electrical power at 28 volts." In this example, the element of Side B is a fan. Thus, the requirement for Side B might be as follows: "The fan [Side B] shall provide impedance, power level and timeline, while using the 28-volt power supply of the electrical system [Side A]." The interface definition includes the data and/or control functions and the way these functions are represented.

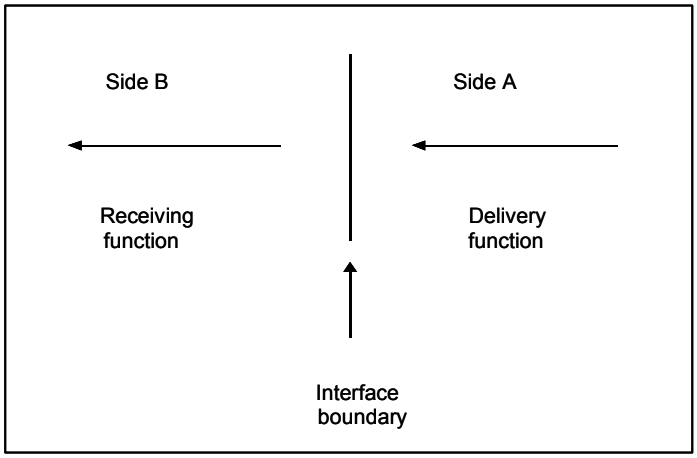



**Figure 4.7-4 Example of a Simple Interface**

#### 4.7.1.2.2 Physical Interfaces

Physical interfaces are used to define and control the features, characteristics, dimensions, and tolerances of one design that affects another. Physical interfaces include material properties of the equipment that affect the functioning of mating equipment. They also include the operating environment of the system.

### 4.7.2 Inputs to Interface Management

The inputs required to initiate Interface Management include both program/project- and product-related data listed in Table 4.7-2. Many of these inputs are developed and refined through the continuous, iterative processes of other SE elements.

**Table 4.7-2. Interface Management Process Inputs**

| Input | Reference |
|---|---|
| CONOPS | Functional Analysis (Section 4.4) |
| Architecture | Trade Studies (Section 4.6) |
| Requirements MNS/iRD | Requirements Management (Section 4.3) |
| International Standards | System Engineering in the Acquisition Management System Program Lifecycle (Chapter 3) |
| FAA Order/Standards | System Engineering in the Acquisition Management System Program Lifecycle (Chapter 3) |
| Functional Analysis | Functional Analysis (Section 4.4) |
| Draft IPP | Integrated Technical Planning (Section 4.2) |
| Trade Study Report | Synthesis (Section 4.5) |
| Engineering solution actions and impacts | Trade Studies (Section 4.6) |
| Interface Control Plan | Integrated Technical Planning (Section 4.2) |
| Interface Change Request | Interface Management (Section 4.7) |

### 4.7.3 Interface Management Process Tasks

The Interface Management process is an integrated and iterative set of activities that ensures that all functional and physical interface requirements are identified, defined, and controlled, including interfaces within the system, as well as those between the instant system and another system. Table 4.7-3 outlines the process. The paragraphs below describe the process tasks.

**Table 4.7-3. Interface Management Process Inputs by Output Product**

| Inputs | Source Process | Initial AMS Phase | Output |
|---|---|---|---|
| Requirements Documents (MNS/iRD) | Requirements Management (Section 4.3) | Mission Analysis (MA) | |
| CONOPS | Functional Analysis (Section 4.4) | MA | |
| Architecture | Synthesis (Section 4.5) | MA | |
| Functional Interface List | Functional Analysis (Section 4.4) | MA | |
| | | | Scope Sheet |
| FAA Policy | External | Investment Analysis (IA) | |
| Standards | External | IA | |

| Inputs | Source Process | Initial AMS Phase | Output |
|---|---|---|---|
| Draft Interface Control Planning section of IPP | Integrated Technical Planning (Section 4.2) | IA | |
| Requirements Documents (fRD)/Changes | Requirements Management (Section 4.3) | IA | |
| System Requirements/Changes | Functional Analysis (Section 4.4)<br>Synthesis (Section 4.5)<br>Trade Studies (Section 4.6) | IA | |
| Physical Architecture | Synthesis (Section 4.5) | IA | |
| Trade Study Report | Trade Studies (Section 4.6) | IA | |
| | | | IRD |
| IRD | | Solution Implementation (SI) | |
| Interface Change Request | External | SI | |
| Physical Architecture | Synthesis (Section 4.5) | SI | |
| Design Definition/Changes | Synthesis (Section 4.5) | SI | |
| Final Interface Control Planning section of IPP | Integrated Technical Planning (Section 4.2) | SI | |
| | | | ICD |
| Interface Revision Proposal | | | |
| | | | Revised IRD/ICD |

#### 4.7.3.1 Task 1: Identify Functional/Physical Interfaces

The first task in the Interface Management process is to identify the functional and physical interfaces, which is accomplished via $N^2$ diagrams. The functional interfaces are identified during the Mission Analysis phase, while the physical interfaces are identified during the Investment Analysis phase.

#### 4.7.3.2 Task 2: Create an $N^2$ Diagram

The $N^2$ diagram is a systematic approach to identify, define, tabulate, design, and analyze functional and physical interfaces. It applies to system interfaces and hardware and/or software interfaces. The $N^2$ diagram is a visual matrix that requires the user to generate complete definitions of all the system interfaces in a rigid bidirectional, fixed framework. Figure 4.7-5 is a basic $N^2$ diagram.

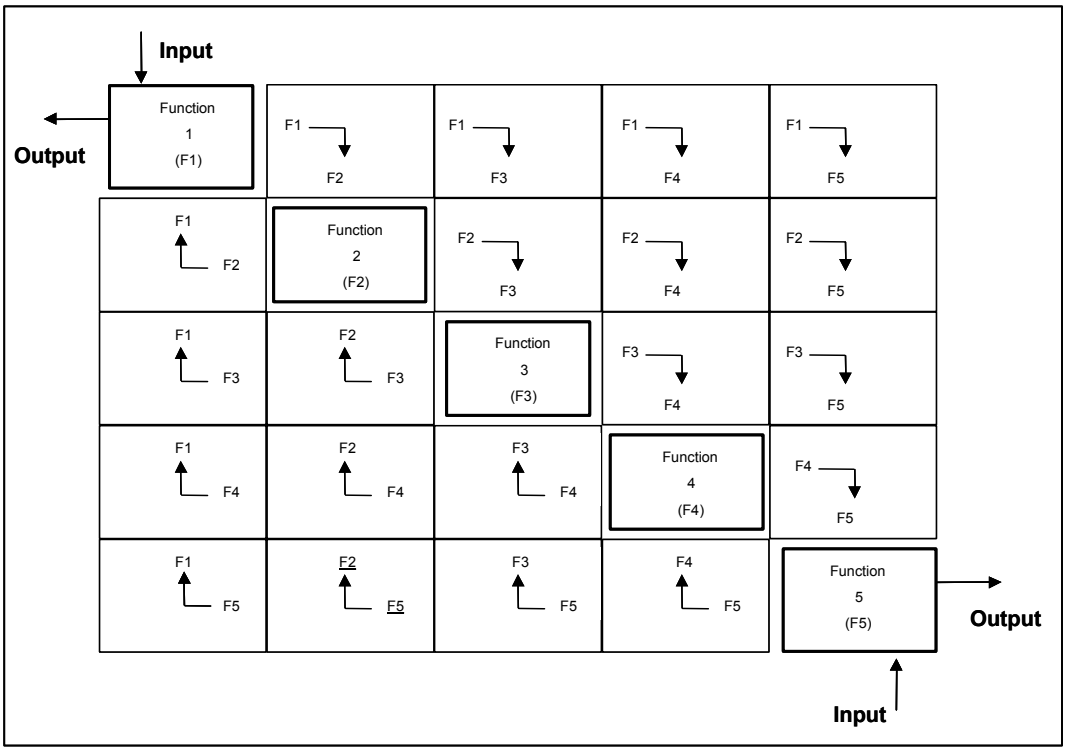



**Figure 4.7-5. Generic $N^2$ Diagram**

The following steps are recommended for creating a functional $N^2$ diagram:

Step 1: Identify the functional interfaces via an $N^2$ chart and develop functional interface list.

- Create an $N^2$ diagram that is N X N square, where N is the number of system functions.
- Place the system functions on the diagram's diagonal axis.
- Moving across the diagram, fill in each square with any output, moving from function F1 to any of the succeeding functions. (Interfaces between functions flow in a clockwise direction.) If there are no outputs to a succeeding function, leave the square blank. (Characteristics of the entity passing between functions may be included in the box where the entity is identified.) Continue in this fashion until the upper half of the $N^2$ diagram is populated.
- Moving down the diagram, fill in each square with any input, moving from function F2 to function F1, from function F3 to functions F2 or F1, and so on with succeeding functions. If there are no outputs to a succeeding function, leave the square blank. Continue in this fashion until the lower half of the $N^2$ diagram is populated.
- Conduct a peer review for completeness.

Step 2: Develop a functional interface list from the functional $N^2$ diagram.

The next action is to identify the physical interfaces via the $N^2$ diagram during the Investment Analysis phase using the selected Physical Architecture.

Step 3: Identify the physical interfaces via an $N^2$ chart and develop physical interface list.

- Create an $N^2$ diagram that is N X N square, where N is the number of system elements.
- Place the system elements on the diagram's diagonal axis.
- Moving across the diagram, fill in each square with any output, moving from system S1 to any of the succeeding systems. (Interfaces between systems flow in a clockwise direction.) If there are no outputs to a succeeding system, leave the square blank. (Characteristics of the entity passing between systems may be included in the box where the entity is identified.) Continue in this fashion until the upper half of the $N^2$ diagram is populated.
- Moving down the diagram, fill in each square with any input, moving from system 1 to system 2, from system 3 to systems 2 or 1, and so on with succeeding systems. If there are no outputs to a succeeding system, leave the square blank. Continue in this fashion until the lower half of the $N^2$ diagram is populated.
- Conduct a peer review for completeness.

Step 4: Develop a Physical I/F list from the Physical $N^2$ chart.

An example of an output from Step 3 appears in Figure 4.7-6. The $N^2$ diagram shall be taken down in successively lower levels to the hardware and software component levels. In addition to interface identification, another main function of the $N^2$ diagram is to pinpoint areas where conflicts may arise between systems and functions so that system integration occurring later in the development cycle proceeds efficiently.

| | Provide electrical power | Provide environmental control | Provide surveillance |
|---|---|---|---|
| Provide electrical power | | Provide power to ESC | Provide power to radar |
| Provide environmental control | Cool electrical components | | Cool radar elements |
| Provide surveillance | | | |

**Figure 4.7-6. Simple $N^2$ Example**

#### 4.7.3.3 Task 3: Define Functional and Physical Interfaces To Prepare Scope Sheets

The third task in the Interface Management process is to define the functional and physical interfaces, which is accomplished via scope sheets and IRDs. . Scope sheets are used to develop the Interface Control planning section of the Integrated Program Plan (IPP) (Integrated Technical Planning (Section 4.2)). This Interface Control planning section defines a management system of interface controls to ensure physical and functional compatibility between interfacing system elements and between systems. This section also provides the means to identify and resolve interface incompatibilities (through a program management mechanism known as the Interface Control Working Group (ICWG)) and determines the impact of interface design changes. Source material for the Interface Control planning section includes the draft IPP and the Integrated Master Schedule. The previously developed $N^2$ diagrams are used to complete a scope sheet for each interface, which, in turn, is used to write the required IRDs.

The following steps shall be taken when scope sheets are prepared:

- Step 1: Review scope sheet format (Figures 4.7-7 and 4.7-8)
- Step 2: Review functional and physical I/F lists
- Step 3: Prepare a scope sheet for each element in the diagonal, which corresponds to internal interfaces
- Step 4: Review final Requirements Documents (fRD) to determine required external interfaces
- Step 5: Prepare scope sheets for all external interfaces
- Step 6: Enter scope sheets into Configuration Management process (Section 4.11)
- Step 7: Evaluate Scope Change Requests and update scope sheets as necessary

| | | | |
|---|---|---|---|
| ICD NUMBER:<br>REV: | DATE INITIATED:<br>DATE: | | |
| ICD TITLE | | | |
| PARTICIPANTS: | | | |
| SCOPE: | | | |
| EQUIPMENT RESPONSIBILITY: | | | |
| **INTERFACE LOCATION (INTERFACE BLOCK DIAGRAM)** | | | |
| EFFECTIVITY: | | | |
| PROGRAM REVIEWS & AUDITS: | | | |
| RELATED ICDs | | | |
| APPROVALS: | | | |
| | | | |
| Participant | Date | Participant | Date |
| | | | |
| ICWG Secretariat | Date | ICWG Chairman | Date |

**Figure 4.7-7. Format of Scope Sheet for Interface Management**

| | |
|---|---|
| ICD NUMBER: 25-DR010M<br><br>REV: 1 | DATE INITIATED: June 25, 3032<br><br>DATE: December 6, 3033 |
| ICD TITLE | Interface Control – Surveillance Radar Product Generator (RPG) – Weather System Processor (WSP) - Electrical Installation Envelope, Mechanical, Environmental, and Data |
| PARTICIPANTS: | Raytheon/Lockheed Martin |
| SCOPE: | This IRD/ICD controls and documents all interface requirements for the RPG to WSP interface. Interface definition is described to the extent necessary to assure compatibility of the RPG to WSP interfacing hardware when used with the specified constraints. The interface consists of mechanical installation of the WSP for cabling, mounting, environmental cooling, and data requirements. Mechanical interfaces include location, orientation, mounting provisions, and power supply. Envelope interfaces include installation, removal, connector, and cable clearances. Environmental interfaces include temperature and humidity constraints. The data interface includes Airport Surveillance Radar (ASR) 27 data (RF, control, data, and timing signals) and WSP data (control and status signals). |
| EQUIPMENT RESPONSIBILITY: | 1. Raytheon – ASR-27 radar product generator<br><br>2. Lockheed Martin – WSP module (hardware and software) |
| **INTERFACE LOCATION (INTERFACE BLOCK DIAGRAM)** | |
| EFFECTIVITY: | PK/RG (TYPE VIII) |
| PROGRAM REVIEWS & AUDITS: | WSCE IRR September 3032, WSCE SER December 3032, WSCE PDR March 3033 |
| RELATED ICDs | |
| APPROVALS: | |

| | | | |
|---|---|---|---|
| | | | |
| Raytheon | Date | Lockheed Martin | Date |
| | | | |
| ICWG Secretariat | Date | ICWG Chairman | Date |

**Figure 4.7-8. Example Scope Sheet**

#### 4.7.3.4 Task 4: Develop Interface Requirements Documents

The next task in the Interface Management process is to develop IRDs, which, in turn, are used to develop ICDs. The designated custodian shall prepare the detailed IRD. FAA-STD-025 provides a checklist for IRD content. Several commonly used FAA standards appear in Table 4.7-4.

The following steps shall be undertaken when IRDs are developed:

- Step 1: Review the inputs listed in Table 4.7-2
- Step 2: Prepare the detailed IRD in accordance with (IAW) FAA-STD-025
- Step 3: Review the IRD for compliance with the fRD
- Step 4: Coordinate the revised draft IRD with all affected organizations
- Step 5: Enter the IRD into the Configuration Management process (Section 4.11)

**Table 4.7-4. Checklist for Interface Requirements Document Standards (In Accordance With FAA-STD-025)**

| Standard | Title |
|---|---|
| FAA-STD-025 | Preparation of Interface Documentation |
| FAA-STD-002 | Facilities Engineering Drawing Preparation |
| FAA-STD-005 | Preparation of Specification Documents |
| FAA-STD-019 | Lighting Protecting, Grounding, Bonding, and Shielding Requirements for Facilities |
| FAA-STD-020 | Transient Lighting Protecting, Grounding, Bonding, and Shielding Requirements for Equipment |
| FAA-STD-023 | Microfilming of Engineering and Electrical Drawings |
| FAA-STD-029 | Selection of Telecommunications Standards |
| FAA-STD-032 | Design Standards for National Airspace System (NAS) Physical Facilities |
| FAA-STD-039 | NAS Open Systems Architecture and Protocols |
| FAA-STD-042 | NAS Open System Interconnection (OSI) Naming and Addressing |
| FAA-STD-043 | NAS OSI Priority |

| Standard | Title |
|---|---|
| FAA-STD-044 | NAS OSI Directory Services |
| FAA-STD-045 | NAS OSI Security Standard |
| FAA-STD-047 | NAS OSI Conformance Testing |
| FAA-STD-048 | NAS OSI Interoperability Testing |
| FAA-STD-049 | FAA Standard for Fiber Optic Telecommunications Systems and Equipment |
| | |
| MIL-STD-005 | Engineering Drawing Practices |
| | |
| ISO 8648-1988 | Information Processing Systems – OSI Internal Organization of the Network |
| ISO/IEC 96467:1998 | Information Technology – OSI – Conformance Testing Methodology and Framework: Implementation Conformance Statements |
| ISO/IEC TR 1000-1-1998 | Information Technology – Framework and Taxonomy of International Standardization Profiles – Part 1: General Principles and Documentation Framework |
| IEEE 315 – 1975 | Graphic Symbols for Electric and Electronics Diagrams (including reference class designations letters) |
| IEEE 315A – 1986 | Graphic Symbols for Electric and Electronics Diagrams (supplement to Institute of Electrical and Electronics Engineers, Inc. (IEEE) and standard 315-1975) |

#### 4.7.3.5 Task 5: Write Interface Control Documents

During this task, the detailed ICD/Interface Control Notice (ICN) is prepared, and an analysis is performed to confirm completeness and accuracy of the interface definition. These documents shall be reviewed for compliance with the defined scope sheets and coordinated. A record of these actions shall be maintained.

FAA-STD-025 provides a checklist for ICD content.

- Step 1: Review the inputs listed in Table 4.7-2
- Step 2: Prepare the detailed ICD IAW FAA-STD-025
- Step 3: Review the ICD for compliance with IRD
- Step 4: Coordinate the revised draft ICD with all affected organizations
- Step 5: Enter the ICD into the Configuration Management process (Section 4.11)

#### 4.7.3.6 Task 6: Revise Interface Requirements Documents and Interface Control Documents

It may be necessary to request changes to the ICD as changes to Requirements or design definition occur.

- Step 1: Review the interface document for changes when design modifications occur or new requirements are added.
- Step 2: Review the IRD/ICD to determine if changes are also required.
- Step 3: Prepare the change request IAW FAA-STD-025 and provide the following information:
  - Description of the problem and the proposed change
  - Analysis showing how the change solves the problem
  - Analysis of how the change impacts system performance, effectiveness, and lifecycle costs
  - Analysis to ensure that the proposed solution does not introduce new problems
  - Description of resources and an estimate of the costs associated with implementing the change
  - Statement of impact to system
- Step 4: Provide change request to ICWG, which shall determine if the authorized Interface Change Request (ICR) is within the scope. In-scope ICRs shall be returned to the ICR originator and the custodian of the ICD for preparation and release of an interface requirement. Out-of-scope ICRs shall be forwarded to program manager.
- Step 5: Coordinate the draft IRD/ICD with all affected organizations.
- Step 6: Enter the changed IRD/ICD into the Configuration Management process (Section 4.11).

### 4.7.4 Outputs of Interface Management

The outputs of the Interface Management process appear in Table 4.7-5. When documented and approved, the IRD is provided to all applicable organizations, while the ICD is provided to technical disciplines responsible for meeting its interface requirements, to customer and program management for coordination, and to the respective test and quality assurances organizations.

**Table 4.7-5. Process Outputs and Destination Process**

| Outputs | Destination Process(es) |
|---|---|
| IRDs | Requirements Management (Section 4.3)<br>Configuration Management (Section 4.11)<br>Synthesis (Section 4.5)<br>Validation and Verification (Section 4.12) |
| ICDs | Requirements Management (Section 4.3)<br>Configuration Management (Section 4.11)<br>Synthesis (Section 4.5)<br>Validation and Verification (Section 4.12) |
| Interface Change Proposal (ICP) | Configuration Management (Section 4.11) |

### 4.7.5 Interface Management Tools

The functional flow diagram (FFD):

- The FFD family is a group of analyses that depicts functional (input-function-output) relationships between functions. This family includes the Department of Defense (DoD) standard FFDs, $N^2$ diagrams, Integrated Definition for Function Modeling (IDEF) tools, and the Unified Modeling Language (UML). The FFD is a multi-tier, time-sequenced, step-by-step diagram of the system's functional flow. Typically, FFDs are prepared to define the detailed, step-by-step, operational and support sequences for systems, but they may also be used effectively to define processes in developing and producing systems. In this method, the functions are organized and depicted by their logical inputs and outputs. Each function is shown in relation to the other functions by how the inputs and outputs feed and are fed by the other functions. Each function is depicted as a node labeled with the function name. (Naming criteria are described in "Introduction to Functional Analysis" (Paragraph 4.4.1).) Arrows leading into the function depict inputs, while arrows leading out of the function depict outputs. If the output of function F0 is an input to F1, then an arrow is shown leaving F0 and going into F1 ("Functional Flow Relationship" (Figure 4.4-12)).

The $N^2$ Diagram

- The $N^2$ diagram (Figure 4.7-5) ensures that all functions identified in the Functional Analyses are reflected in functional interfaces. Each node in the $N^2$ diagram indicates a possible functional interface. Note that in the example (Figure 4.7-6), "Provide Electrical Power" supplies power to other functions, and "Provide Environmental Control" supplies cooling to the other functions. However, the "Provide Surveillance" function does not provide any quantity to the other functions.

### 4.7.6 Interface Management Process Metrics

Table 4.7-6 lists the Interface Management process metrics.

**Table 4.7-6. Interface Management Process Metrics**

| Quality Metrics | Cycle Time Metrics | Cost* Metrics |
|---|---|---|
| Scope Sheet in Compliance with Requirements (% “Yes”) | Time from iRD to IRD Approval | Cost to implement IRDs |
| IRD in Compliance with Requirements (% "Yes") | Time from IRD Approval to ICD Release | Cost to implement ICDs |
| ICD/Interface Requirement Compliance with Interface Requirements (% "Yes") | Time from ICR Approval to Interface Requirement Release | Cost to implement ICRs |
| Design Compliance with ICD/Interface Requirement Requirements (% "Yes") | | |
| Number of interfaces discovered after initial release of ICD | | |
| *NOTE: Cost is only direct program costs. | | |

### 4.7.7 Terms and Definitions

**ICD:** The ICD is one of the two basic products of the interface task. In its final form, the ICD is a clear, more detailed documentation of the interface requirements. It shows clearly which items belong to each side of the interface. It shows the exact value of the interface functions and their tolerances. The ICD is **not** used to design any part of the system. That information resides in the product definition (part model or drawing set) of each component involved in the interface. The requirements associated with the interface functions reside in the performance sections of the specifications. Hence, the ICD serves as a good coordination vehicle.

**Interface Requirements:** All interface requirements are classed as functional and physical requirements, as well as constraints that exist at a common boundary between two or more functions, system elements, configuration items, or systems.

**IRD:** The IRD defines requirements associated with external physical and functional interfaces between the particular system and other associated system(s). The IRD is one of the two basic products of the interface task. In its final form, the IRD is primary documentation of the interface requirements.

**Interface Control Planning Section of IPP:** The Interface Control planning section of the IPP documents the formal management system of interface controls that ensures physical and functional compatibility between interfacing hardware, software, and facilities. The plan provides the means for identifying and resolving interface incompatibilities and for determining the impact of interface design changes. This Interface Control planning guides the management, control, and documentation of all system functional and physical interfaces. The Interface Control planning section also contains interface requirements and templates for preparing, revising, and processing ICDs unique to the program. The Interface Control planning

section addresses supplier participation in the interface process. (Integrated Technical Planning (Section 4.2) provides detailed instructions on this topic.)

**ICWG:** The ICWG is established through the IPP (and System Engineering Management Plan (SEMP)). The ICWG is the forum for discussing interface issues. ICWG meetings serve two purposes: to ensure effective, detailed definition of interfaces by all cognizant parties, and to expedite baselining of initial IRDs, ICDs, and subsequent drawing changes by encouraging resolution of interface issues. The ICWG shall consist of ICWG Chair, IRD/ICD Custodian(s), and management personnel from associated teams. (Integrated Technical Planning (Section 4.2) provides detailed instructions on this topic.)

### 4.7.8 References

1. Blanchard, Benjamin S., System Engineer Management, John Wiley & Son, 603 Third Ave., New York, N.Y., 10158-0012, 1998.
2. Buede, Dennis M., The Engineering Design of Systems, John Wiley & Son, 603 Third Ave., New York, N.Y., 10158-0012, 2000.
3. EIA 731.1, Systems Engineering Capability Model, January 1999.
4. Grady, Jeffery O., Systems Requirements Analysis, McGraw-Hill, March 1999.
5. NASA Systems Engineering Handbook, SP-6105, June 1995
6. System Engineering Fundamentals, DSMC, Defense Acquisition University Press, Ft. Belvoir, VA, 1999.

## 4.8 Specialty Engineering

Specialty Engineering is a subset of System Engineering (SE) that defines and evaluates specific areas, features, and/or characteristics of a system. Specialty Engineering supplements the acquisition process by defining these characteristics and assessing their impact on the program. SE relies on specialty domain expertise to define and characterize specific requirements. SE's function in this process is to integrate the design engineer and specialty engineer's activities, coordinate and open communication lines between the design engineer and specialty engineer, and focus the engineering effort toward the common goal of satisfying the customer—not to perform detailed Specialty Engineering work.

Analysis of the system is a primary means of conducting Specialty Engineering. These analyses are categorized under Specialty Engineering because they require specialized engineering skills. These specialized skill areas include system safety engineering (SSE); Reliability, Maintainability, and Availability (RMA); Human Engineering (human factors); Electromagnetic Environmental Effects ($E^3$); quality Engineering; Information Security Engineering; and Hazardous Materials Management/Environmental Engineering. Engineers in these disciplines perform analyses throughout the system's lifecycle. The results are used to derive, validate, and verify requirements; evaluate system design progress and technical soundness; and manage risk. At a minimum, analysis results are available at standard design milestones, including the design, acquisition, and program reviews. The results are communicated via reports. In the case of supplier involvement, deliverables are in accordance with contract requirements. The general process for performing Specialty Engineering is depicted in Figure 4.8-1, which lists the key inputs necessary to initiate the task, providers, process tasks, outputs required, and customers of process outputs.

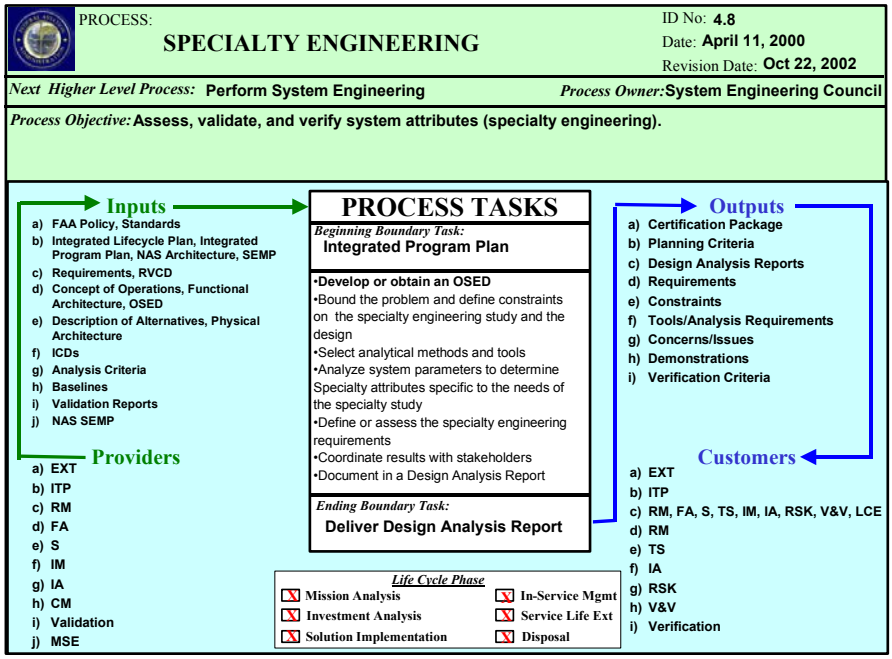



**Figure 4.8-1. Specialty Engineering Process-Based Management Chart**

## 4.8.0 Introductory Material

### 4.8.0.1 Introduction to Specialty Engineering

Specialty Engineering is conducted throughout the system's lifecycle. Specialty Engineering analyses are conducted early to derive and validate requirements. In addition, the Specialty Engineering disciplines support the Trade Studies (Section 4.6), Synthesis (Section 4.5), and Functional Analysis (Section 4.4) efforts in selecting and designing solutions to requirements. Later in the lifecycle, after requirements at all levels are validated, these analyses provide support in verifying the requirements by describing and assessing the characteristics of the design and/or operations. As early as possible in the lifecycle, the Specialty Engineering disciplines find and resolve potential program risk. Finding and controlling risk early assists in seeking the lowest possible cost and increases the probability of program success and operator acceptance of the product.

This section contains a description of the functions, objectives, and products of the various disciplines included in Specialty Engineering.

#### 4.8.0.1.1 Description of Specialty Engineering Disciplines

Specialty Engineering analyses provide characteristics of the system from a specific technical perspective. Table 4.8-1 provides a general description of the Specialty Engineering disciplines.

**Table 4.8-1. Specialty Engineering Disciplines**

| Specialty Engineering Discipline | Description |
|---|---|
| SSE | Evaluation and management of the safety risk associated with a system using measures of safety risk identified in various hazard analyses, fault tree analyses, safety risk assessments, and hazard tracking and control. |
| RMA | Quantitative and qualitative analyses of the attributes of the system to perform reliably. Quantitative assessments are in the form of probabilistic, mean, and/or distribution assessments. Qualitative analyses are in the form of failure mode assessments. Evaluation of the design's ability to meet operational readiness requirements through preventive and corrective maintenance. |
| Human Factors Engineering | Human factors is a multidisciplinary effort to generate and compile information about human capabilities and limitations and apply that information to:<br>– equipment, systems, facilities<br>– procedures, jobs, environments<br>– staffing<br>– training |

| Specialty Engineering Discipline | Description |
|---|---|
| | – personnel and organizational management<br><br>for safe, comfortable, and effective human performance. |
| $E^3$ | Analysis of the system for susceptibility and/or vulnerability to electromagnetic fields or capability to generate such fields that might interfere with other systems, identify sources of interference, and means for correction within the levels prescribed by law, program requirements, spectrum management, or recognized standards.<br>$E^3$ is composed of Electromagnetic Interference (EMI) and Electromagnetic Compatibility (EMC) |
| Quality Engineering | Evaluation of a system's ability to meet its requirements and to mitigate product defects. |
| Information Security Engineering | Evaluation of the vulnerability of the system to unauthorized access and use, or susceptibility to sabotage. Assessment of the ability of the system to survive a security threat in the expected operational environment. |
| Hazardous Materials Management/Environmental Engineering | Determination of environmental impacts at deployment sites and during operations, including both environmental impacts on the system and system impacts on the environment during all phases of the product life. |

In addition to resolving problems and defining requirements early, Specialty Engineering supplies information to the other SE functions, including Requirements Management (Section 4.3), Risk Management (Section 4.10), Configuration Management (Section 4.11), and Validation and Verification (Section 4.12). The major relationships between Specialty Engineering and other SE processes are depicted in Figure 4.8-2.

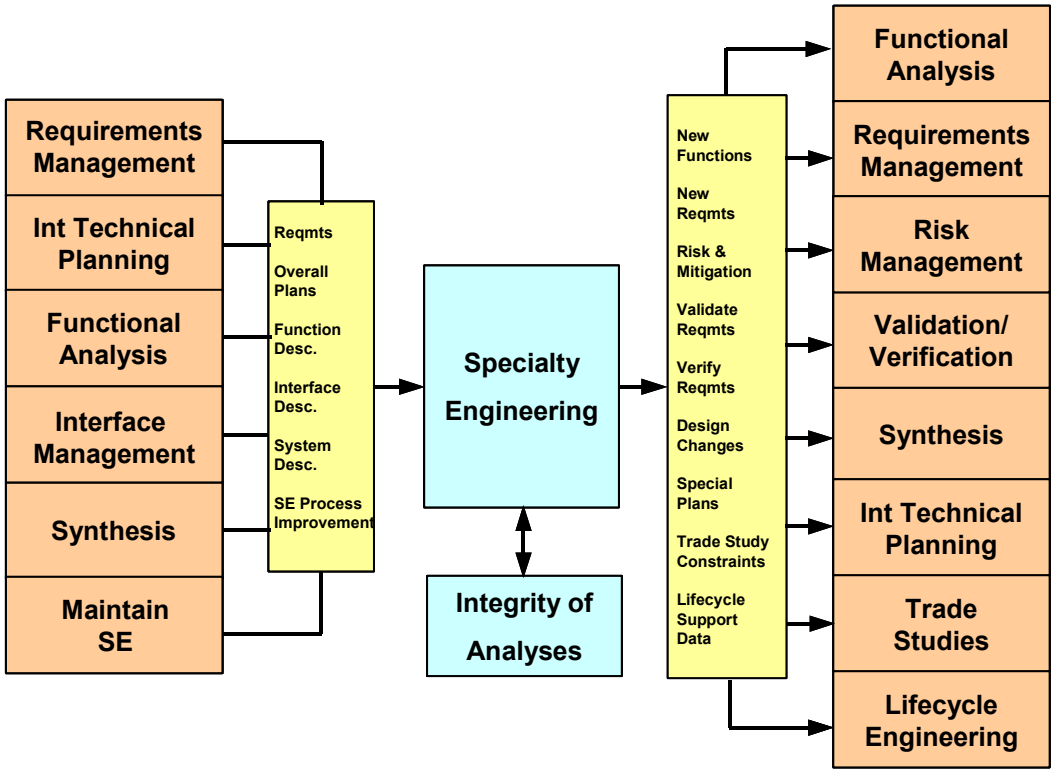



**Figure 4.8-2. Major relationships Between System Engineering Elements and Specialty Engineering**

The relationships depicted in Figure 4.8-2 are further described in Table 4.8-2.

**Table 4.8-2. Major Effects of Specialty Engineering on Other System Engineering Processes**

| Affected SE Process | How Affected |
|---|---|
| Integrated Technical Planning (Section 4.2) | The Integrated Technical Planning process feeds Specialty Engineering. Integrated Technical Planning produces the plans for Specialty Engineering, SE, and all other SE processes. The plans detail what is to be done, who is to do it, the standards of performance, and when each task is to be performed. |
| Requirements Management (Section 4.3) | The Requirements Management process both feeds and is fed by Specialty Engineering. The system under study is described in order to perform Specialty Engineering analyses. Requirements are a key component of any description and they are an output of the Requirements Management process. Specialty Engineering studies often find characteristics that create a need for new or different requirements. Sometimes, the Specialty Engineering disciplines find areas of conflict between two or more requirements. In either case, the Specialty Engineering function develops the new or changed requirements and these are an input to the Requirements Management process. |
| Functional Analysis (Section 4.4) | The Functional Analysis process both feeds and is fed by Specialty Engineering. To execute a Specialty Engineering |

| Affected SE Process | How Affected |
|---|---|
| | analysis, the specialist shall have a thorough understanding of the system functions. This understanding is a result of performing a Functional Analysis of the system. |
| Interface Management (Section 4.7) | Specialty Engineering both feeds and is fed by Interface Management. The system under study is described in order to perform Specialty Engineering analyses. Interface Requirements Descriptions (IRD) are key components of any system description and are an output of the Interface Management process. Specialty Engineering studies often find characteristics that create a need for new or different interface requirements or descriptions. Sometimes, the Specialty Engineering disciplines find areas of conflict between two or more interfaces. In either case, the Specialty Engineering function develops the new or changed requirements, which are inputs to the Interface Management process. |
| Risk Management (Section 4.10) | Specialty Engineering feeds the Risk Management process. Specialty Engineering studies and analyses are designed to find and assess potential problem areas of a design as early as possible. When a potential problem is found, the information becomes an input to the Risk Management process for mitigation and control. |
| Configuration Management (Section 4.11) | Specialty Engineering outputs are inputs to the Configuration Management process. During the execution of Specialty Engineering analyses, it may be discovered that additional or changed design features are required, or changes to operating, maintenance, or installation procedures are needed. When these discoveries occur, the proposed changes become inputs to the Configuration Management process. |
| Validation and Verification (Section 4.12) | Specialty Engineering outputs feed the Validation and Verification process. Early in the program's lifecycle, Specialty Engineering is used to validate requirements, which is accomplished by comparing the requirements defined in early Specialty Engineering analyses to those defined in current/later analyses. If the Specialty Engineering analyses find a need for an existing requirement, then the requirement may be considered validated.<br><br>Specialty Engineering feeds Verification Criteria to the Verification process. Specialty Engineering is also used to verify requirements later in the system's lifecycle. Verification may be accomplished either by test or SE Assessment. Specialty Engineering is a form of assessment and may be used to demonstrate verification. |

#### 4.8.0.2 Inputs and Providers to Specialty Engineering

Table 4.8-3 depicts the inputs needed to conduct Specialty Engineering analyses.

58 **Table 4.8-3. Specialty Engineering Process Inputs**

| Process Input | Input Purpose/Description | From Process |
|---|---|---|
| FAA Policy and Standards | Policy and standards, such as the Acquisition Management System (AMS), define what is expected and how well it needs to be done. | AMS and FAA Orders |
| Integrated Lifecycle Plan | The Integrated Lifecycle Plan provides planning information necessary to support the system throughout its lifecycle. | Integrated Technical Planning (Section 4.2) |
| Integrated Program Plan (IPP) | The IPP provides information on the overall plan for conducting the program. It provides information on program constraints, system constraints, and Specialty Engineering plans.<br>Each specific program maintains the IPP. It is an aggregate plan that includes and integrates all the program specific plans. The IPP details what tasks are to be performed, who is to do them, and when the tasks are to be performed. | Program's IPP<br>Integrated Technical Planning (Section 4.2) |
| National Airspace System (NAS) Architecture | The NAS Architecture is the technical blueprint for NAS Modernization and guides the Federal Aviation Administration (FAA) on what systems are planned for modernizing the NAS. | Synthesis (Section 4.5) |
| System Engineering Management Plan (SEMP) | The SEMP defines the plan for conducting SE in the AMS and the program. | System Engineering in the Acquisition Management System Program Lifecycle (Chapter 3) |
| Requirements | Requirements provide information about the system's required characteristics, specifications, performance, and requirements. They assist in developing the system description.<br>System requirements are documented in the initial Requirements Documents (iRD), final Requirements Documents (fRD), and system specification(s). The execution teams and SE develop and maintain the requirements documents. | Requirements Management (Section 4.3) |
| Requirements Verification Compliance Documents (RVCD) | The RVCD records the verification status of all requirements. | Requirements Management (Section 4.3) |

| Process Input | Input Purpose/Description | From Process |
|---|---|---|
| Concept of Operations (CONOPS) | The CONOPS is a user-oriented document that describes system functional characteristics for a proposed system from the user's viewpoint. It explains the existing system, current environment, users, interactions among users and the system, and organizational impacts. The CONOPS document is written in order to communicate overall quantitative and qualitative system characteristics to the user, buyer, developer, and other organizational elements. | Functional Analysis (Section 4.4) |
| Functional Architecture | The Functional Architecture identifies, analyzes, and describes the functions of a system. It provides information required for a system description and assists in the definition of requirements.<br>Functional Analysis is the process of turning a need or system requirement into a description and an architecture of functions (system behaviors or behavior descriptors). The execution teams and/or SE perform and maintain the Functional Architecture. | Functional Analysis (Section 4.4) |
| Operational Services and Environmental Description (OSED) | The OSED is a comprehensive, holistic Communications, Navigation, and Surveillance (CNS)/Air Traffic Management (ATM) system description. It describes the services, environment, functions, and mechanizations that form a system's characteristics. | Functional Analysis (Section 4.4) |
| Description of Alternatives | Description of Alternatives is described as Physical Architectures. When performing Trade Studies (Section 4.6), a number of alternatives shall be competitively evaluated. | Synthesis (Section 4.5) |
| Physical Architecture | Physical Architecture identifies and defines the system and its components, including the physical interfaces among products, subsystems, humans, lifecycle processes, and external interfaces to higher-level systems or interacting systems. | Synthesis (Section 4.5) |
| Interface Control Document (ICD) | The ICD controls interface design. | Interface Management (Section 4.7) |
| Analysis Criteria | Criteria for specialty engineering analyses are specified to establish the degree of validation required for the analyses and associated tools, the methods to use to ensure that the data is of the proper quality and range, and the level of documentation required. | Integrity of Analysis (Section 4.9) |

| Process Input | Input Purpose/Description | From Process |
|---|---|---|
| Baselines | When the requirements and design have reached sufficient maturity, they are baselined to facilitate management of the configuration. | Configuration Management (Section 4.11) |
| Validation Reports | Validation Reports document the results of the Validation effort. They report requirements that are validated and those that are considered nonconforming. | Validation (Section 4.12) |
| NAS SEMP | The NAS SEMP describes the overall SE used in the FAA. | Manage System Engineering (Section 4.14) |

#### 4.8.0.3 General Specialty Engineering Process Tasks

Most, if not all, Specialty Engineering disciplines follow a similar process when conducting associated analyses. The following paragraphs provide general guidance on performing Specialty Engineering in the FAA. These processes are depicted in Figure 4.8-1. The process tasks are:

- Describe the system in physical and/or functional terms. This task has to be completed before the specialists may begin; if not, the specialists may perform this task, as long as it is performed according to the guidance in Functional Analysis (Section 4.4) and Interface Management (Section 4.7)).
- Bound the problem and define Constraints on the Specialty Engineering study and the design.
- Select analytical methods and tools.
- Analyze system parameters to determine specialty attributes specific to the views of the Specialty Engineering study.
- Define or assess the Specialty Engineering Requirements.
- Coordinate results with stakeholders.
- Document the analysis in a Design Analysis Report (DAR).

The following paragraphs detail the process tasks depicted in Figure 4.8-1.

##### 4.8.0.3.1 Task 1: Obtain or Develop an Operational Services and Environmental Description

The first task in performing a Specialty Engineering analysis is to understand and describe the system under study at an appropriate level. The OSED is an excellent source of this information; it is a system description that is developed in the Functional Analysis process (Section 4.4).

It is recommended that the specialty engineer use the existing descriptions to frame the Specialty Engineering analysis. However, there are times when the existing system descriptions are insufficient in detail for the specialist. In these cases, the specialty engineer develops the system description. When developing the system description, the specialty engineer shall comply with the guidance in Functional Analysis and Interface Management (Section 4.7).

Functional Analysis describes the desired behaviors of a system. These behaviors provide critical insight into how the system is intended to perform and, therefore, are a critical input to any Specialty Engineering analysis. To perform an assessment of a system, the engineer is required to understand the functions of that system and be able to relate the specialties to these functions. Normally, the Functional Analysis is completed before the Specialty Engineering process begins, and all that is required of the specialty engineer is to obtain and review the Functional Analysis and use it to enhance or complete the system description. In some cases, either because the engineers failed to perform it or because it is too early in the design process, the Functional Analysis is not available. In these cases, the specialty engineer shall refer to guidance in Functional Analysis and perform the Functional Analysis independently.

#### 4.8.0.3.2 Task 2: Bound the Problem and Define Constraints on the Study and Design

Every system problem or analysis has breadth and depth. The breadth of a system analysis refers to the system boundaries. Boundaries limit the system to elements of the system model that affect or interact with each other in order to accomplish the central mission(s) or function. Depth refers to the level of detail in the description. The level of detail in an analysis varies inversely with the breadth of the system. For a system as broad as the NAS, the description and analysis are general in nature with little detail on individual components. On the other hand, a simple system, such as a valve in a landing gear design, includes significant detail to support the assessment.

Constraints on the design play an important role in the conduct of the analysis and the credibility of the results. It is essential to identify the Constraints before the analysis to account for their influence on the methods used and the alternatives chosen. As part of determining the Constraints, the scope of the analysis, the ground rules, and assumptions are identified. Identifying the customer(s) for the analysis is important with respect to defining the scope. The analysis may be subject to contractual restraints if it is a deliverable; therefore, it is necessary to consider these restraints when defining the scope of the effort. The project schedule and budget may also impose limits on the analysis, which may affect the assumptions and ground rules. The analysis team and the recipients of the report shall be aware of all the scope limitations, ground rules, assumptions, and guidelines that apply to the assessment and product design. The following sources are used to identify Constraints:

- CONOPS defined via Functional Analysis (Section 4.4)
- Contract Statement of Work (SOW), including referenced standards and procedures
- Compliance documents that apply to the analysis methods and report
- Customer-specified requirements on cost, schedule, and product performance
- Management-imposed business goals and Constraints
- Functional, performance, and interface requirements derived from the design concept

- Functional, performance, and interface requirements imposed by the use of commercially available or preexisting hardware and software
- Operational constraints imposed by the user
- Environmental constraints imposed by the physical and operational environment
- Constraints imposed by the production or Verification process (Section 4.12)
- Design constraints imposed by standard practices that are defined by the government or standards-setting bodies

#### 4.8.0.3.3 Task 3: Select Analytic Methods and Tools

To ensure Integrity of Analyses (Section 4.9), the engineer selects analytic methods and tools that meet the program phase; the system analysis needs; and cost, schedule, and skill constraints. It is important to select methods and tools that match the analysis objectives within the resource limitations of the effort.

#### 4.8.0.3.4 Task 4: Analyze System Parameters To Determine System Attributes

In this step, the attributes of the design are determined by using the methods and tools appropriate to the Specialty Engineering discipline. The appropriate guidelines and handbooks for each Specialty Engineering discipline are listed in Table 4.8-4. The AMS FAA Acquisition System Toolset (FAST) often contains guidelines for these activities. For example, it is recommended that the team, if conducting a safety assessment, consult the FAA System Safety Handbook (SSH) and the NAS System Safety Management Plan (SSMP) found in the FAST. For some analyses, it is recommended that the results include programmatic attributes, such as cost and schedule impacts, as appropriate to the analysis.

In addition, the SE or project team, as part of this process, conducts technical or peer reviews of the analysis and its results. The technical community conducts this independent evaluation before the Specialty Engineering DARs are submitted.

The results of Specialty Engineering analyses confirm design attributes necessary for acceptable product performance, cost, schedule, and risk. When an attribute is not confirmed, the analysis and/or the baseline shall be revised.

Revision may be implemented through changes in scope, ground rules, assumptions, and analytic methods. The analysis process is reactivated with the intent of determining an alternative result that is acceptable and valid. Alternatively, the results of the analysis may drive revision of the Requirements or design Baseline. This revision is accomplished by preparing appropriate change proposal documentation for input to the Configuration Management process (Section 4.11).

**Table 4.8-4. Guidelines and Handbooks for Conducting Specialty Engineering**

| Phase | Analysis | Guidance/Reference |
|---|---|---|
| Mission Analysis | $E^3$<br>EMC requirements | FAST. (2000). Environment/Energy/ Safety/Health. http://fast.faa.gov/<br>FAST. (2000). Radio Spectrum Management. http://fast.faa.gov/ |
| | Environmental Requirements Analysis | FAST.[1] Environment/Energy/Safety /Health. http://fast.faa.gov/ |
| | Human Factors Functional Analysis | FAST. Human Factors. http://fast.faa.gov/ |
| | Human Factors System (Mission) Analysis | FAST. Human Factors. http://fast.faa.gov/ |
| | Maintainability Requirements Analysis | FAST. Sustainment and Maintenance. http://fast.faa.gov/ |
| | Operational Safety Assessment (OSA) | FAST. System Safety Management. http://fast.faa.gov/<br>FAA SSH[2], Chapter 4.<br>NAS SSMP[3], Chapters 3 and 4. |
| | Reliability Requirements Analysis | (Reserved) |
| Investment Analysis | Comparative Safety Assessment (CSA) | FAST. System Safety Management. http://fast.faa.gov/<br>FAA SSH, Chapter 4<br>NAS SSMP |
| | EMC Control Plan | FAST. (2000). Environment/Energy/ Safety/Health. http://fast.faa.gov/<br>FAST. (2000). Radio Spectrum Management. http://fast.faa.gov/ |
| | Human Factors Function Allocation | FAST. Human Factors. http://fast.faa.gov/ |

[1] Federal Aviation Administration, Federal Acquisition System Tools (FAST), Office of Research and Acquisitions (ARA), [On-line] Available: http://fast.faa.gov.

[2] U.S. Federal Aviation Administration, "FAA System Safety Handbook," FAA Office of System Safety (ASY), Washington, D.C. (2000).

[3] U.S. Federal Aviation Administration, "NAS Modernization System Safety Management Plan," FAA Office of Architecture and SE (ASD), Washington, D.C. (2000).

| Phase | Analysis | Guidance/Reference |
|---|---|---|
| | Human Factors Program Plan | FAST. Human Factors. http://fast.faa.gov/ |
| | Maintainability Plan | FAST. Sustainment and Maintenance. http://fast.faa.gov/ |
| | Preliminary Hazard Analysis (PHA) | FAST. System Safety Management. http://fast.faa.gov/<br>FAA SSH, Chapter 8<br>NAS SSMP |
| | Quality Engineering Plan | FAST. Quality Assurance. http://fast.faa.gov/ |
| | Reliability Program Plan | (Reserved) |
| | Specialty Engineering Support of Trade Studies or Alternatives Analysis | FAST. Investment Analysis. http://fast.faa.gov/<br>Synthesis of Alternatives (Section 4.8) |
| | System Safety Program Plan (SSPP) | FAST. System Safety Management. http://fast.faa.gov/<br>FAA SSH, Chapter 5<br>NAS SSMP |
| | Environmental/ Hazardous Material Analysis | FAST. Environment/Energy/Safety /Health. http://fast.faa.gov/ |
| **Solution Implementation** | Failure Modes and Effects Analysis (FMEA) | (Reserved) |
| | Failure Modes and Effects Criticality Analysis (FMECA) | (Reserved) |
| | Failure Reporting Analysis and Corrective Action System (FRACAS) | (Reserved) |
| | Failure Review Board | (Reserved) |
| | Hazard Tracking and Risk Resolution | FAST. System Safety Management. http://fast.faa.gov/<br>FAA SSH, Chapter 3<br>NAS SSMP |
| | Human Factors Demonstrations, Models, and Mockups | (Reserved) |

| Phase | Analysis | Guidance/Reference |
|---|---|---|
| | Human Factors Error Analysis | (Reserved) |
| | Human Factors Operational Sequence Diagrams | (Reserved) |
| | Human Factors Operator Task Analysis | (Reserved) |
| | Human Factors Timeline Analysis | (Reserved) |
| | Human Factors Workload Analysis | (Reserved) |
| | Maintainability Analysis | FAST. Sustainment and Maintenance. http://fast.faa.gov/ |
| | Maintainability Demonstration | FAST. Sustainment and Maintenance. http://fast.faa.gov/ |
| | Maintainability Modeling | FAST. Sustainment and Maintenance. http://fast.faa.gov/ |
| | Maintenance Task Analysis | FAST. Sustainment and Maintenance. http://fast.faa.gov/ |
| | Operating and Support Hazard Analysis (O&SHA) | FAST. System Safety Management. http://fast.faa.gov/<br>FAA SSH, Chapter 8<br>NAS SSMP |
| | Reliability Centered Maintenance (RCM) | (Reserved) |
| **Solution Implementation** | Reliability Development Growth Testing (RDGT) | (Reserved) |
| | Reliability Modeling | (Reserved) |
| | Sneak Circuit Analysis | (Reserved) |
| | Subsystem Hazard Analysis (SSHA) | FAST. System Safety Management. http://fast.faa.gov/<br>FAA SSH, Chapter 8<br>NAS SSMP |
| | System Hazard Analysis (SHA) | FAST. System Safety Management. http://fast.faa.gov/<br>FAA SSH, Chapter 8<br>NAS SSMP |

164

#### 4.8.0.3.5 Task 5: Define and Document Specialty Engineering Requirements

The Specialty Engineering products described in "Task 4: Analyze System Parameters to Determine System Attributes" (Paragraph 4.8.0.3.4) result in the definition and assessment of Specialty Engineering-related Requirements. These Requirements shall meet the standards for requirements definition and documentation described in Requirements Management (Section 4.3). In addition, these Requirements shall be validated and verified, as described in Validation and Verification (Section 4.12).

#### 4.8.0.3.6 Task 6: Coordinate Results With Stakeholders

The results of the Specialty Engineering process (particularly the DARs and Requirements) shall be coordinated with the project/program stakeholders. This coordination is conducted in both formal and informal forums. The informal forums include peer reviews and working groups. The formal forums include Acquisition Reviews and Design Reviews, as described in Integrated Technical Planning (Section 4.2).

#### 4.8.0.3.7 Task 7: Document the Specialty Engineering Analysis in a Design Analysis Report

The primary output of any Specialty Engineering function is the DAR, which documents the results of the specific analysis with rationale. Each DAR shall contain the following results:

- Description of the system's special characteristics
- List of existing Requirements that were either validated or verified in the analysis
- Residual risks
- Candidate Requirements found as a result of the analysis

These Requirements are inputs to the Requirements Management process (Section 4.3) and shall be considered for inclusion in iRD and fRD. The rationale includes the scope, ground rules, assumptions, constraints, methods, and tools applicable to the analysis.

The Specialty Engineering outputs are often used to validate and/or verify requirements. In addition, change proposal documentation is produced if the conclusions of the analysis call for a revision to the Requirements or design Baseline. This revision is an input to the Configuration Management process (Section 4.11) for authorization to change the Baseline as the analysis indicates.

Requirements for contents and format may be applicable to the DAR as specified by the contract. Figure 4.8-3 provides a sample outline of the contents of the DAR.

1.0 Executive Summary
2.0 Introduction
3.0 Summary of results
4.0 Summary of conclusions (including residual risks)
5.0 Recommendations (including mitigation)
6.0 System Description
  6.1 Summary
  6.2 Operational Services and Environment Description (OSED)
  6.3 Functional Analysis (if applicable)
  6.4 Requirements (if applicable)
7.0 Description of system special characteristics (detailed analysis worksheets or data)
8.0 List of candidate requirements
9.0 List of requirements that were validated and/or verified with rationale
10.0 Analysis methodology with rationale

**Figure 4.8-3. Sample Outline of a Design Analysis Report**

#### 4.8.0.4 Outputs of Specialty Engineering

The following paragraphs describe the outputs of Specialty Engineering. The outputs are:

- Certification Package
- Planning Criteria
- DARs (specific to the Specialty Engineering study)
- Specialty Engineering Requirements
- Constraints
- Tools/Analysis Requirements
- Concerns/Issues
- Demonstrations
- Verification Criteria

##### 4.8.0.4.1 Certification Package

Reserved

#### 4.8.0.4.2 Planning Criteria

Any Planning Criteria necessary for performing Specialty Engineering throughout the remainder of the program's lifecycle need to be provided to the Integrated Technical Planning process (Section 4.2)

#### 4.8.0.4.3 Design Analysis Report

The DAR is the means of documenting and reporting the methods and results of the Specialty Engineering analyses. Figure 4.8-3 provides a sample outline of a DAR.

#### 4.8.0.4.4 Specialty Engineering Requirements

In the course of performing an analysis, the specialty engineer typically defines, validates, or verifies Requirements. Occasionally, the specialist discovers characteristics of the system that are not adequately specified in the existing Requirements or specification documents. If this occurs, the specialist defines those necessary Requirements consistent with the specialist's area of expertise and the requirements standards described in Requirements Management (Section 4.3).

#### 4.8.0.4.5 Constraints

Constraints necessary for performing Specialty Engineering throughout the remainder of the program's lifecycle need to be provided to the Trade Studies process (Section 4.6).

#### 4.8.0.4.6 Tools/Analysis Requirements

Tools/Analysis Requirements for performing Specialty Engineering throughout the remainder of the program's lifecycle need to be provided to the Integrity of Analyses process (Section 4.9).

#### 4.8.0.4.7 Concerns/Issues

Appendix D contains guidance on Concerns/Issues as a product of Specialty Engineering.

#### 4.8.0.4.8 Demonstrations

Demonstrations are often used to verify compliance with Requirements in servicing, reliability, maintainability, transportability, and human factors engineering. Demonstrations are used to verify what is accomplished by operating, adjusting, or reconfiguring items performing their design functions under specific scenarios. The items may be instrumented and quantitative limits of performance monitored; however, only check sheets are required rather than recordings of actual performance data. This method is used when actual demonstration techniques may be used to verify compliance with a Requirement. Observations made by engineers or instrumentation are compared with predetermined responses based on the requirements. An example of this verification method is the demonstration of installing and uninstalling an aircraft engine in a required amount of time.

Demonstrations may also be used to validate unstable Requirements. If there is a risk inherent to a Requirement, Demonstrations may be used to determine the correct characteristics needed.

"Test and Evaluation Verification" (Paragraph 4.12.2.2.1,Verification by Demonstration) has more information on Demonstrations.

#### 4.8.0.4.9 Verification Criteria

The specialist may be called upon to define specific verification requirements, as described in "Step 3: Develop Verification Approach" (Paragraph 4.12.2.5.2.2.3). The Verification Criteria or requirements are added to the Verification Requirements Traceability Matrix (VRTM).

### 4.8.0.5 Specialty Engineering Tools

The tools used in Specialty Engineering are often unique to each Specialty Engineering discipline. They include databases, drawing tools, requirements and Functional Analysis tools, word and document processors, and spreadsheets. The selection of specific tools depends on criteria established by the particular program. These tools are identified and controlled as documented in individual Specialty Engineering plan sections of the IPP.

### 4.8.0.6 Specialty Engineering Process Metrics

The schedule completion of Specialty Engineering analyses measured against the plan is a measure of the degree to which these analyses are being effectively managed. The effectiveness of Specialty Engineering analyses may be measured by the rework of analyses or incompatibility with measured performance as an indication that these analyses are reaching inaccurate conclusions.

Of the seven general measurement categories discussed in section, the five that are applicable to Specialty Engineering are Schedule and Progress, Resources and Cost, Process Performance, Customer Satisfaction, and Product Quality. These measures, along with other candidate measures for Specialty Engineering, are provided in Table 4.8-5. It is recommended that each effort tailor these measures and add other applicable project-specific measures to ensure the contribution of necessary information to the decisionmaking processes.

**Table 4.8-5. Candidate Measures for Specialty Engineering***

| Schedule and Progress | Resources and Cost | Product Size and Stability | Product Quality | Process Performance | Technology Effectiveness | Customer Satisfaction |
|---|---|---|---|---|---|---|
| Achievement of specific milestone dates | Total effort compared to plan | Documentation of special engineering characteristics | Technical performance | Process productivity | Technology impact on product | Customer survey results |
| Test status | Resource utilization | Requirements | Defects | Process activity cycle time | Baseline changes | Performance rating |
| Percent of analysis studies completed (schedule/progress) | | Percent of requirements derived from specialty analyses | Standards compliance | Depth of the specialty analyses as a percentage versus the target depth | | |

*NOTE: The measures above and only general examples to indicate the type of information that might be included in the individual section measurement matrix.

### 4.8.1 System Safety Engineering

SSE (also called Safety Risk Management) is a Specialty Engineering discipline within SE. It is recommended that system/safety engineers and program managers refer to the FAA SSH and the NAS Modernization SSMP for detailed information regarding the planning and conduct of SSE. The following paragraphs describe how system safety is integrated into a system's overall SE.

#### 4.8.1.1 What Is System Safety Engineering?

SSE is the application of engineering and management principles, criteria, and techniques to optimize the safety of a system within the program's operational and programmatic constraints. These engineering and related management tools are used to identify, evaluate, and control hazards associated with a system. A hazard is a real or likely event that has the potential to harm people or damage the system. SSE's goal is to identify the hazards in a system early and continuously, to assess the risk (severity and likelihood) of each hazard, and to actively control the highest risk hazards. The NAS Modernization SSMP, Figure 4.2-1 (Risk Assessment Matrix) under the Safety Risk Management hyperlink in the FAST (http://fast.faa.gov/toolsets/index.htm), provides more information on this topic, as do Table 4.2.1 (Severity Definitions) and Table 4.2-2 (Likelihood or Probability Definitions).

As illustrated in Figure 4.8-4, the SSE process is a closed loop method of Risk Management (Section 4.10).

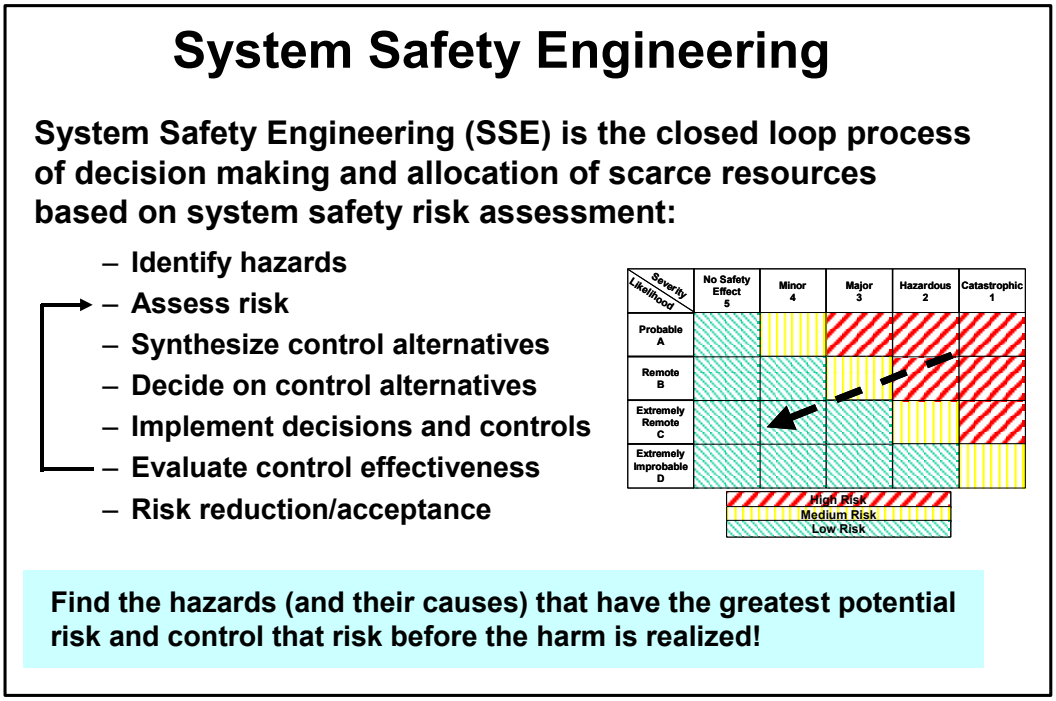



**Figure 4.8-4. Closed Loop Nature of System Safety Engineering**

To conduct SSE in the AMS, the program performs hazard analyses, as described in the NAS SSMP, Chapters 4 and 5 (http://fast.faa.gov/toolsets/index.htm), and the SSH, Chapter 8 (http://fast.faa.gov/toolsets/index.htm). Figure 4.8-5 shows what safety analyses are performed relative to the phases and decisions of the Integrated Product Development System of the AMS. These safety analyses are timed to best support the phased needs and decisions in the overall AMS process.

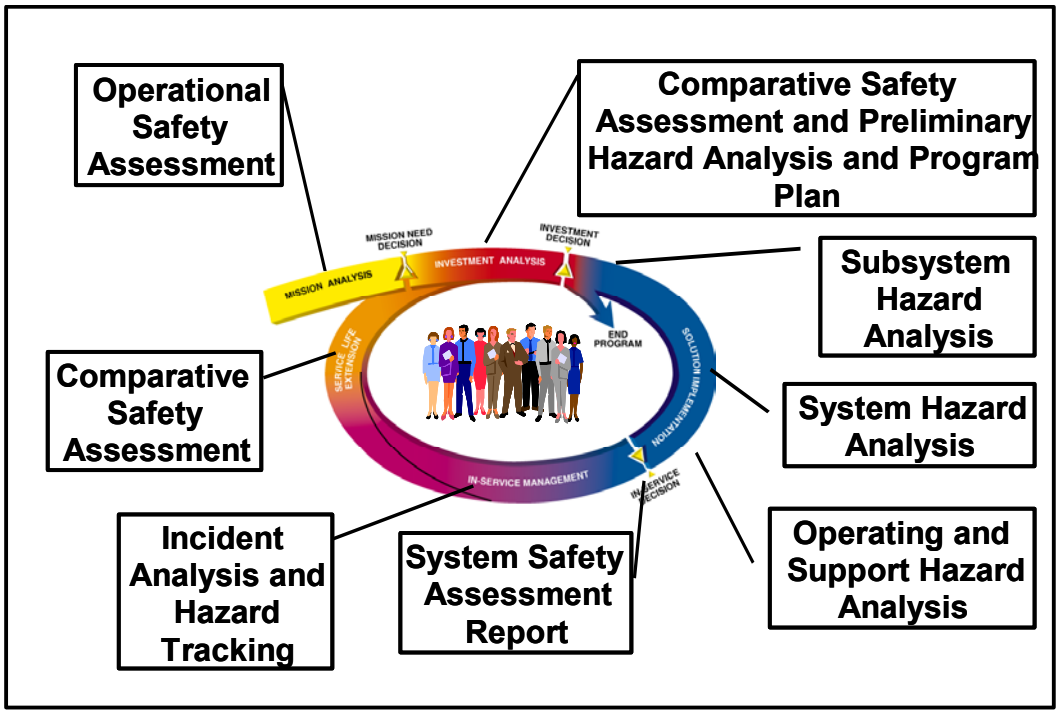



**Figure 4.8-5. Types of Safety Hazard Analyses and Their Relative Position in the Acquisition Management System**

### 4.8.1.2 Why Perform System Safety Engineering?

There are two basic reasons for performing SSE on a program:

- To comply with FAA Orders and AMS direction (e.g., FAA Order 8040.4 and AMS, Paragraph 2.9.13)
- To reduce total cost of development and improve program integration

The FAA's primary role is to ensure the safety of the NAS. In performing this role, the FAA has issued FAA Order 8040.4 (http://fast.faa.gov/toolsets/index.htm), which directs all FAA organizations to employ safety risk management in decisionmaking. The AMS was amended to comply with FAA Order 8040.4. The AMS now requires programs to execute system safety and to brief the system safety program status at all Joint Resources Council (JRC) meetings and Acquisition Reviews. The SSH, Chapter 2 (http://fast.faa.gov/toolsets/index.htm), the SSMP, Chapter 6 (http://fast.faa.gov/toolsets/index.htm), and the AMS provide more information on this subject. For example, AMS Paragraph 2.9.13 reads:

> *System Safety Management shall be conducted and documented throughout the acquisition management lifecycle. Critical safety issues identified during mission analysis are recorded in the Mission Need Statement; a system safety assessment of candidate solutions to mission need is reported in the Investment Analysis Report; and Integrated Product Teams provide for program-specific safety risk management planning in the Acquisition Strategy Paper.*

Each line of business involved in acquisition management shall institute a system safety management process that includes, at minimum, the following:

- Hazard identification
- Hazard classification (severity of consequences and likelihood of occurrence)
- Measures to mitigate hazards or reduce risk to an acceptable level
- Verification that mitigation measures are incorporated into product design and implementation
- Assessment of residual risk

Status of system safety shall be presented at all JRCs. The FAST provides detailed guidelines for system safety management (http://fast.faa.gov/toolsets/SafMgmt/IndexStart.htm).

The second reason for conducting safety risk management is that it reduces cost and improves system integration and SE overall.

- System safety looks for programmatic risks that may impact system performance, schedule, and costs.
- System safety finds problems early. As Figure 4.8-6 shows, the earlier in the lifecycle a problem is found and managed, the easier and less expensive it is to correct.

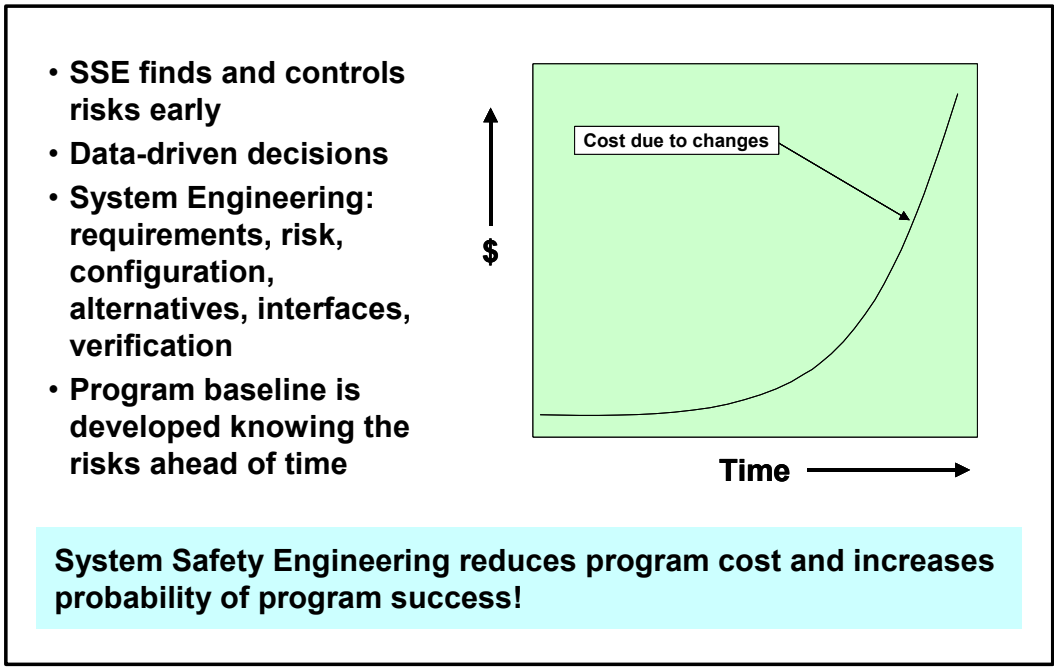



**Figure 4.8-6. Benefits of System Safety Engineering**

- The outputs of the system safety process feed other SE processes, improving the overall SE of the system (Figure 4.8-7).

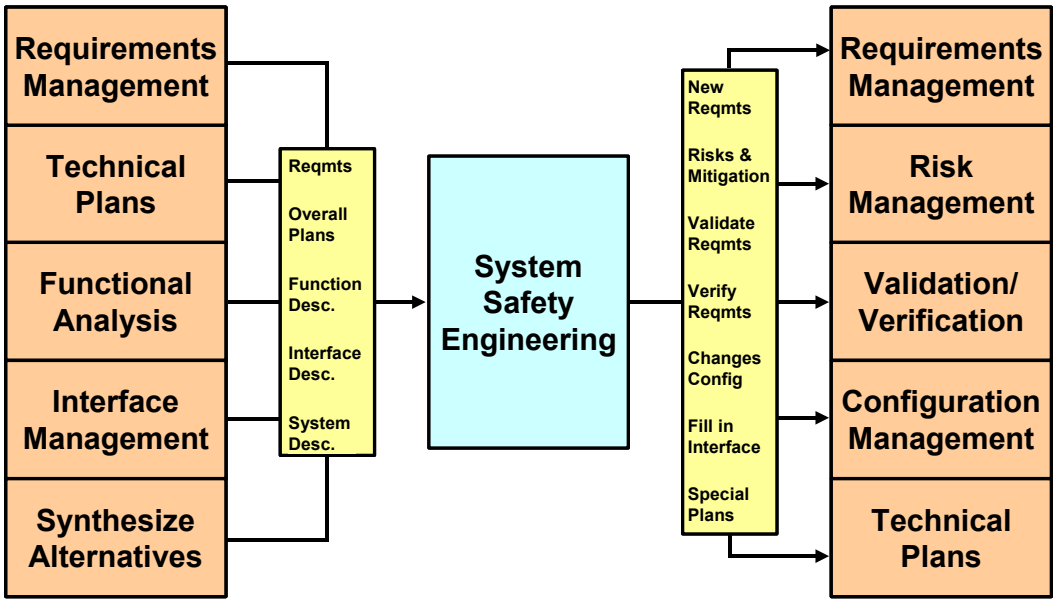



**Figure 4.8-7. System Safety Engineering's Relationship to Other System Engineering Processes**

### 4.8.1.3 System Safety Engineering Process Tasks

SSE follows the process tasks outlined in "General Specialty Engineering Process Tasks" (Paragraph 4.8.0.3).

### 4.8.1.4 System Safety Engineering Outputs/Products

The following products are outputs of SSE.

#### 4.8.1.4.1 Program Planning

Each program is required to have an SSPP. The NAS Modernization SSMP (http://fast.faa.gov/toolsets/index.htm) is the overall plan for conducting safety risk management in the AMS. It is recommended that individual programs consult the SSMP when developing a program-specific SSPP that meets the NAS SSMP requirements. The FAA SSH, Chapter 5 (http://fast.faa.gov/toolsets/index.htm), also provides guidance on this topic.

#### 4.8.1.4.2 Analysis Products

Table 4.8-6 lists the products of SSE. Detailed directions for how to develop these products are referenced in the table.

387 **Table 4.8-6. Products of System Safety Engineering**

| System Safety Process Products | How To Reference |
|---|---|
| Operational Safety Assessment (OSA) | FAA SSH, Chapters 2 and 4 (http://fast.faa.gov/toolsets/index.htm)<br>NAS SSMP, Section 5.2.1 (http://fast.faa.gov/toolsets/SafMgmt/section5.htm#5.2.1) |
| Comparative Safety Assessment (CSA) | FAA SSH, Chapters 2 and 4<br>NAS SSMP, Section 5.2.2 (http://fast.faa.gov/toolsets/SafMgmt/section5.htm#5.2.2) |
| Preliminary Hazard Analysis (PHA) | FAA SSH, Chapter 8<br>NAS SSMP, Section 5.2.3 (http://fast.faa.gov/toolsets/SafMgmt/section5.htm#5.2.3) |
| SSPP | FAA SSH, Chapter 5<br>NAS SSMP, Section 5.2.4 (http://fast.faa.gov/toolsets/SafMgmt/section5.htm#5.2.4) |
| SSHA | FAA SSH, Chapter 8<br>NAS SSMP, Section 5.2.5 (http://fast.faa.gov/toolsets/SafMgmt/section5.htm#5.2.5) |
| SHA | FAA SSH, Chapter 8<br>NAS SSMP, Section 5.2.6 (http://fast.faa.gov/toolsets/SafMgmt/section5.htm#5.2.6) |
| O&SHA | FAA SSH, Chapter 8<br>NAS SSMP, Section 5.2.7 (http://fast.faa.gov/toolsets/SafMgmt/section5.htm#5.2.7) |
| Health Hazard Assessment (HHA) | FAA SSH, Chapter 8<br>NAS SSMP, Section 5.2.8 (http://fast.faa.gov/toolsets/SafMgmt/section5.htm#5.2.8) |
| System Safety Assessment Report (SSAR) | NAS SSMP, Section 5.2.9 (http://fast.faa.gov/toolsets/SafMgmt/section5.htm#5.2.9) |
| Safety Requirements Verification Table (SRVT) | NAS SSMP, Section 5.2.11 (http://fast.faa.gov/toolsets/SafMgmt/section5.htm#5.2.11) |
| Hazard Tracking System | FAA SSH, Section 2.2.3 |

| System Safety Process Products | How To Reference |
|---|---|
| (HTS) | NAS SSMP, Section 5.2.10 (http://fast.faa.gov/toolsets/SafMgmt/section5.htm#5.2.10) |

### 4.8.2 Reliability, Maintainability, Availability – Reserved

### 4.8.3 Human Engineering – Reserved

### 4.8.4 Electromagnetic Environmental Effects – Reserved

### 4.8.5 Quality Engineering– Reserved

### 4.8.6 Information Security Engineering – Reserved

### 4.8.7 Hazardous Materials Management/Environmental Engineering – Reserved

# 4.9 Integrity of Analyses

## 4.9.1 Introduction to Integrity of Analyses

To ensure credible, useful, and sufficient data/results for program management's decisionmaking process, the integrity and fidelity of the various analysis tools shall be understood and validated. This validation takes several forms through the attributes of the tool suite, proficiency and skills of the analyst, and validity of input data. An Analysis Management Plan that outlines the details of the various analysis methods and tools is either generated or incorporated into the Integrated Program Plan (IPP). It is recommended that this planning effort reflect the available inputs and program constraints in terms of technical capabilities, schedule requirements, and cost requirements.

Analysis is a logical examination or study of a system to determine the nature, relationships, and interaction of its parts and environment. Analysis emphasizes baseline system performance and/or compares development, production, or usage alternatives. Analysis is performed throughout the product's entire lifecycle to support program decisions, encompassing technical performance and system acquisition considerations. Specific analyses are used throughout System Engineering (SE). Analyses conducted in support of a program only add value if the results are credible, useful, and sufficient.

Analysis, as described here, encompasses a broad range of perspectives. The nature of the system dictates that analysis may be performed relative to the entire system (or its subsets), the system's interaction with other systems, and/or the environment in which the system operates. Analysis may focus on the operational, functional, or physical aspects of the system and its interfaces. Analyses may range from simple to complex, quantitative to qualitative, top down to bottom up, and basic formulas to sophisticated simulations. Several specific scenarios that require analyses include:

- Exploring system concepts regarding viability and technology maturity
- Determining operational system requirements and measures of system merit
- Determining key system performance relationships to cost and other acquisition parameters
- Evaluating key system quality factors, including reliability, readiness, and maintainability
- Evaluating potential changes to improve performance, reduce cost, etc.
- Assessing risks and potential risk mitigation options
- Synthesizing allocated requirements into an acceptable physical design
- Evaluating specific physical designs (components and interfaces)
- Determining system characteristics before building or integrating the system
- Verifying system, subsystem, and component performance at various stages
- Monitoring production quality
- Diagnosing observed or perceived system deficiencies
- Evaluating produced and fielded system performance
- Evaluating processes used to support and achieve results

A wide range of tools may support analysis, including a spacecraft facility, wind tunnel, manned aircraft simulator, iron bird, computational model, physical model, computer-aided design model, spreadsheet, photograph, or paper and pencil. The analysis methods used, including tools, shall provide the required level of fidelity in representing the system or subsystem and any associated interfaces. The selected analysis method may be quantitative or qualitative, or both. The common feature of all tools is that the tools are approximations of the system being analyzed. The level of fidelity achieved is one primary feature that often sets one tool apart from another tool.

Integrity of Analyses is a disciplined process applied throughout a program to ensure that analyses provide the required levels of fidelity, accuracy, and confirmed results in a timely manner. Competent users who iteratively apply a validated set of tools to a clearly defined data set ensure integrity. The Integrity of Analyses process (Figure 4.9-1) identifies the following tasks taken to ensure integrity:

- For each analysis, identify objectives, level of detail, and degree of validation required
- Select and/or develop the tools needed to meet the identified needs
- Ensure availability of analysts proficient in using the selected tools
- Ensure availability of proper and correct input data for each analysis conducted
- Perform analysis (reference task)
- Verify that analysis results are correct, useful, and sufficient

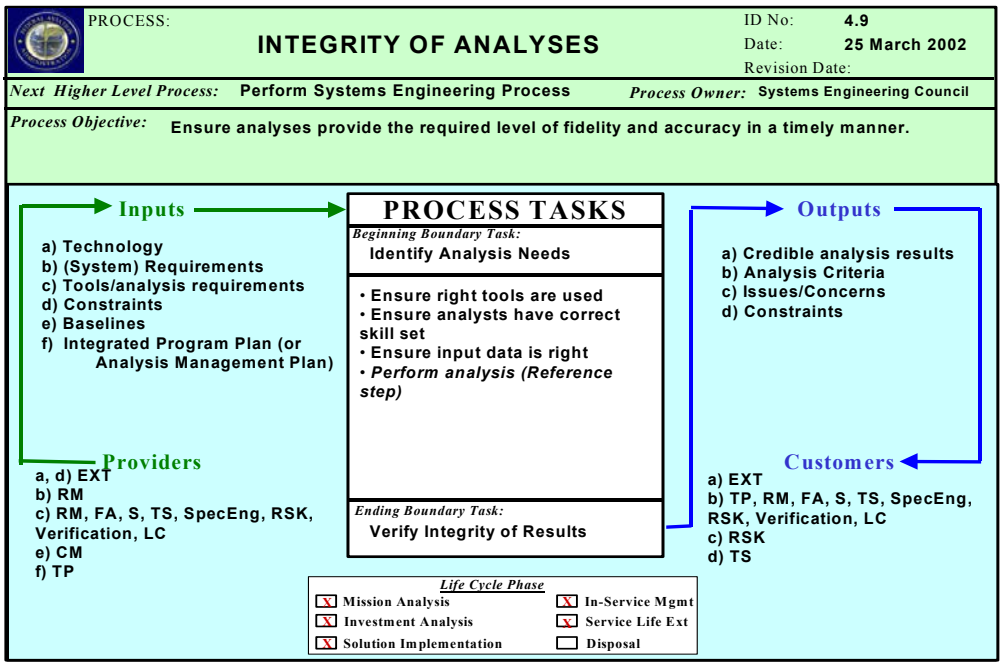



**Figure 4.9-1. Integrity of Analyses Process-Based Management Chart**

#### 4.9.1.1 Purpose

The Integrity of Analyses process provides systematic guidance that leads to analysis results, including the following:

- **Credible.** Results are valid and their implementation is feasible.
- **Useful.** Results align to their intended use in the program decisionmaking process.
- **Appropriate.** Quantity and quality are sufficient to properly aid decisionmaking without performing excessive analysis.
- **Verifiable.** Results are accompanied by a methodology, rationale, and traceability that produce an appropriate confidence level in the results.

Executing the process tasks identified in Figure 4.9-1 results in selecting the required analysis methods, performing the analysis, and verifying the results.

The initial selection of the method, tools, or model to be used in an analysis focuses on finding a practical tool that provides the most visibility into the problem with the least complexity. The process is implicitly iterative and is used across the program throughout its lifecycle. Because the process is iterative, there is an ongoing need to use the best approach to select the right method, tool, or model, considering the preferences of the stakeholders and other teams' previous experience with different tools. In addition, the limitations of budgets, technology, and schedule shall be evaluated. The bottom line is to have analyses in place that guard against mistakes and embed a consistent level of confidence in the integrity of the analysis. The analysis, in turn, contributes significantly to the success of the decisionmaking processes of program management, teams, stakeholders, and contract managers. This process is achieved by addressing the methods of analysis to be used, attributes of the toolset, quality of the workmanship, and validity of the input data. The following paragraphs define the tasks that need to be completed to achieve analysis with integrity. Figure 4.9-2 illustrates the process tasks as well as the interactions between the Integrity of Analyses process with other SE elements.

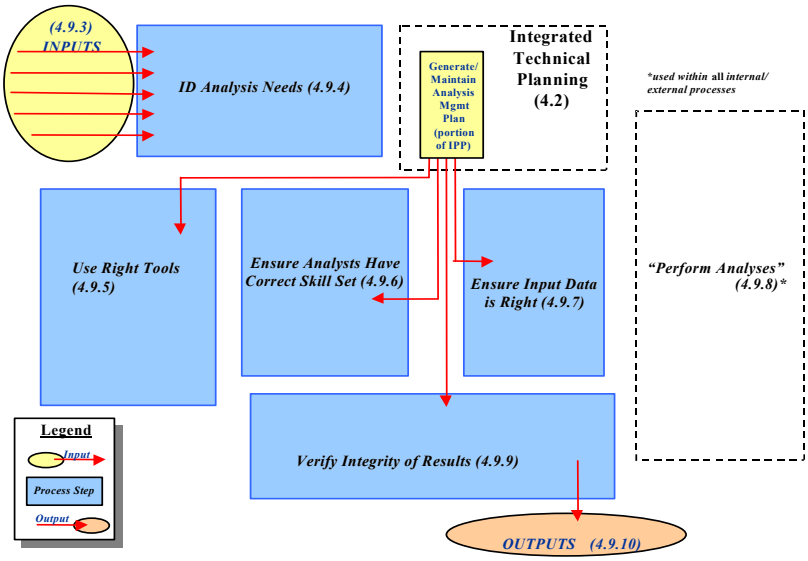



**Figure 4.9-2. Integrity of Analyses Process**

### 4.9.2 Inputs to Integrity of Analyses

**Technology.** Technology insertion determines the methods and tools employed for various analyses. The degree of technology insertion is driven by schedule and economic factors.

**(System) Requirements.** These requirements are defined to the extent that the results of a given analysis support a programmatic decision, whether driven by technical, cost, or schedule consideration. The requirements are generated from customer, internal, or supplier sources and may be expressed textually or as models.

**Tools/Analysis Requirements.** The various process elements discussed in Integrated Technical Planning (Section 4.2) that perform analyses provide requirements for tools and analysis for the project, which are constrained by program, technical, schedule, and cost requirements and plans imposed by project management. These requirements are typically reflected in the planning information developed under Integrated Technical Planning .

**Constraints.** The analysis needs are frequently a balance between the desires and costs of analytic excellence (usually championed by the analysts) and the program's cost/risk/benefit constraints, which are usually reflected in the program's budgets, schedules, and goals.

**Baselines.** This data defines the aspect of the system being modeled or analyzed. The data set is under configuration control to the extent that all elements of the program are using the same baseline.

**IPP.** As part of the IPP or as a standalone plan, the Analysis Management Plan contains the planning effort for the right tools, data, and analyst skill set. The Analysis Management Plan is developed and maintained under the Integrated Technical Planning process (Section 4.2).

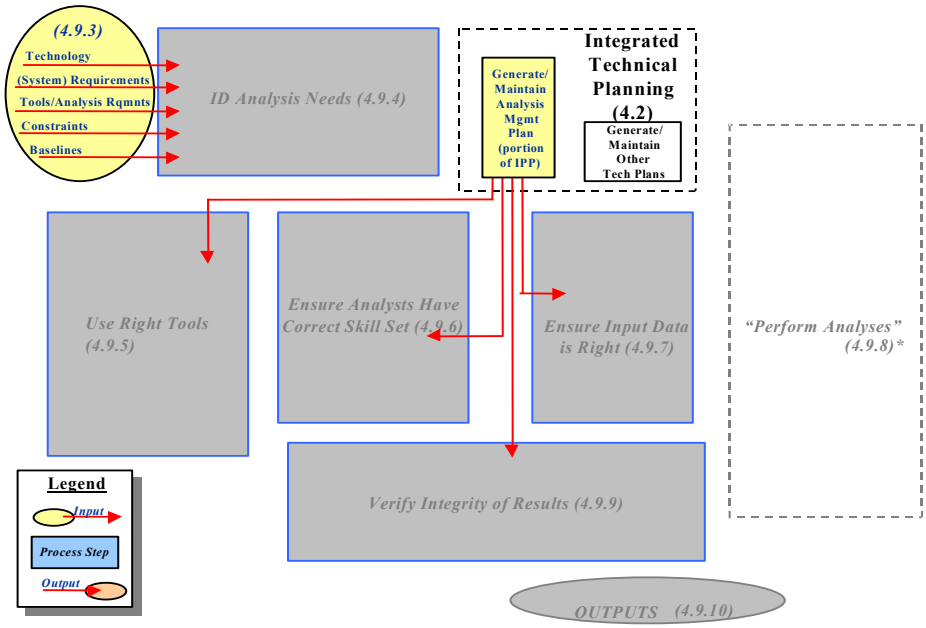



**Figure 4.9-3. Integrity of Analyses Process Inputs**

### 4.9.3 Integrity of Analyses Process Tasks

#### 4.9.3.1 Task 1: Identify Analysis Needs

As indicated by the definition in the introduction of this section, analysis is used to investigate system structure or behavior. The analysis results form a decision aid that emphasizes certain aspects of system structure or performance in a limited number of architectures or configurations. Analyses are performed for a variety of specific needs relative to the system's lifecycle. For an analysis to be truly effective, the analysis results shall be closely aligned with the expressed needs and the decisions the analysis is designed to support. It is good engineering practice to identify and plan around these needs. It is recommended that specific analysis needs be identified in the following areas:

- Understand the various perspective(s) required to aid in decisionmaking (e.g., system users, acquirers, builders, testers, and suppliers). Analysis results shall address what the stakeholders require and be capable of undergoing translation to address different stakeholder perspectives.
- Codify objectives, requirements, and constraints for the analysis itself and for managing the analysis. This includes using appropriate case definition and acceptable analysis products, as well as criteria that ensure suitability and effectiveness of the analysis when the analysis is completed. It is recommended that a concerted effort be made to identify which requirements are firm or soft and which conditions enable change.
- Obtain sufficient system and environmental definition required to conduct the analysis cases. This includes defining analysis boundaries, necessary assumptions, rationale, frequency and depth of analysis, interactions required with other analyses, and capabilities of the toolset.
- Identify control and decision points required to manage analysis methods and tools effectively. Established exit criteria for each analysis phase are useful.
- Understand data flow and organization needs associated with the analysis.

The Integrity of Analyses process tasks appear in Figure 4.9-4. Once the needs are clearly understood and addressed, the foundation is laid for managing the analysis set to obtain the needed results, which then serve as the basis to generate the Analysis Management Plan, as described in Integrated Technical Planning (Section 4.2).

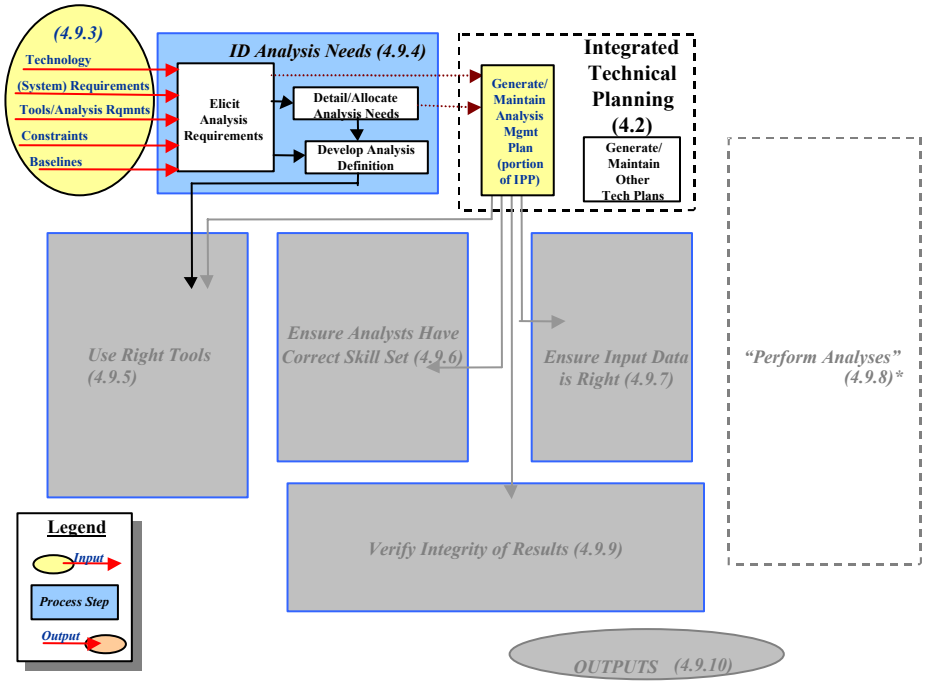



**Figure 4.9-4. Identify Analysis Needs**

Early analysis planning is key to program success. The quality of the analysis effort across the program is augmented by developing the program's Analysis Management Plan, a living document that manages and controls program analysis activities. The plan typically begins with a clear statement of project management goals, philosophies, and policies, followed by data to support planning for the analyses to be performed. The analysis needs are frequently a balance between the desires and costs of analytic excellence (usually championed by the analysts) and the cost/risk/benefit constraints of the program's budgets, schedules, and goals.

The Analysis Management Plan identifies required levels of analysis and data required to perform an analysis, defines procedures for ensuring analyst competency, contains details on the subset of analysis methods and tools that may be used for a validated analysis, and defines the criteria to ensure integrity of the analysis results. The plan provides specific tailoring required by the project and is updated when a new tool is validated on the program or when a currently validated tool is updated to reflect a change in product design and is subsequently revalidated. Because new methods and tools may be needed for product variants, and because multiple versions of a product may exist concurrently, the Analysis Management Plan may refer to multiple validated versions of the same tool.

#### 4.9.3.2 Task 2: Ensure the Right Tools Are Used

Developing meaningful system performance and cost estimates, establishing the associated system performance and design requirements, and defining acceptable tolerances may be accomplished only if analyses and performance models/databases are well defined and controlled and demonstrate validity. In addition, it is essential that analysis tool/model changes, updates, and predicted performance variances are properly identified and tracked over time.

Independent but related tools/models and simulations shall also be validated by comparison with a single reference baseline to ensure consistency of results.

An excellent and frequently stated guideline for choosing a tool/model is to select one that provides the most visibility into the problem but has the least complexity. It is recommended that practical tools/models include only features that are necessary for exploring the interactions between the study, object system, and its environment. There are many inhibitors to applying this guideline. Familiarity with a tool or a model often biases use of the tool. Lack of familiarity, inadequate training, or a “not invented here" syndrome may cause engineers to avoid a tool. The customer may expressly require use of a particular simulation tool or model. Management may demand that a single model be used throughout the program’s life. A sound SE approach to select the right tool may overcome a number of these inhibitors. To ensure the proper selection of a tool, it is recommended that the requirements of the analysis be considered, including:

- Analysis objectives
- Required level of fidelity and accuracy
- Cost controls
- Schedule constraints
- Need for additional resources

Analysis needs are allocated to tool components. Allocation includes assessing the level of fidelity required for each study function. For example, one study may require high fidelity if thrust, fuel flow, and range are being assessed, but lower fidelity in target selection. Tools that satisfy the functions and allocated study requirements for the model may be selected from existing tools or modifications to existing tools or by creation of new tools (Identify Candidate Tools task). Each tool shall be examined to verify its ability to meet the analysis needs of the project before it is selected for use. Existing tools may not provide the functionality needed for the analysis. Under these circumstances, the project is faced with modifying an available commercial-off-the-shelf (COTS) product, developing a proprietary tool for that application, or reconsidering the analysis scope. This evaluation shall be performed periodically to ensure that the tools continue to satisfy current project requirements. In addition, analysis shall be performed to assess the availability of new technology in tools, as it becomes available, and determine when it is prudent to switch to the newer technology, factoring in the costs of migration in terms of people, time, and money. The considerations for selecting the right tool(s) appear in Figure 4.9-5 and discussed in detail below.

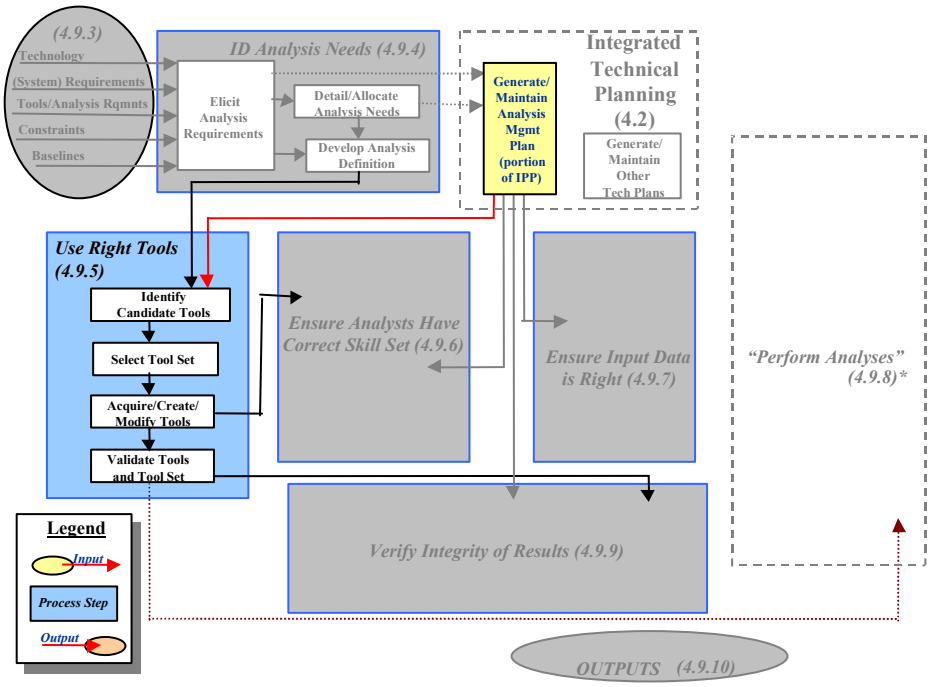



**Figure 4.9-5. Use the Right Tools**

#### 4.9.3.2.1 Tool Fidelity

One of the more confusing aspects of proper model selection is fidelity. High fidelity, and high visibility into cause and effect are usually contradictory goals for a tool. As fidelity increases, basic top-level interactions and characteristics become obscured, which is particularly true of tools that incorporate random choices. Even in totally deterministic tools, the decision logic may become so complex that the visibility is lost. Occasionally, high fidelity may block goal achievement. High fidelity tools often provide more than is realized, and engineers remain blissfully unaware of the true causes of model results.

For practical reasons, such as cost and schedule, the best choice is to use the least fidelity possible but a model that still includes the desired effects. One shall constantly look for interactions between system components and the environment that require modeling at a higher fidelity. If the system under study includes a human decision process, then one of the best ways to determine such interactions is by talking to people who operate the real system to determine what things they attempt to use to their advantage or what they attempt to avoid. A common belief is that all man-in-the-loop models are high fidelity and complicated, which is not necessarily the case. In one example recently used by a military contractor, aircraft, ships, and missiles are point masses with movements constrained by simple tables or equations. Sensor capabilities are modeled by simple equations that are one step above cookie-cutters. Interactions with the environment, such as earth curvature and atmospheric attenuation on infrared sensors, are modeled equally simply. This example surfaced because of a deliberate attempt during requirements allocation on a specific project to emphasize human decisionmaking rather than hardware modeling. It is a good example of breaking the traditional mold with an inexpensive alternative that fully satisfies study needs.

At the other extreme, a lack of fidelity may also distort answers. Operations analysts have used a series of air-to-air combat models to answer questions about issues such as the value of increased thrust in fighter aircraft. The early versions of these models used cookie-cutter launch envelopes for the missiles with no provisions for infrared (IR) signature and missile-seeker effects. Head-on engagements with IR-guided missiles resulted in mutual kills. As a result, increased maximum thrust for close-in maneuvering combat on current-generation aircraft like the F/A-18 or F-16 showed little improvement in expected kills and losses (two of the standard measures of effectiveness for combat aircraft). When moderately realistic IR features were introduced into the model, pilots were given the opportunity to reduce thrust in head-on situations and evaluate the results. Suddenly, the aircraft were spending about half of the engagement maneuvering in idle power to reduce the IR signature and increase survivability. Because there was much less time required to fly at maximum thrust to keep up aircraft energy (speed and altitude), the value of increased thrust over the shorter duration began to have a significant impact on the aircraft's effectiveness and survivability.

#### 4.9.3.2.2 Use of Validated Tools

Validation dictates that error incurred during examination or study is within a tolerance band that ensures that results satisfy the expressed need to the agreed confidence level. A validated analysis method or tool is one that has been proven to provide credible results at the associated level of fidelity. Validation may be performed using top-down or bottom-up techniques. Bottom-up validation is performed by comparing the methods and tool outputs, with varying sets of test case inputs, to the results of (1) another more complex validated model using the same test cases or (2) actual real-world performance (i.e., telemetry gathered in an actual flight). Bottom-up validation via real-world performance is usually difficult because it is nearly impossible to determine the model inputs required to simulate the real-world system. Top-down validation ensures credibility by verifying the top-down structure and performance of individual components. The best choice for validation is top-down validation because the issues are better understood, and there are fewer hidden assumptions.

For example, a software model that was previously validated to simulate a missile flight path without Global Positioning System (GPS) might be revalidated following the addition of a GPS receiver model in two ways:

- **Bottom-Up Validation.** The overall results of the entire model (with the new software for GPS added) are compared to either another model or real-world data.
- **Top-Down Validation.** Only the new software component added to simulate the GPS receiver (i.e., by comparison to actual hardware) and its interface with the other previously validated software are validated.

Regardless of whether validation is top down or bottom up, the algorithms and inputs used in the analysis tool shall be demonstrably correct and traceable back to their origin. Validated methods/tools shall be used regarding a program management decision. Examples of methods/tools that are candidates for formal validation include preflight modeling of unmanned aircraft/space vehicles, formal functional qualification testing, and Risk Management (Section 4.10). However, a significant amount of valuable analysis may be performed with unvalidated tools. Use of an unvalidated methodology/tool simply introduces the additional risk that the results may have reduced credibility. Often, this risk is acceptable when weighed against the inconvenience, increased cost, or inability to meet the schedule associated with forcing the analyst to use a validated method/tool. In most cases, a new or modified tool initially

is used without validation and a decision is made later regarding whether to perform validation based on expected future use of the tool.

#### 4.9.3.2.3 Tool Validation Process

Authorized analysis methods and tools are used over a broad spectrum of applications. Analysis tool validation is specific to the analyzed system(s) and performance for which the toolset is demonstrated. Both applicability and use are defined for each case validation. Situations may occur in which analysis data is required to support the program before full toolset validation is completed. To address these situations, provisional and limited validations have been identified. Both types of validations exist to satisfy program needs for analysis data in advance of full validation; however, neither invalidates the need for full validation.

- **Full Tool Validation.** Tools are validated when they have met all accreditation requirements, been recommended for validation by the responsible organization, and been reviewed by the Configuration Control Board (CCB) responsible for the baseline involved. (Configuration Management (Section 4.11) provides more information on this topic.)
- **Provisional Tool Validation.** Provisional validation may be granted when model performance has been essentially demonstrated, but compliance with all validation requirements has not been achieved.
- **Limited Tool Validation.** Tool validation may be limited to indicate that performance demonstration for full validation is incomplete, though all data indicate that model performance is correct and consistent for a limited analysis. Documentation requirements may be tailored for limited validation.

The analysis toolset validation process supports and is key to the analysis oversight responsibilities of each implementing program. Validation is based on demonstrating model performance, analyzing toolset configuration management/controls, and documenting the analysis methodology. As part of the approval process, the program manager designates approval authority for formal validation, which may be accomplished by forming an Analysis Review Board or through the Program CCB, as discussed in Configuration Management (Section 4.11). If an Analysis Review Board is established, its membership consists of a program manager, a member of SE, and a member of each project team using the analysis tools. The Program Board (Analysis or CCB) reviews applicability and use for which the analysis tool suite is to be validated. The term CCB is used for this board.

Once the analysis method/toolset has been authorized for use, the implementing program determines whether the toolset requires validation for its usage and the degree of validation. The degree of validation required varies with the lifecycle stage and other factors. Methods of validation include verifying the ability of the tool to provide answers for known test cases or to cross-check the results with other tools or methods for agreement.

The tool validation portion of the Analysis Management Plan specifies what is to be tested, how it is to be tested, and what comparisons are to be made to reference checkcases and other data in validating the analysis methods and tools. Reference checkcases provide the responsible organization a comparative way to demonstrate that a toolset may be validated. It is recommended that comparison of analysis data to reference checkcase data be included as the first step in any validation plan. To complete validation, the responsible organization may

propose any cost-effective combination of the following methodologies listed in order of decreasing priority:

- Comparison of data with the real system
- Comparison to other analysis applications whose validation basis is actual test comparison
- Comparison of data with other validated toolsets
- Technical audit of toolset performance
- Demonstration of toolset capability

It is recommended that a sensitivity analysis be performed to characterize the behavior as each input is individually varied. The purpose for which the toolset is being validated is the primary concern in determining the mix of methodologies selected for validation.

Demonstrated performance refers to the ability of an analysis to produce results that compare favorably with results obtained from the system being modeled over common areas of performance. The responsible organization proposes its performance demonstration as part of the Analysis Management Plan. The overall demonstration shall be controlled by a matrix that has analysis capabilities/characteristics on one axis and test scenarios (demonstrations) on the other axis. This matrix identifies how each analysis capability/characteristic is to be demonstrated for the purpose of certification. Once all performance demonstrations have been completed and action items assigned to the responsible organization(s) are closed, the CCB reviews the toolset validation package for completeness and assesses the need for further review before approving the validation package. In addition, a validation test matrix, which provides the CCB with a guide to validation requirements and completion status, is developed by the responsible organization. This test matrix is provided as part of the validation plan and is used by the CCB at subsequent review meetings to track validation completion status.

#### 4.9.3.2.3.1 Validation Approval Package

The final validation package shall conform to the approved Analysis Management Plan. Approval of the following documentation is required as part of the analysis toolset validation process:

- Analysis Management Plan
- Configuration control plan sections related to analysis toolset control
- Analysis certification report
- Analysis tool users manual
- Analysis tool version definition

A member of the responsible organization presents the validation package to the CCB for final approval. Once the package has been approved, the responsible organization is provided with a signed validation certificate.

#### 4.9.3.2.4 Validated Methods and Tool Configuration Management

The responsible organization maintains Configuration Management (Section 4.11) and controls the validated set of tools. All validated methods and tools are under configuration control and

are documented in the approved Program Analysis Management and Configuration Management plans. Validated tool Configuration Management shall ensure traceability of all changes to validated tools over time, identification of the specific versions of the toolset used to develop analysis results, and modeling of embedded hardware/software subsystems or components.

#### 4.9.3.2.4.1 Criteria for Analytical Tool Validation Update

Changes within the analysis toolset that do not introduce changes to modeled systems or their performance domain do not require a validation update as long as regression testing demonstrates identical results. The responsible CCB defines specific validation update requirements for each analysis toolset as part of the review. The following guidelines are used to determine if a validation update is required:

- If the analysis tool revision creates significant differences in analysis results, applicability, or use
- If there is significant program visibility and community interest in a functional characteristic that was modified
- If there are significant hardware/component changes to the systems being addressed that impact reference models, databases, or simulations
- If change accumulations account for a significant deviation from the previously validated baseline

If required changes impact more than one tool or model, the responsible CCB ensures that all affected tools/models are appropriately revised and that installed changes continue to provide comparable analysis results. In every case, the responsible CCB reassesses applicability to determine if changes are required.

### 4.9.3.2.5 Analysis Reference Standards System

Analysis results are not expected to precisely replicate results from the modeled system(s). Additionally, analysis results may not be consistent among themselves. Analyses are compared to a standard reference set of baselines to ensure consistency of results when they are used to substantiate and evaluate specific areas of system performance. Only reference analyses, reference models, and reference databases are employed as performance or design baselines.

#### 4.9.3.2.5.1 Reference Analyses

A set of authorized, validated analyses (certified in the case of simulations) is established as reference analysis methods. Accreditation of reference methods usually includes validation using actual test data. Reference simulations serve as the principal performance baseline(s) for the appropriate CCB action and provide a point of departure for derived analyses that may be used to establish the effect of proposed system design changes or to assess system sensitivities. Reference analysis methods typically include reference models and/or databases and are used to generate reference checkcases.

#### 4.9.3.2.5.2 Reference Models

In cases where overlap exists between elements being modeled by more than one validated tool, the function modeled in one particular validated tool is identified by the CCB as a reference

model. Reference models are established to capitalize on primary expertise in specific areas of performance and to provide consistency at the subsystem level. Reference models shall be segregated, validated, and made available to the analysis community.

#### 4.9.3.2.5.3 Reference Databases

Reference databases are established in cases where there is no advantage to modeling a subsystem function. Reference databases are created by a model that is used to generate tables of values that constitute the database. The database then represents the selection subsystem performance through tabulated values. Reference databases are established by the responsible CCB to provide consistency at the subsystem level, take maximum advantage of specific areas of expertise, and simplify analyses.

#### 4.9.3.2.5.4 Reference Checkcases

Reference checkcases are selected, reviewed, and distributed to each CCB and are available to the responsible organizations as the basis for certification comparison. Reference checkcases are generated by reference analysis methods, often are based on actual test events, and include relevant inputs, initial conditions, assumptions, and expected outputs in a form (e.g., hard copy and electronic media) usable by each responsible organization.

### 4.9.3.3 Task 3: Ensure Analysts Have Correct Skill Set

This task addresses the confidence required of each analysis, as stated in the Analysis Management Plan. Part of that confidence, from a programmatic sense, is derived from the analyst's proficiency. Quantification of that component of confidence may be difficult or impossible to determine, but qualitatively it shall be addressed. There are three elements involved: (1) establish the training required, (2) select the analyst(s), and (3) train the analyst(s) based on a gap analysis between the skill set required to perform the analysis and the skills the analyst already possesses.

The Analysis Management Plan shall describe the approach to be used for each analysis to ensure that the analyst possesses the necessary level of proficiency to perform the analysis. Such approaches include:

- Acceptance of credentials (e.g., validated professional degrees, personnel performance reviews, known track record) or stipulation by supervisors. The currency of such information is important—this aspect is addressed in the "Establish Training" element.
- Training accomplished within a defined previous period (and whether subsequent test or demonstration of performance validated such training). In the sense of on-the-job training, a policy of ongoing revalidation of analysts is useful, if for no other reason than to maintain a current roster of analysts and their credentials.
- Satisfactory performance in a validation trial to be judged against prescribed target results.

An extreme case in which the program may be at great risk, unless a particular analysis has a high confidence, may lead to the need to certify analysts as world-class experts in using a particular version of the tool to be used; to ensure that analysts have recent experience in its use in a similar application; and to possibly require a number of identical but independent analyses by different but independent analysts to produce results within a specified tolerance of each other or against a reference case or test data. An example might be the thermal analyses

that were required early in the International Space Station program that necessitated by a proposed change in orbit inclination coupled with a major change in assembly sequence in orbit.

For the purposes of this manual, programmatic issues as well as purely technical issues are included. Consequently, it is appropriate to discuss the distinction between competence and proficiency. Competence deals with one's ability to achieve excellence in results, no matter how much it takes. Analysts may be distinguished on the basis of the ease, speed, and/or clarity with which their results are produced. No guidelines are offered herein, but it is recommended that the matter be addressed in the Analysis Management Plan.

Another evaluation method is the technique of peer review. The practice of using a nearby colleague (typically of approximately the same competence) to review the analysis has been useful. In analytic work, the opportunities for simple neglect or even typographical errors are great, and it is impossible to easily detect personal errors; however, with peer review, these kind of problems are more easily found. An analyst may believe himself/herself capable of a certain job (and credentials may imply that), but peers may discover that his/her sphere of expertise does not include the analysis in question. (The most dangerous situation is often when one fails to realize what he/she does not know.) The Analysis Management Plan documents the implementation of program management's policies in this area. This process task appears in Figure 4.9-6.

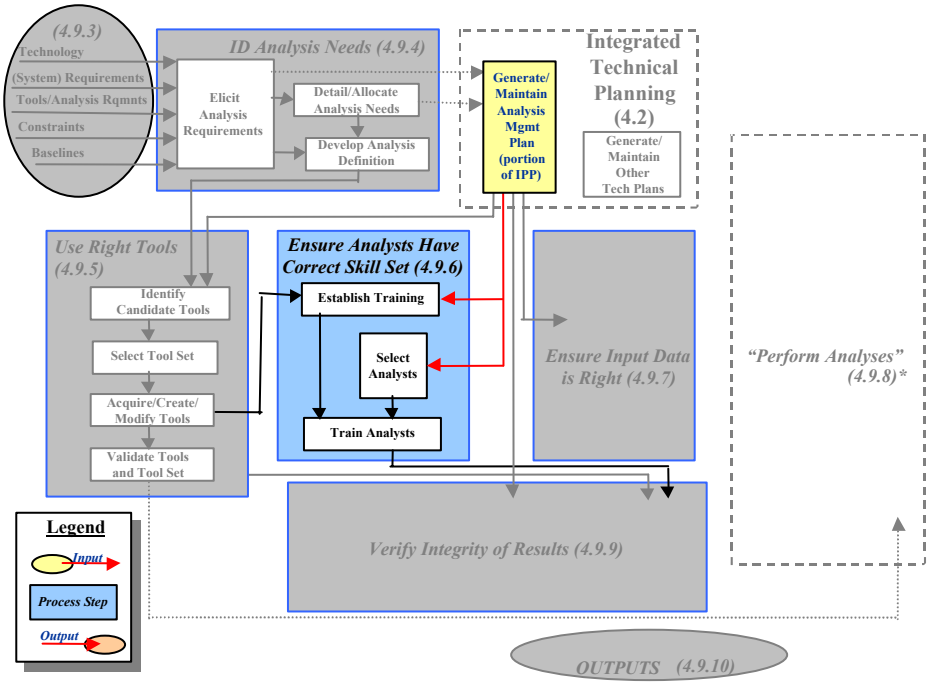



**Figure 4.9-6. Ensure Analysts Have Correct Skill Set**

#### 4.9.3.4 Task 4: Ensure Input Data is Right

It is ultimately an analyst's responsibility to determine that the data used in an analysis is appropriate for that analysis. This responsibility then flows upward in a program and organization, and the Analysis Management Plan addresses how that member's responsibility shall be supported. Special attention shall be paid to instances where analyses need to be merged or where one analysis provides input data for use in subsequent analyses. In such cases, it is especially necessary for analysts to use compatible data that agree in quality and type. The considerations involved are shown in Figure 4.9-7.

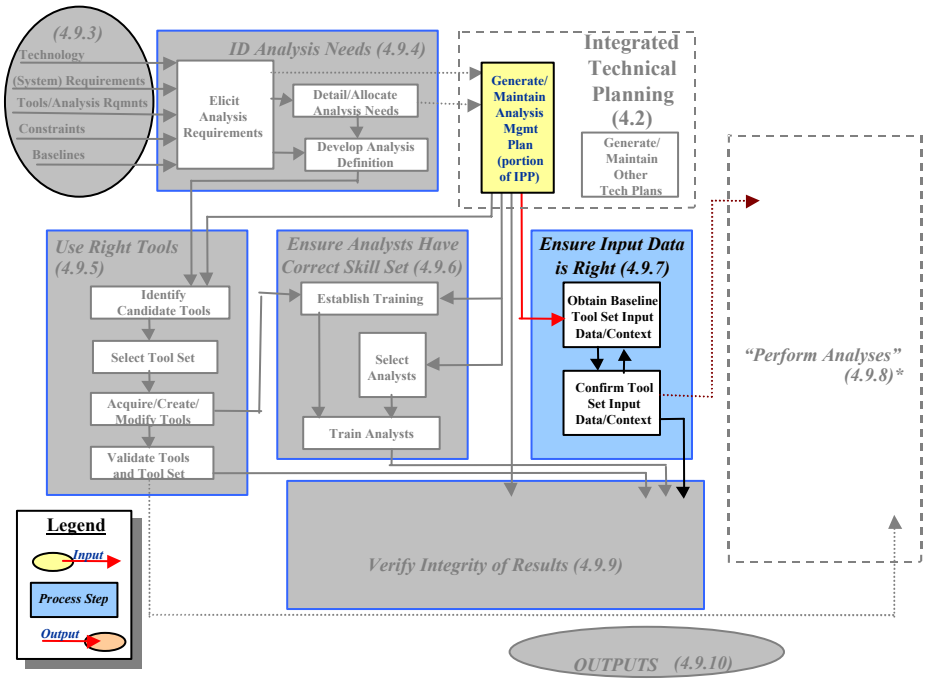



**Figure 4.9-7. Ensure Input Data Is Right**

##### 4.9.3.4.1 Data Sources and Control

The Analysis Management Plan specifies acceptable sources for each kind of data: general or universal data (such as atmospheric properties); corporate data (possibly proprietary, such as material properties or design limits); and program-specific data (such as tradeoff factors like the partial derivative of aircraft range, with respect to takeoff weight, along nominal mission). Organizational standards and libraries may exist that may be referenced, and a program may supplement those with program-unique data or define its own. The object is to provide baselined data and configuration control that baseline by the process defined in the program plan. Using baselined data results in traceable analytic results. The consistency derived from all analysts using the same baselined data produces results that may be confidently merged, compared, and/or interpreted. Besides the issue of where the data physically resides and where it may be retrieved, there is the need to document and control the identification of the data's original source. If it represents measured data, its measurement error, range of uncertainty, or confidence interval shall be recorded.

#### 4.9.3.4.2 Data Quality

One factor that shall be determined for each planned analysis is the numerical confidence interval that is acceptable in the results, which, in turn, leads to a requirement for precision, accuracy, and granularity of the input data, as well as its treatment within the algorithms. Note that granularity includes the effect of decisions regarding the number of significant digits to be used. (There is no difference between declaring Pi to be 3.14 and defining it as the ratio of a circle's circumference to its diameter if both were measured to three significant digits.) Where the scope, required level of precision, or coarseness of an analysis calls for it, the Analysis Management Plan shall specify how baselined data may be approximated or granulized.

#### 4.9.3.4.3 Documentation of Data within Analyses

An analysis is ultimately credible if it may be independently repeated. When repeatability is considered, it is clear that part of that ability is knowing exactly what data was used to create the first result; therefore, that data shall be carefully documented. The source; pedigree of validation; and extent of accuracy, precision, and granularity shall be documented, and the reader shall be made confident that all data were considered, even constants and parameters that are frequently forgotten, especially if they may have been hard-coded within a relation or equation.

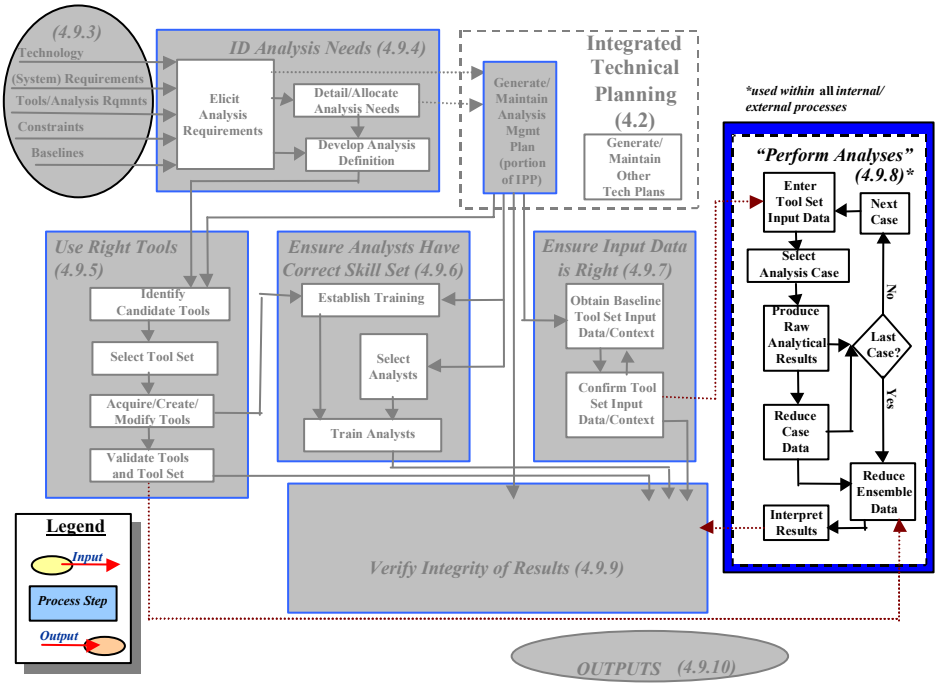



**Figure 4.9-8. Perform Analyses (Reference)**

#### 4.9.3.5 Task 5: Perform Analysis

The actual analyses performed are described in the other sections of this manual. The Integrity of Analyses process supports the other SE processes and is intended to provide a disciplined framework for conducting any required analysis, whether technical, programmatic, or

administrative in nature. The interaction between the Integrity of Analysis process and the actual performance of analyses appears in Figure 4.9-8

#### 4.9.3.6 Task 6: Verify Integrity of Results

In general, an analysis is useless unless it may be independently repeated; therefore, the inputs and all underlying assumptions shall be documented. It is recommended that criteria be established in the Analysis Management Plan for each type of analysis to ensure that the results are accurate, correct, and sufficient. The criteria are enforced by developing, validating, and using analysis templates. Comparing results from two or more truly independent analyses may be performed to achieve confidence in the results when the accuracy and/or validity of the analysis tools and methods have not been proven. The greater the independence of the individual analyses, the greater the confidence in the validity of the result.

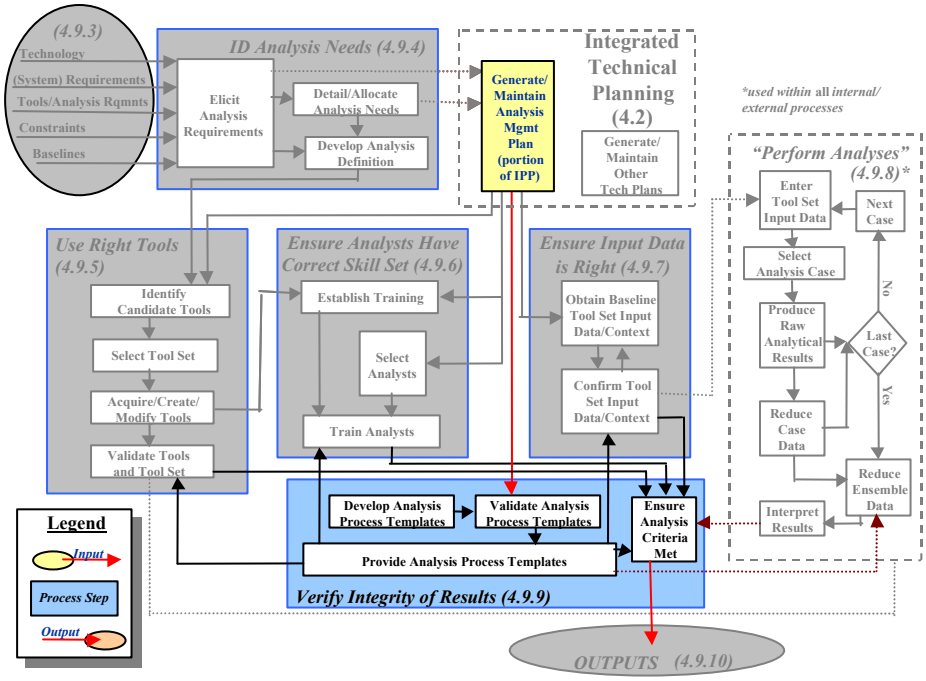



**Figure 4.9-9. Verify Integrity of Results**

Sufficiency of the analysis shall be addressed: Did the analysis consider the entire envelope of interest? Were the selected portions of the envelope adequate to draw a proper conclusion? Did the analysis account for all significant effects? In rare cases, it may be necessary to perform an analysis to determine precisely which effects need to be considered to substantiate the results of an analysis.

- Even though a modeling technique or simulation tool has been proven to be completely accurate, errors may still be present in an analysis. Errors in an analysis may be caused by omitting terms that have a significant effect on the result; using the wrong input data (e.g., wrong atmospheric model); and misinterpreting/misunderstanding input data (e.g., wrong units, wrong reference coordinate system). Independent analysis may catch these errors only if there is no collaboration between the analysts. The criticality of the results of the analysis typically determines the degree of verification justified.

- Another type of innocent error is caused by an analyst's misunderstanding of the problem statement (i.e., requirements). In this case, a completely valid answer may be presented to the wrong question.
- An analysis may be nothing more than an opinion poll. Evaluating the characteristics of the population considered only ensures this accuracy of this type of analysis.
- Ultimately, the results are verified by users per the original plan. If the results are insufficient, then a root cause analysis is performed where appropriate. The outcome of this analysis may result in the original analysis being reconducted by modifying:
  - Methods
  - Tools
  - Inputs
  - Users

### 4.9.4 Outputs of Integrity of Analyses

The two major outputs of the Integrity of Analyses process are analysis criteria captured in the Analysis Management Plan and credible analysis results (Figure 4.9-10). In addition, any residual concerns/issues generated by this process are furnished to the Risk Management process (Section 4.10) for resolution. Any constraints driven by tool selection, skill requirements, or other programmatic considerations documented in the Analysis Management Plan are furnished to the Trade Studies process (Section 4.6) to assist in defining the appropriate trade space.

#### 4.9.4.1 Analysis Criteria

The Analysis Management Plan is used to specify the analysis philosophy to be imposed. It is recommended that every analysis be understood as being bounded or constrained by all the pressures implied above. It is important to capture these issues in the Analysis Management Plan so that aspects such as the following are treated in the plan:

- The degree of validation required for each tool and type of analysis shall be specified. Tools as simple as an Excel spreadsheet or as complex as man-in-the loop simulations may be used to support programmatic decisions. A method shall be developed to verify that the correct equations are used for the analysis and that they have been properly implemented in the spreadsheet. Whatever the tool, the plan specifies the procedure for acquiring/developing, maintaining, and validating that tool. Typically, a program has a configuration control function (its own or some core organization's) from which validated tool lists may be drawn and referenced.
- Methods shall be specified to ensure that analysts are proficient in using the tools and executing the analyses. This consists of providing proper documentation, training, and review procedures.
- Methods and analysis criteria shall be specified to ensure that data of the proper quality and range, from documented sources with valid pedigrees, are under configuration control and, thus, traceable when referenced by the analysis documentation.
- The required level of documentation for each type of analysis shall be specified, usually in the form of templates. Formal analysis shall provide sufficient documentation to permit reconstruction of the results from the input data. Quick analysis used to rule out a

possible system design may not require the level of documentation or substantiation as analyses that are required to support the final system configuration.

- The review policy for each type of analysis shall be specified.

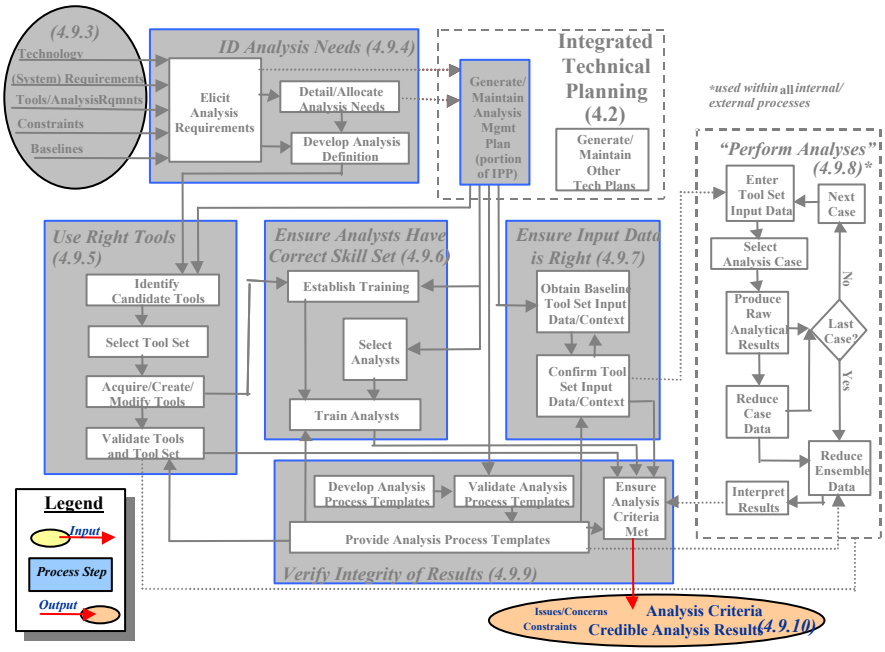



**Figure 4.9-10. Integrity of Analyses Outputs**

### 4.9.4.2 Credible Analysis Results

Simulated results are not expected to precisely replicate results from the simulated systems. Additionally, analysis results may not be consistent among themselves. Models shall be compared to standard reference baselines to ensure consistency of results when employed to substantiate and evaluate specific areas of system performance. Only reference simulations, reference models, and reference databases may be employed as performance or design baselines. Analysis results are meaningless without description of the analysis method and the assumptions that generated those results. If the tool version used to generate the results is not validated, the differences between the validated version and the version used, as well as the validation plans for the new version, are also to be presented.

## 4.9.5 Integrity of Analyses Process Metrics

There are four general measurement categories that are applicable to Integrity of Analyses, and they are shown in Table 4.9-1, along with candidate measures for analysis management. It is recommended that each effort tailor these measures and add other project-specific measures that are applicable to ensure that they contribute the necessary information to the decisionmaking processes.

**Table 4.9-1. Integrity of Analyses Measurement Categories**

| Schedule and Progress | Resources and Cost | Product Quality | Process Performance |
|---|---|---|---|
| Percent of analysis tasks completed on schedule | Existing validated model is ratio of analysis to total hours (Total = analysis hours + verification hours) | Percent of analysis "passing" verification step (first pass) | Average number of days to complete analysis (per same tool and complexity) |

# 4.10. Risk Management (Satisfies iCMM process area 13 criteria)

## 4.10.1. Introduction to Risk Management

The Perform Programmatic Risk Management process (Figure 4.1-10) provides an organized, systematic decisionmaking process that identifies risks to achieving program/project goals, analyzes these risks, effectively mitigates them, and tracks the progress of mitigation efforts. Programmatic Risk is defined as an event or situation with a realistic (non-zero and not 100%) likelihood/probability of occurring and an unfavorable consequence/impact to the successful accomplishment of the well-defined program goals should it occur. The process should be applied at all levels, from small projects to large programs, and should be applied continuously throughout the program's lifecycle looking at all aspects of the program (see Figure 4.10-2). The risks must also be "rolled up" from a project or several projects to a program. Risk roll-up involves a review of the consequences/impacts from a higher (program) level. The risks to meeting the objectives or benefits of these projects or programs are typically known as programmatic risks. This process complies with the requirements of the iCMM (Process Area 13). It also satisfies EIA 632 requirement 24 and EIA 731 Focus Areas 2.5-2 through 2.5-8.

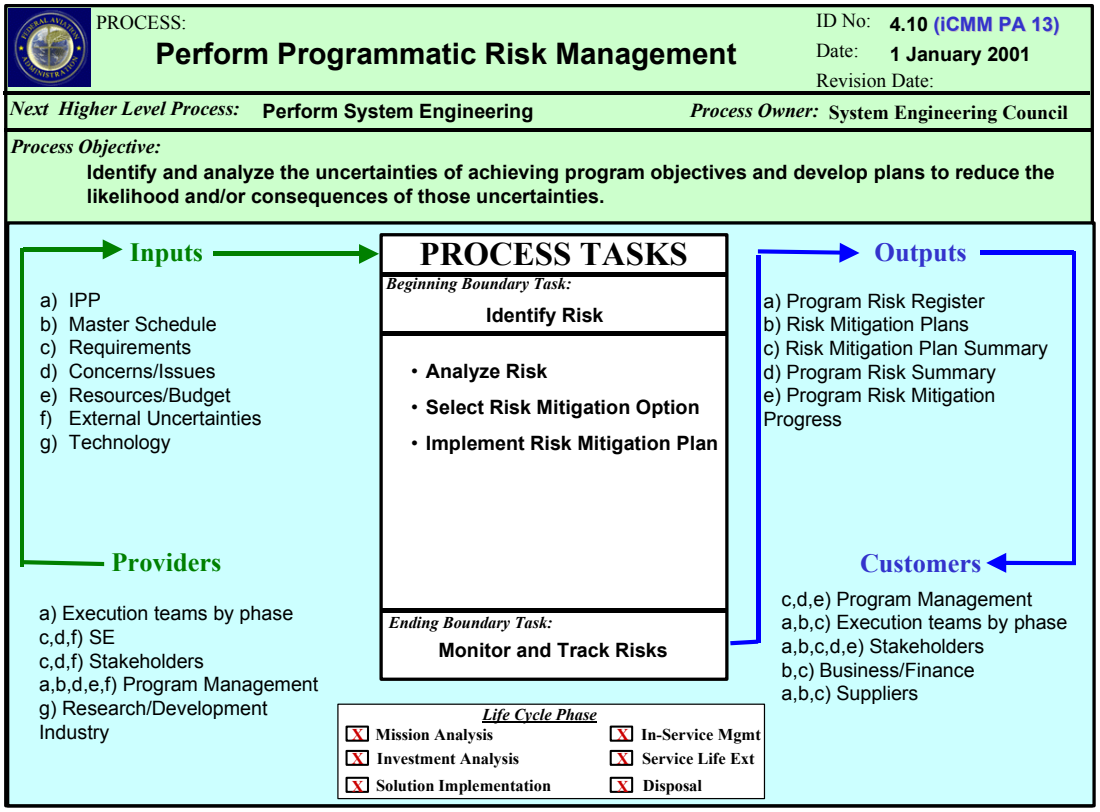



**Figure 4.10-1. Risk Management Process-Based Management Chart**

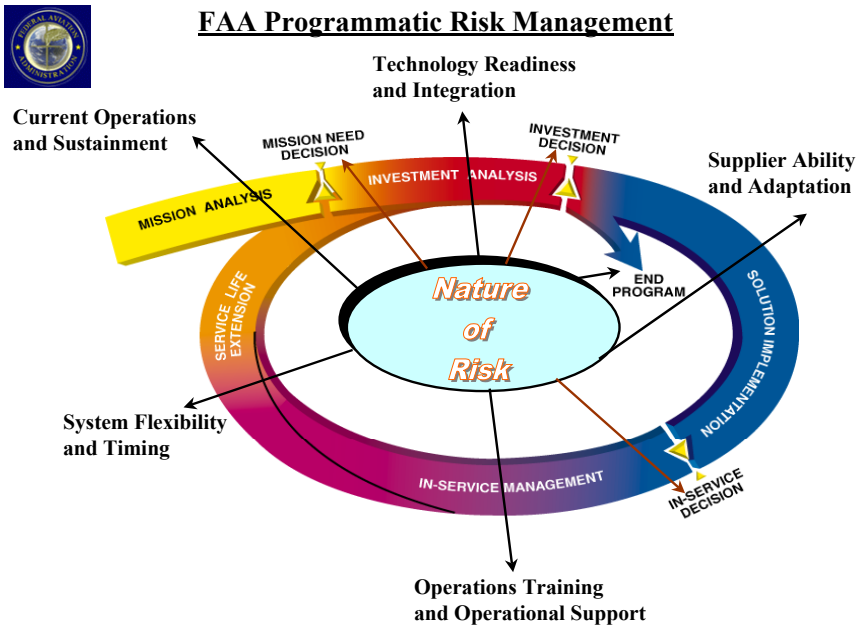



**Figure 4.10-2.**

### 4.10.1.1.Function of Risk Management

Programmatic risk management is a basic system engineering element of successful program management (Figure 4.10-3). When properly executed, Risk Management engages all disciplines and execution teams and is present in all program stages/phases. The functions (Figure 4.10-4) of the process are to:

- Identify each risk to the program
- Analyze and assess the negative consequences/impact and the likelihood/probability of the risk actually occurring and determine risk resolution date
- Develop specific approaches and plans that will mitigate the risk
- Implement the risk mitigation plan
- Monitor and track risk mitigation effectiveness

Based on results from these functions, program management can then determine:

- The amount of schedule and budget reserves that should be allocated and to what, based on identified risks
- How to measure overall program performance with respect to each risk
- How much and what type of help will be needed from other sources
- When to look at the process to see if the mitigation effort is working
- When to add mitigation efforts, costs, and milestones to program master schedule and budget

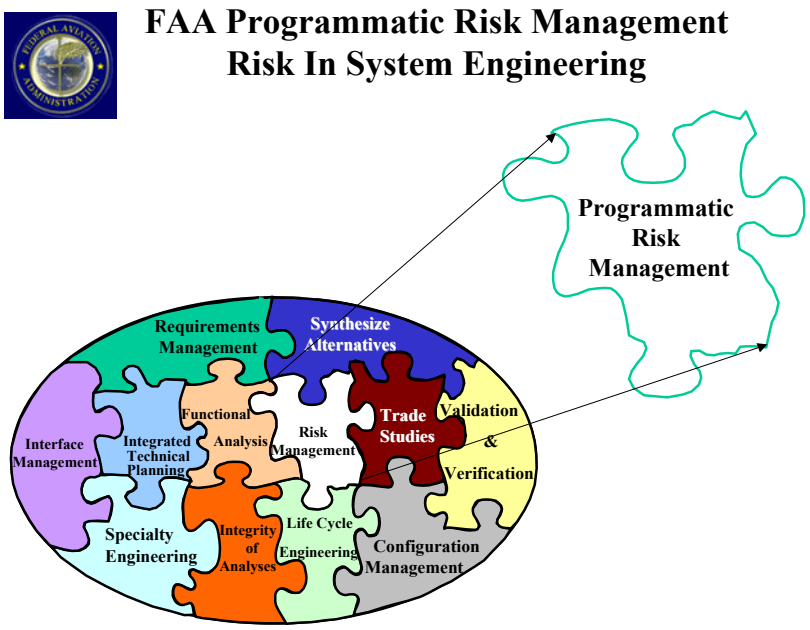



**Figure 4.10-3.**

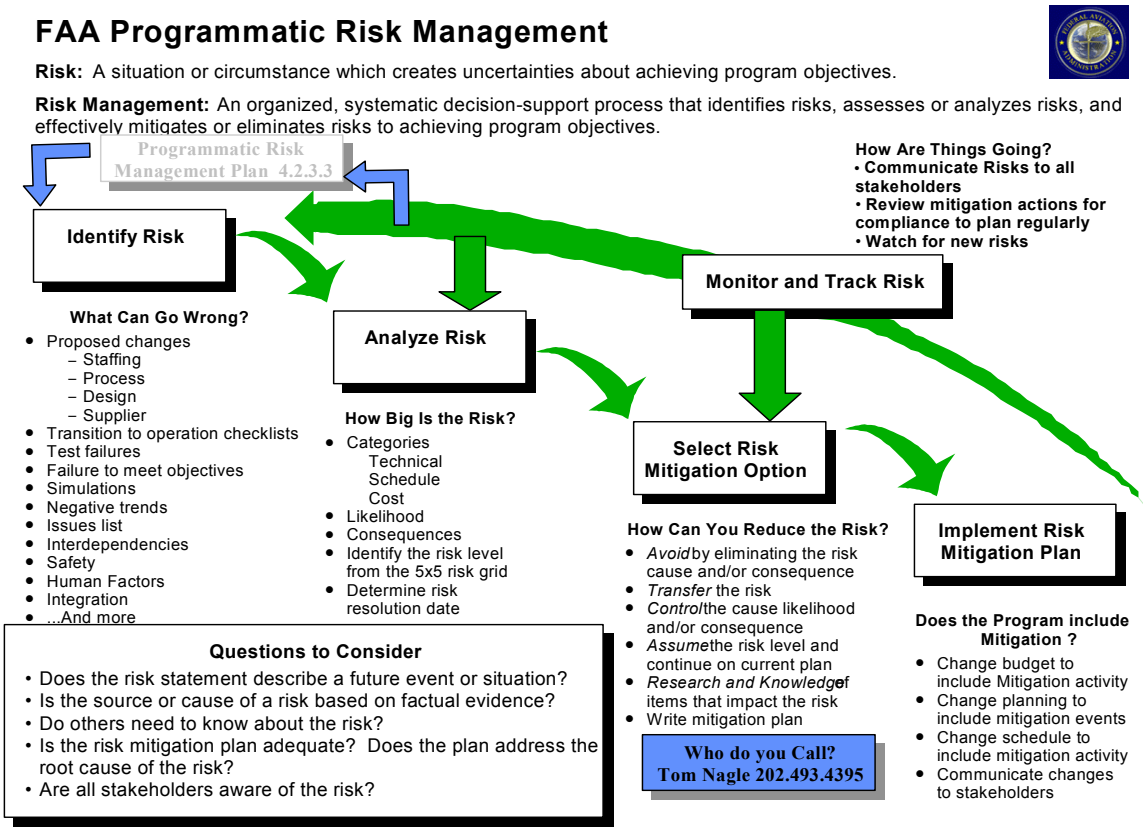



**Figure 4.10-4.**

#### 4.10.1.2. Risk Management Objectives

The fundamental objective of the Programmatic Risk Management process is to identify and analyze uncertainties of achieving program objectives and develop plans to reduce the likelihood and/or consequences of those uncertainties.

This process is applied to ensure that a program meets technical, schedule, and cost commitments; delivers a product that satisfies all stakeholders' lifecycle needs; and provides the expected benefit. Four lower-level objectives are established as part of meeting the overall objective:

- Timely identification of risks (identifying a potential problem with sufficient leadtime so the team can implement appropriate alternate plans)
- Consistent assessment of the level of risk across a program (providing a structured decisionmaking framework for prioritizing resource application)
- Communication of risk mitigation actions across the program/project (ensuring that all elements of the program/project are aligned in resolving risks)
- Review of risk mitigation action performance.

### 4.10.2. Risk Management Process Description *(Satisfies iCMM PA-14, BP13.04 criteria)*

Every participant in a program/project shares in the responsibility of assessing and mitigating risks. This process is a part of the overall program/project management and system engineering process. The process must be aligned with the individual products (hardware, services, and software) that result from consistent functional analysis and requirements allocations, the Integrated Program Plan (IPP), the master schedule, the associated funding, and the identified goals and benefits. The program is assessed as to risks associated with impacts on program benefits, interdependent programs, or environments. For each product, risks are evaluated against the acquisition baseline technical requirements, schedule, and cost, leading to the successful satisfaction of the program objectives. Risks are identified and assessed, and appropriate risk mitigation actions are established that comply with the program/project risk management plan within the IPP (see Section 4.2.2.2 in Integrated Technical Planning (Section 4.2)). This plan is developed and tailored (when the technical nature of the program demands tailoring) to satisfy specific program/project needs. *(Satisfies iCMM BP 13.01 criteria)*

Results from each assessment are a starting point for the risk mitigation plan to support program management decisions (technical, schedule, and cost). The products of this process are also shared with stakeholders to achieve alignment/acceptance of the resource decisions. All risks are examined at each program/project/event/item/peer review as defined in the risk management plan. Updates will reflect changes in risk resulting from planned mitigation activities or other unplanned events. Risk progress is actively tracked. For each risk, a "risk resolution date" (RRD) is established, marking the event at which either the risk no longer exists or when the program must be modified to accommodate the negative consequences. The question **must** be asked and answered: "What happens on this date?" Risk is "rolled up" when it is taken from a lower-level project to a higher-level program.

An essential element of the Programmatic Risk Management process from an organizational point of view is the non-advocate concept. The purpose of a non-advocate is to provide an impartial, objective assessment of the project team's results, especially with respect to the assignment of risk levels. The input of a non-advocate is essential on those projects where two or more of the project specialists disagree on the risk levels. A non-advocate would typically be, but not be limited to, a program management person (above or at the same level of the

program/project manager), a stakeholder representative, and/or a person from another project or program. The responsibility of a non-advocate is to examine and assess all aspects of the program/project risk management process before each review. For small projects, one or two non-advocates may be acceptable. A non-advocate will provide an assessment to program/project managers for consideration and action.

#### 4.10.2.1.Overview

The top-level process for Programmatic Risk Management is shown in Figure 4.10-1. The process includes steps that result in the identification of potential risks, analysis and assessment of risk, development of risk mitigation plans, implementation of the Risk Mitigation Plan, and monitoring of risk status. The process is iterative and is used across the program throughout the program's lifecycle, with the nature of the risks changing to coincide with the lifecycle stage.

#### 4.10.2.2.Inputs to Risk Management

The inputs required to initiate Programmatic Risk Management include both program/project- and product-related data as shown in Table 4.10-1. Many of these inputs are developed and refined through the continuous, iterative use of other system engineering processes. Each item in the table is to be evaluated for resultant program risk.

**Table 4.10-1. Inputs to Risk Management**

| Input | Reference |
|---|---|
| Requirements Documents | 4.3.3 |
| Integrated Program Plan (IPP) | 4.2.1 |
| System Engineering Management Plan (SEMP) | 4.2.3.2 |
| Risk Management Plan | 4.2.2.2 |
| Analysis Integrity Criteria | 4.9.5.5 |
| NAS Architecture | 4.5.5 |
| Trade Study Report | 4.6.1.4 |
| Design Analysis Report | 4.8.4.3 |
| Controlled Data and Reports | 4.11.8 |
| Contract | |
| Product Configuration Data | 4.11.3 |
| Acquisition Program Baseline (APB) | FAST |
| Acquisition Reviews | |
| Specialty Engineering Reports | 4.8 |
| Interfaces | |
| Test Plans and Reports | |
| Manufacturing/Production Information | |
| Resources/Budgets | |
| External Uncertainties | |
| Technology | |
| Acquisition Strategy Paper | |

| Input | Reference |
|---|---|
| Mission Need Statement (MNS) | |
| Concept of Operations (CONOPS) | |
| AMS Documents | |
| Technical Analysis | |
| Contractor Outputs | |
| Statement of Work (SOW) | |
| Verification Results | 4.12 |
| Training Results | |
| Maintenance Results | |
| Lessons Learned | FAST |
| Operational Results | |
| Safety Assessments | |
| Security Assessments | |
| Human Factors Assessments | |
| Integrated Master Schedule (IMS) | |
| Program Review Results | |
| System Safety Program Plan | 4.8 |
| LMS Documentation | |

### 4.10.3. Risk Management Process Tasks

The Programmatic Risk Management process is summarized in Figure 4.10-1. The major process steps shown in Figure 4.10-4 are described in the remainder of this section.

#### 4.10.3.1.Task 1: Identify Risk *(Satisfies iCMM BP 13.02 criteria*)

Risk identification includes examining a program/project and documenting potential risks in the areas of technical, schedule, and cost. Risk identification must be performed during each stage of the program or whenever significant changes occur in plans or program status. Circumstances requiring risk assessments include:

- Programmatic changes
- Unfavorable trends in Technical Performance Measures (TPMs) and predicted system performance, schedules, and financial status
- Design/program/peer reviews
- Change proposals (including proposed changes in requirements)
- Occurrence of a major unforeseen event
- Newly identified risks
- Special assessments at the direction of agency management
- Changes or risks in interdependent programs

- Environment changes

As shown in Figure 4.10-1, participants in risk identification include all stakeholders, users, suppliers, and appropriate members of execution teams. Teams will consider all likely risk sources in identifying potential risks to the program/project. Risk identification is based on the current program/project goals supported by the associated technical, schedule, and cost requirements and plans.

A programmatic risk has two aspects: (1) the likelihood/probability that an event will occur and (2) an unfavorable consequence/impact should it occur. The likelihood that a risk will occur should not be so low as to be negligible (i.e., probability essentially equal to zero) nor should it be equal to one, typically indicating that it has, in fact, already been realized. A risk must also have a negative consequence/impact if realized. Positive consequences are not considered in the FAA risk identification and analysis process; these are considered opportunities. Note that if there is no uncertainty (i.e., the situation or circumstance is certain to occur or has already occurred), there is no risk, even though the item has an unfavorable consequence. This situation should be handled as a management issue, for which a corrective action plan must be generated and implemented. This essentially requires a rebaselining of the program or possibly cancellation of the program if the negative consequences are too severe.

Each risk must have a “risk resolution date.” This is a date when either the risk no longer exists or when the program must be modified to accommodate the negative consequences. This date must be documented when the risk is identified. The question **must** be asked and answered: “What happens on this date?” The negative consequence of the outcome of the event that occurs on the given date is the basis for the risk.

#### 4.10.3.1.1. Potential Sources of Risk

Risks originate from three basic areas—technical, schedule, and cost. A risk identification flow is shown in Figure 4.10-5. Technical risk is based upon the likelihood that the program as planned will be unable to deliver a product to satisfy the technical requirements. As such, well-documented, defined, and quantified technical requirements are necessary to define a technical risk. Most of the risks listed in Table 4.10-2 will be technical risks. Schedule risk results from the likelihood that the program actions cannot be accomplished in the planned program timing. A detailed program schedule identifying each accomplishment and the critical path is necessary to develop schedule risks. Cost risk results from the likelihood that the program will not accomplish planned tasks within the planned budget. A detailed budget, in which the cost of each accomplishment is specified and any management reserve is known, is needed to determine a cost risk. Potential loss of funding is not a programmatic risk in this risk process. Within the FAA risk process, *cost* is the ultimate expenditure required for a resource and the end product produced by that resource. *Budget* is the forecast of all costs that will be planned for a given project/program, and, *funding* is the supply of money provided to accomplish a given project/program. The risk source is based on the root cause of the risk, and, as such, only a single source will cause a risk. The source is technical, schedule, or cost and not a combination or all of these. This should not be confused with the consequence, which can be performance (technical), benefit, cost, and/or schedule impact.

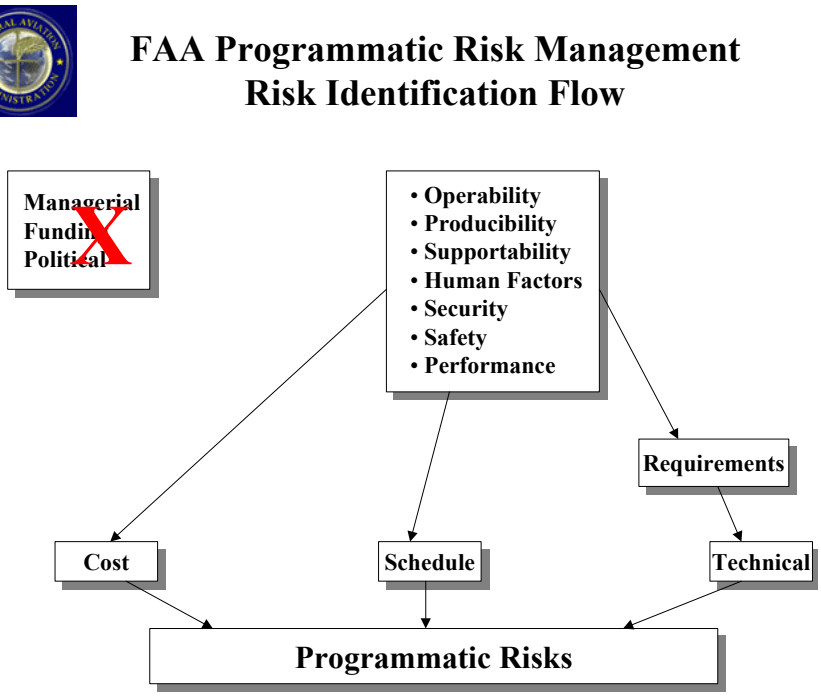



**Figure 4.10-5.**

For each risk area, many sources should be considered. For technical risk, likely sources include technology maturity, complexity, dependency, stakeholder uncertainty, requirement uncertainty, and testing/verification failure. Sources of schedule risks may include incomplete identification of tasks, time-based schedule (as opposed to event-based schedule), critical-path scheduling anomalies, competitive optimism, unrealistic requirements, and material availability shortfalls. Cost risks may stem from an uncertain number of production units, supplier optimism, additional complexity, change in economic conditions, competitive environment, supplier viability, and lack of applicable historical data.

Table 4.10-2 provides the potential sources of risk to be considered in the process of program risk assessment.

**Table 4.10-2. Potential Sources of Risk**

| Potential Sources of Risk | |
|---|---|
| • Safety | • Test |
| • Security | • Verification |
| • Maintainability | • System Integration |
| • Reliability | • Staffing |
| • Supportability | • Tools |
| • Human Factors | • System Performance |
| • Availability | • Technology |
| • Decommissioning | • Planning |

| Potential Sources of Risk | |
|---|---|
| • Producibility<br>• Commonality<br>• Training<br>• Operations | • Transition<br>• Environments<br>• Interdependencies (both FAA and non-FAA) |

#### 4.10.3.1.2. Risk Identification Methods

Risk identification begins at the lowest feasible level and normally includes inputs from all stakeholders and suppliers. Anyone can identify a potential risk. Experts should review programs to determine risks. Similar programs should be reviewed for determined risks as well as actual problems. The objective of this step is to produce as comprehensive a list as possible of potential risks. This may be achieved using any combination of methods, such as group discussions, interviews, trend/failure analysis, risk templates, lessons learned, trade studies, Best Practices, metrics, and acquisition documentation. The focus should be on root causes and not on symptoms of a more basic problem. The problem must be defined at the lowest level (root cause) so that the mitigation plan actually addresses the problem.

This process includes screening the list of risks for duplication and consolidation as appropriate. Program Management errors are not risks and must be corrected before the program moves forward. This screening considers program-level ramifications and should ensure that program integration risks are adequately covered. A Risk Worksheet (Figure 4.10-6) can be used to document newly identified potential risks.

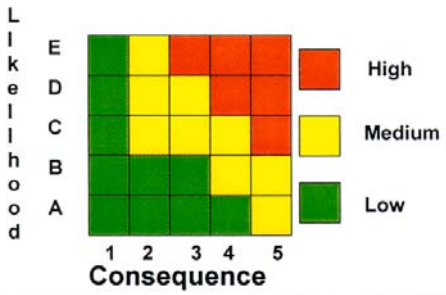
## FAA Risk Worksheet

**Program/Project Title**_____ **Seq. #:** _____

**Submitted by:** _____ **Date:** _____

| **1 Risk:** | **2Point of Contact** |
|---|---|
| **3 Source and Root Cause:** | |

| **4 Risk Assessment** | | | **Rationale** |
|---|---|---|---|
| o **Technical** | o **Schedule** | o **Cost** | |
| **Likelihood** | A B C D E | | |
| **Consequence** | 1 2 3 4 5 | | |




**Consequence Definition:**

**Risk Resolution Date:**

| **5** | **Mitigation Options** | Description | New Risk Level if Implemented |
|---|---|---|---|
| ☐ | **Avoidance** | | H M L |
| ☐ | **Transfer** | | H M L |
| ☐ | **Control** | | H M L |
| ☐ | **Assumption** | | H M L |
| ☐ | **Research & Knowledge** | | H M L |

226

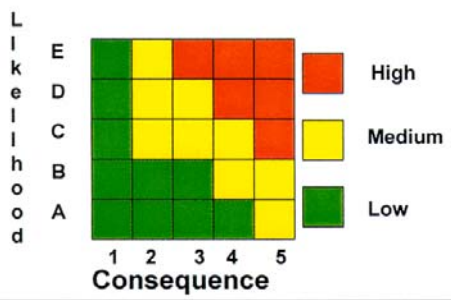
**FAA Risk Worksheet**

**Program/Project Title**_____ **Seq. #:** _____

**Submitted by:** _____ **Date:** _____

| | |
|---|---|
| **1 Risk:** | **2Point of Contact** |
| **3 Source and Root Cause:** | |

| **4** | **Risk Assessment** | | **Rationale** |
|---|---|---|---|
| o **Technical** | o **Schedule** | o **Cost** | |
| **Likelihood** | A B C D E | | |
| **Consequence** | 1 2 3 4 5 | | |




**Consequence Definition:**

**Risk Resolution Date:**

| **5** | **Mitigation Options** | Description | New Risk Level if Implemented |
|---|---|---|---|
| ☐ | **Avoidance** | | H M L |
| ☐ | **Transfer** | | H M L |
| ☐ | **Control** | | H M L |
| ☐ | **Assumption** | | H M L |
| ☐ | **Research & Knowledge** | | H M L |

**Figure 4.10-6.**

### 4.10.3.2.Task 2: Analyze and Assess Impacts of Risk (Satisfies iCMM BP 13.03 criteria)

Risk analysis assesses each component of an identified risk: (1) the likelihood that the risk will occur and (2) the consequence to the program should it occur, as depicted in Figure 4.10-7. The basic tool used for qualitative risk analysis is the risk template, which contains a set of definitions to be used to evaluate the likelihood and consequence of a particular risk. The set of templates that a program uses may change over time as new templates are added or existing

templates are changed, combined, or eliminated. The program may choose to use program-unique templates (only if the technical elements of the program demand it), which are based on program or stakeholder requirements, provided rationale is given. However, modification of templates limits the ability to "roll-up" risks to a higher program level, and, as such, a mechanism must be developed to correlate risks developed through modified templates to the risks developed with the standard FAA templates. The program/project is responsible for the choice, coordination, and control of the templates used on the program. These decisions are contained in the Risk Management Plan section of the IPP (see Section 4.2.2.2 in Integrated Technical Planning (Section 4.2)).

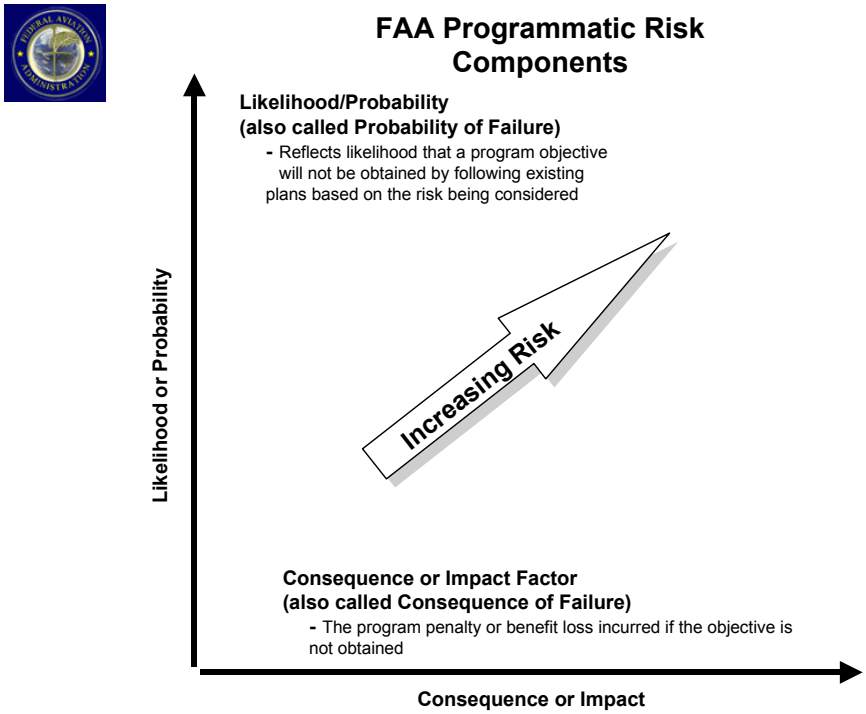



**Figure 4.10-7.**

#### 4.10.3.2.1. Likelihood (Probability) Determination

A likelihood (probability) template is developed that applies to the specific risk/program under analysis. A new template is developed and documented if none of the existing program templates are found to be applicable. This action must be coordinated within the program/project using the criteria of the Risk Management Plan. Correlation of the new templates to the standard FAA templates in this manual must be established. For interviews, a Numerical Interpretation of Likelihood Descriptors matrix (Figures 4.10-8, 4.10-8A, and 4.10-8B) can be used to translate words into numbers. The words are placed in random order on Figure 4.10-8 such that the form can be used to generate a sample of stakeholder estimates of probability. The terms have been placed in an ordered sequence in Figure 4.10-8A based on an FAA early survey, and actual values from that survey are provided in Figure 4.10-8B. A sample of a typical risk likelihood template is shown in Figure 4.10-9. Figure 4.10-10 provides the FAA definitions of the risk likelihood levels.

260

## Numerical Interpretation of FAA Likelihood Descriptors

### *(Exercise)*

**Statement**

| | 0 | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| We doubt | | | | | | | | | | | |
| We believe | | | | | | | | | | | |
| Very good chance | | | | | | | | | | | |
| Not likely | | | | | | | | | | | |
| Probably not | | | | | | | | | | | |
| Probably | | | | | | | | | | | |
| Probable | | | | | | | | | | | |
| Low Likelihood | | | | | | | | | | | |
| Likely | | | | | | | | | | | |
| Improbable | | | | | | | | | | | |
| Highly unlikely | | | | | | | | | | | |
| Highly likely | | | | | | | | | | | |
| Chances are slight | | | | | | | | | | | |
| Better than even | | | | | | | | | | | |
| Almost no chance | | | | | | | | | | | |
| Near certainty | | | | | | | | | | | |

**Probability (%)**

261 **Figure 4.10-8.**

## Numerical Interpretation of FAA Likelihood Descriptors

### *(Exercise - 1)*

**Statement**

| | 0 | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Near certainty | | | | | | | | | | | |
| Highly likely | | | | | | | | | | | |
| Likely | | | | | | | | | | | |
| Very good chance | | | | | | | | | | | |
| Probable | | | | | | | | | | | |
| Probably | | | | | | | | | | | |
| We believe | | | | | | | | | | | |
| Better than even | | | | | | | | | | | |
| Probably not | | | | | | | | | | | |
| We doubt | | | | | | | | | | | |
| Low Likelihood | | | | | | | | | | | |
| Improbable | | | | | | | | | | | |
| Not Likely | | | | | | | | | | | |
| Chances are slight | | | | | | | | | | | |
| Highly unlikely | | | | | | | | | | | |
| Almost no chance | | | | | | | | | | | |

**Probability (%)**

262 **Figure 4.10-8A**

263

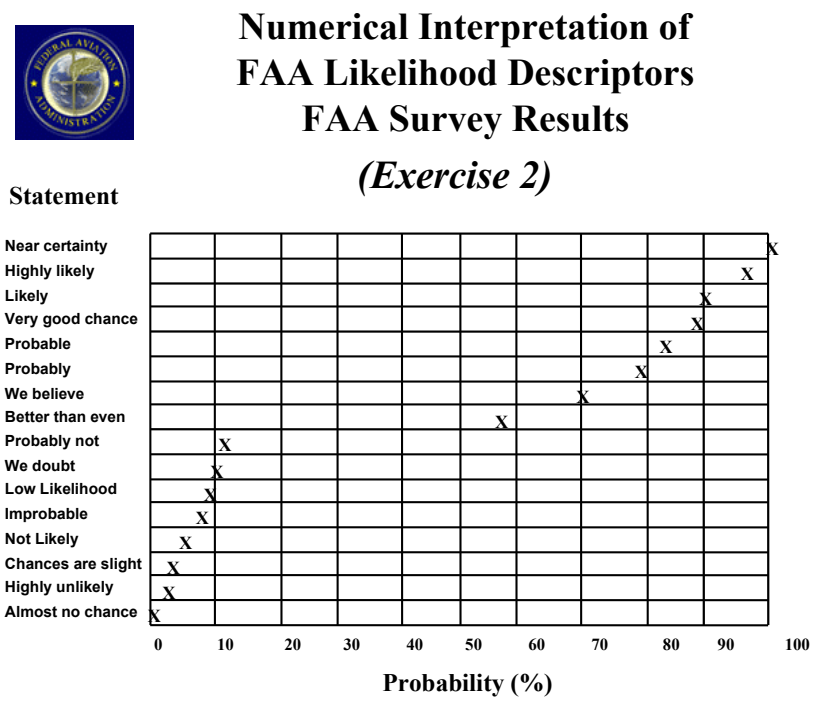



**Figure 4.10-8B.**

## Typical Risk Likelihood Templates

| Risk | Template | | Likelihood | | | | |
|---|---|---|---|---|---|---|---|
| **Category** | **Nbr.** | **Name** | **Near Certainty (E)** | **Highly Likely (D)** | **Likely (C)** | **Low Likelihood (B)** | **Not Likely (A)** |
| Complexity | 1 | Component Integration | Complex component interface based on new design | Complex component interfaces based on existing system | Typical component interfaces based on new design | Typical component interfaces based on existing design | No component interfaces or off-the-shelf |
| Dependency | 2 | Technology Dependence | Dependent on new technologies that are not yet funded | Dependent on new technologies that are in development | Dependent on innovative use of existing technologies | Minor modification of existing system | Off-the-shelf design |
| Experience | 3 | Key Personnel | Expertise not assigned | Core expertise only assigned to program | Core expertise assigned with other personnel available but not yet assigned | All required expertise assigned | All required expertise assigned with established refresher training program |
| Process Control | 4 | Processes | Processes are informal | Processes are partially documented and approved | Processes are largely documented and approved | Processes are in place and approved | Using standard processes with documented continuous improvement process |
| Maturity | 5 | Design Concept | New concept requires significant development | Proof of concept has been demonstrated | Similar concept exists on another program; able to meet requirements by analysis | Similar concept exists on another program; data is available showing compliance with requirements | Fully developed design that meets requirements |

**Figure 4.10-9.**

#### 4.10.3.2.2. Consequence Determinations

Another set of templates is used to evaluate consequence/impact to the program should the risk materialize. As was the case with likelihood templates, if none of the existing program consequence templates are found to be applicable to a particular risk, new templates may be developed and documented. Correlation of the new templates to the standard FAA templates in this manual must be established. Consequence templates are shown for three areas of program impact: technical (Figure 4.10-11), schedule (Figure 4.10-12), and cost (Figure 4.10-13). It must be remembered that each of these results in a programmatic risk, which threatens the benefits of a program and may also have interdependency impacts. All NAS programs are developed to provide benefit(s) to the system. Programmatic risk ultimately reflects in impacts to benefit(s). All benefit losses are derived from losses in either technical, schedule, or cost risks. This is a significant part of the risk consequence that must be defined. The cost/benefit analysis must be reexamined as a result of cost and/or benefit impacts to provide the information needed to make program decisions.

# FAA Programmatic Risk Likelihood Definitions

***What is the likelihood the risk will happen?***

A. Low: Your approach and processes will effectively avoid or mitigate this risk based on standard practices.

*The chance of a negative outcome based on existing plans is not likely. This likelihood level assessment should be based on evidence or previous experience and not on subjective confidence. This assessment level requires the approach and processes to be well understood and documented. Little or no management oversight will be required.*

B. Minor: Your approach and processes have usually mitigated this type of risk with minimal oversight in similar cases.

*There is a low likelihood but reasonable probability that a negative outcome is possible. Present plans include adequate margins (technical, schedule, or cost) to handle typical problems. This assessment level requires the approach and processes to be well understood and documented. Limited management oversight will be required.*

C. Moderate: Your approach and processes may mitigate this risk, but workarounds will be required.

*A negative outcome is likely, or the current approach and processes are only partially documented. Alternative plans or methods exist to achieve an acceptable outcome even if the risk is realized. Present plans include adequate margins (technical, schedule, or cost) to implement the workarounds or alternatives to overcome typical problems. Significant management oversight will be required.*

D. Significant: Your approach and processes cannot mitigate this risk, but a different approach might.

*A negative outcome is highly likely to occur, or the current approach and processes are not documented. While alternative plans or methods are believed to exist to achieve an acceptable outcome, there are not adequate margins (technical, schedule, or cost) to implement the workarounds without impacting the program management reserves in performance, schedule, or cost. Significant management involvement is required.*

E. High: Your approach and processes cannot mitigate this type of risk; no known processes or workarounds are available.

*A negative outcome is going to occur with near certainty. No alternative plans or methods have been documented. Alternatively, the risk item has yet to be evaluated adequately to be well understood, so there is a high level of uncertainty about the program success. Urgent management involvement is required.*

**Figure 4.10-10.**

## FAA Technical Consequence Definitions

***Given the risk becomes real, what would be the magnitude of the impact on system performance?***

1. Low: Given that the risk is realized, there would be minimal impact.

   *A successful outcome is not dependent on this issue; the technical performance goals will be met. There would be no impact on the success of the program.*

2. Minor: Given that the risk is realized, there would be a minor performance shortfall but the same approach could be retained.

   *The resulting technical performance would be below the goal but within acceptable limits. There would be no need to change the basic design, process, or approach. There would be no impact on the success of the program.*

3. Moderate: Given that the risk is realized, there would be a moderate performance shortfall but workarounds would be available.

   *The resulting technical performance would be below the goal. The basic design, process, or approach could be retained with only minor changes, and the overall system performance would still be acceptable as a result of workarounds such as the reallocation of functions or performance goals. There would be only a limited impact on the success of the program.*

4. Significant: Given that the risk is realized, the performance would be unacceptable but workarounds would be available.

   *The resulting technical performance would be unacceptably below the goal. The design, process, or approach would require a significant change to achieve an acceptable performance level. Additional workarounds such as the reallocation of functions or performance goals could also be required. The success of the program could be jeopardized.*

5. High: Given that the risk is realized, the performance would be unacceptable with no known workarounds.

   *The resulting technical performance would be unacceptably below the goal. There are no known alternatives or solutions. The success of the program would be in doubt.*

**Figure 4.10-11.**

## FAA Schedule Consequence Definitions

***Given the risk becomes real, what would be the magnitude of the impact on the schedule?***

1. Low: Given that the risk is realized, there would be minimal impact.

   *The program schedule is not dependent on this issue. There would be no impact on the success of the program.*

2. Minor: Given that the risk is realized, additional activities would be required to meet key dates.

   *One or more key dates in the program schedule, but not critical path events, would be jeopardized; there are identified schedule workarounds that would be sufficient to mitigate the schedule impact. There would be no impact on the success of the program.*

3. Moderate: Given that the risk is realized, there would be a minor schedule slip, and one or more need dates would be missed.

   *One or more key need dates in the program schedule, but not critical path events, would be at least one month late; there are identified schedule workarounds that would be sufficient to keep the program critical path from being affected. There would be only a limited impact on the success of the program.*

4. Significant: Given that the risk is realized, the program critical path would be affected.

   *One or more events on the program critical path would be at least one month late. There are identified schedule workarounds that would be sufficient to meet major program milestones. The success of the program could be jeopardized.*

5. High: Given that the risk is realized, a key program milestone cannot be achieved.

   *Completion of a key program milestone would be late, and the success of the program would be in doubt. The slip requires a re-baseline of the program.*

**Figure 4.10-12.**

## FAA Cost Consequence Definitions

***Given the risk becomes real, what would be the magnitude of the impact on cost?***

1. Low: Given that the risk is realized, there would be minimal cost impact.

   *Program cost is not dependent on this issue. There would be no impact on the success of the program.*

2. Minor: Given that the risk is realized, the total costs, operating cost or unit production cost would increase by = 1%.

   *The program costs and/or the production unit cost would increase by = 1%. There would be no impact on the success of the program.*

3. Moderate: Given that the risk is realized, there would be a minor increase in financial need. The program costs, operating cost or unit production cost could increase above 1% up to = 5%.

   *The program costs and/or the production unit cost would increase above 1% to = 5%. There would be only a limited impact on the success of the program.*

4. Significant: Given that the risk is realized, the total costs, operating cost or unit production cost would increase by above 5% to = 10%.

   *The program costs and/or the production unit cost would increase above 5% to = 10%. The success of the program could be jeopardized.*

5. High: Given that the risk is realized, the total costs, operating cost or unit production cost would increase by greater than 10%.

   *The program costs and/or the production unit cost would increase by greater than 10%. The success of the program would be in doubt.*

**Figure 4.10-13.**

#### 4.10.3.2.3. Risk Level Determination

The likelihood and consequence are considered to be independent, but tied to the same event, and are mapped into a risk grid to determine the individual risk level (e.g., high (red), medium (yellow), or low (green)) as shown in Figure 4.10-14. This mapping facilitates the prioritization and trend analyses of risks throughout the life of the program. Use of a color code for each

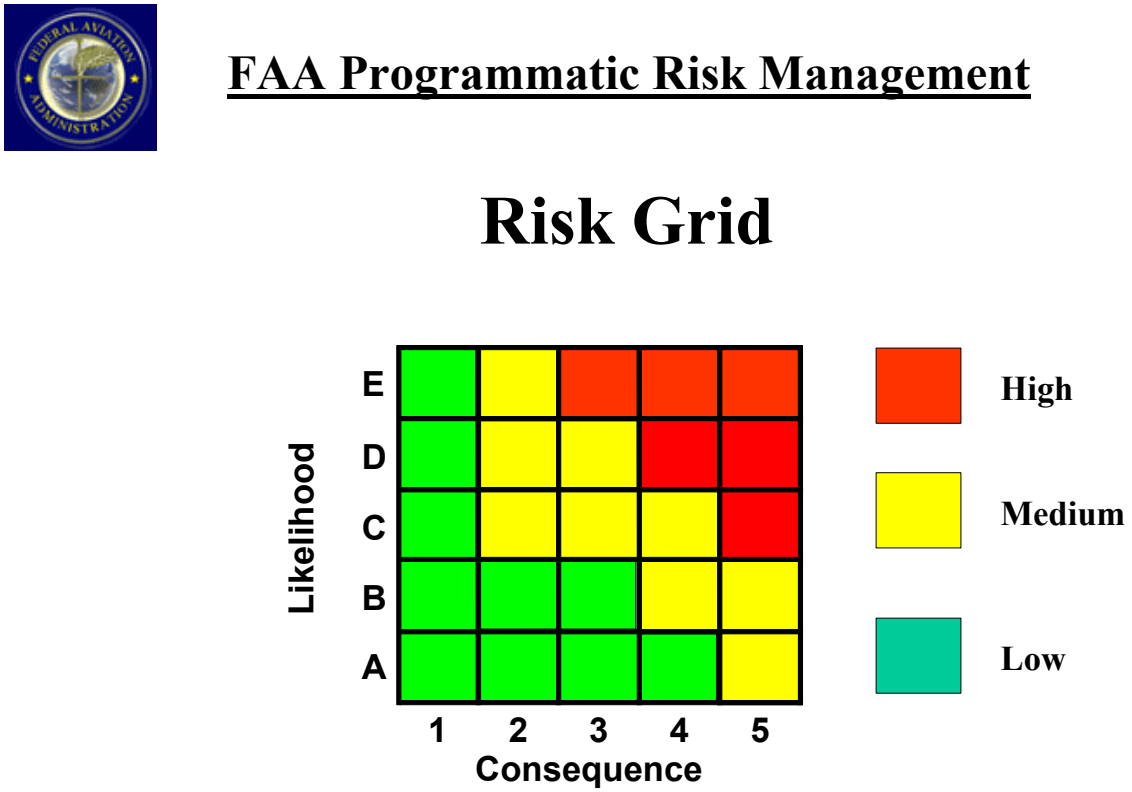



**Figure 4.10-14.**

risk level supports effective communication of program health internally and externally.

Risk level definition high (red) is likely (a high probability) to cause significant disruption of schedule, increase in cost, or degradation of performance. Concerted and continual emphasis and coordination may not be sufficient to overcome major difficulties. Medium (yellow) may cause some disruption of schedule, increase in cost, or degradation of performance. Special emphasis and close coordination will probably be sufficient to overcome difficulties. Low (green) has little potential for disruption of schedule, increase in cost, or degradation of performance. Normal emphasis and coordination will probably be sufficient to overcome difficulties. The threshold for differentiating between high, medium, and low may change from program to program, but not for risk to risk, and should be determined early in the life of the program.

Figure 4.10-15 is a summary of how the consequence and likelihood are consolidated to define the risk level. Each risk must also be evaluated using the GAO (General Accounting Office) process defined in Figure 4.10-16. At this point, it is necessary to make sure that the risk resolution date is documented.

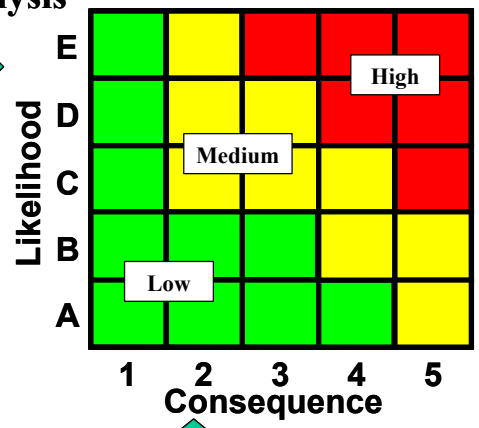
354

# FAA Programmatic Risk Analysis

What is the Likelihood the risk will happen?

Likelihood

| Level | | Approach and Processes |
|---|---|---|
| A | Not Likely | … will effectively avoid or mitigate this risk based on standard practices. |
| B | Low Likelihood | … have usually mitigated this type of risk with minimal oversight in similar cases. |
| C | Likely | … may mitigate this risk, but workarounds will be required. |
| D | Highly Likely | … cannot mitigate this risk, but a different approach might. |
| E | Near Certainty | … cannot mitigate this type of risk; no known processes or workarounds are available. |




Consult non-advocates to support risk analysis

*Always consider Interdependencies and benefits impacts*

Given the risk is realized, what would be the magnitude of the impact?

Consequence

| Level | Technical | Schedule | Cost |
|---|---|---|---|
| 1 | Minimal Impact | Minimal Impact | Minimal Impact |
| 2 | Minor performance shortfall, same approach retained | Additional tasks required, able to meet key dates | Development or acquisition cost increase to ≤ 1% |
| 3 | Moderate performane shortfall, workarounds available | Minor schedule slip, will miss need date without workaround | Development or acquisition cost increase >1% & ≤ 5% |
| 4 | Unacceptable performance but workarounds available | Program critical path impact but workaround availale | Development or acquisition cost increase >5% &≤ 10% |
| 5 | Unacceptable performance and no alternatives exist | No known way to achieve program milestones | Development or acquisition cost increase >10% |

355

356 **Figure 4.10-15.**

357

358

359

360

361

362

363

364

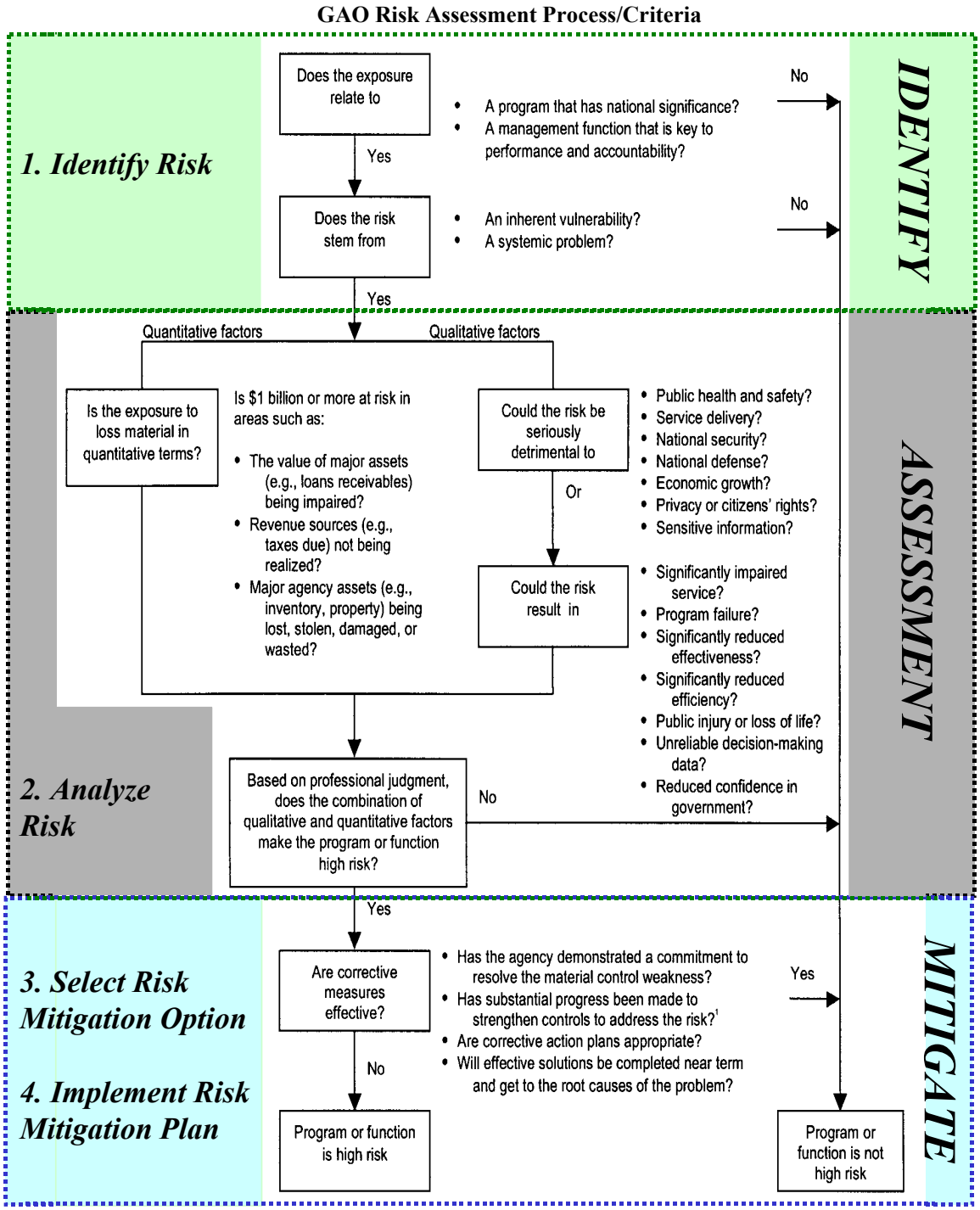
365

366

367

368

# FAA Programmatic Risk Management




**GAO process requires examination of risk and the development of a mitigation effort. Shown is Figure 5 of GAO/OCG-00-12, Page 9. (August/2000).**

**Figure 4.10-16.**

#### 4.10.3.3. Task 3: Select Risk Mitigation Option *(Satisfies iCMM BP 13.05 criteria)*

Risk mitigation requires a conscious management decision to approve, fund, schedule, and implement one or more risk mitigation actions. Risk mitigation plans and mitigation actions are reviewed frequently at major reviews, program reviews, acquisition reviews, and milestone reviews. The objective of the risk mitigation step is to implement appropriate and cost-effective risk mitigation plans to mitigate or eliminate the risks. Appropriate risk mitigation techniques are selected, and mitigation actions are developed, documented, and implemented. The risk mitigation plan/implementation and tracking are the core of risk management.

Risk mitigation actions fall into one, or a combination, of the following categories (Figure 4.10-4):

- Avoidance
- Transfer
- Control
- Assumption
- Research and Knowledge

"Avoidance" is a strategy to avert the potential of occurrence and/or consequence by selecting a different approach or by not participating in the program. This technique may be pursued when multiple designs or programmatic options are available. It is more likely used as the basis for a "Go"/"No-Go" decision at the start of a program. Some examples are selection of state-of-the-practice rather than state-of-the-art technologies and prequalification of suppliers. The avoidance of risk is from the perspective of the overall program/project, which includes the stakeholders, contractors, and execution groups. Thus, an avoidance strategy is one that involves all of the major parties to the program/project and permits a program/project-wide avoidance of the risk.

"Transfer" is a strategy to shift the risk to another area, such as another requirement, an organization, a supplier, or a stakeholder. Examples include reallocating of requirements, securing supplier product warranties, and negotiating of fixed-price contracts with suppliers. Note that at the program level the risk remains. The transfer of the risk is accomplished primarily to optimize, in a sense, the overall program risk and to assign ownership to the party most capable of reducing the risk. It is possible that the risk level will change as a result of the risk transfer.

"Control" is a strategy of developing options and alternatives and taking actions that lower or eliminate the risk. Examples include new concepts, more analysis, redundant systems and/or components, and alternate sources of production.

"Assumption" is simply accepting the likelihood/probability and the consequences/impacts associated with a risk's occurrence. Assumption is usually limited to low risks. This is a program/senior management option, not a program option. FAA practice is to develop mitigation plans for all medium and high risks.

“Research and Knowledge” may mitigate risk through expanding research and experience. Since risk arises from uncertainty and inexperience, it may be possible to effectively mitigate risk simply by enlarging the knowledge pool, leading to reassessment that reduces the likelihood of failure or provides insight into how to lessen the consequences.

At this point, several alternatives for mitigating the risk have been identified and analyzed for selection of the preferred approach. Alternatives include detailed plans for mitigating the risk in

several small, sequential steps; alternative steps; or entirely new (non-baselined) approaches to accomplishing the program. Further, contingency plans are identifiable alternatives that can be implemented if a mitigation plan fails, and the risky event or conditions occur with more serious consequences than anticipated. The mitigation steps are the major milestones of the mitigation plan. Contingency plans need not be extremely detailed.

Trade study techniques can be performed to help select the preferred risk mitigation plan. While the proper criteria and their weights for each analysis are dependent on the risks to be mitigated, the following should be included:

- Does the option mitigate the likelihood or consequence of the risk?
- Does the option fit within program scope?
- Is the option easy to implement?
- Are new risks avoided or introduced?
- What is the cost of mitigation?
- What is the schedule for mitigation?

The risk level is the first criterion used to determine the need for a risk mitigation plan. Program risks that fall into the medium or high categories require risk mitigation plans. Risks that are assessed as low typically do not require mitigation plans but may have certain aspects that would be prudent to monitor. If this is the case, risk mitigation plans may be formally or informally implemented for these low-risk issues based on the specific Risk Management Plan for a program.

It is essential that those responsible for plan implementation have a thorough understanding of the risk to be mitigated. This can be accomplished with a good summary statement of the risk. It should include descriptions of the future event or condition, which confirms trouble for the program; the root cause(s) of the event outcome or conditions; and the specific effects to the program if the event or conditions occur with negative consequences. The risk should not be stated in terms of its mitigation plan.

The status should also include a summary of risk mitigation efforts that references more detailed documentation. A Risk Mitigation Plan Summary (Figure 4.10-17) is used to report the analysis and actions on an individual risk.

The risk mitigation plan documents the specific steps to be implemented, the sequence in which they will be implemented, and the points in time at which they will be implemented. Development of a risk mitigation plan includes an assessment of the expected outcome following implementation. The same method initially used to assess the risk, such as risk templates, should be used to provide a forecast of the risk level after completion of each action of the risk mitigation plan. The expected impact of each mitigation event on risk level can be projected using a format similar to that of Figure 4.10-18 (a "waterfall chart").

# Risk Mitigation Plan Summary

Page ____ of ____

Status: _____

| Risk Title | Program/Project<br>Project Lead |
|---|---|

Description of Risk

Statement of Root Cause

Consequence if Realized

*Risk Type:* T S C

*Likelihood* ___

*Consequence* ___

*Comments:*

*Risk Resolution Date:*

*Place X in One Cell*

| Likelihood | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| E | | | | | |
| D | | | | | |
| C | | | | | |
| B | | | | | |
| A | | | | | |

Consequence

**Risk Mitigation**

| | 2002 | | | | | | | | | | | | 2003 | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | J | F | M | A | M | J | J | A | S | O | N | D | J | F | M | A | M | J | J | A | S | O | N | D |
| HIGH | | | | | | | | | | | | | | | | | | | | | | | | |
| MEDIUM | | | | | | | | | | | | | | | | | | | | | | | | |
| LOW | | | | | | | | | | | | | | | | | | | | | | | | |
| | J | F | M | A | M | J | J | A | S | O | N | D | J | F | M | A | M | J | J | A | S | O | N | D |

**Figure 4.10-17.**

## Risk Mitigation Waterfall Schedule

***Sample***

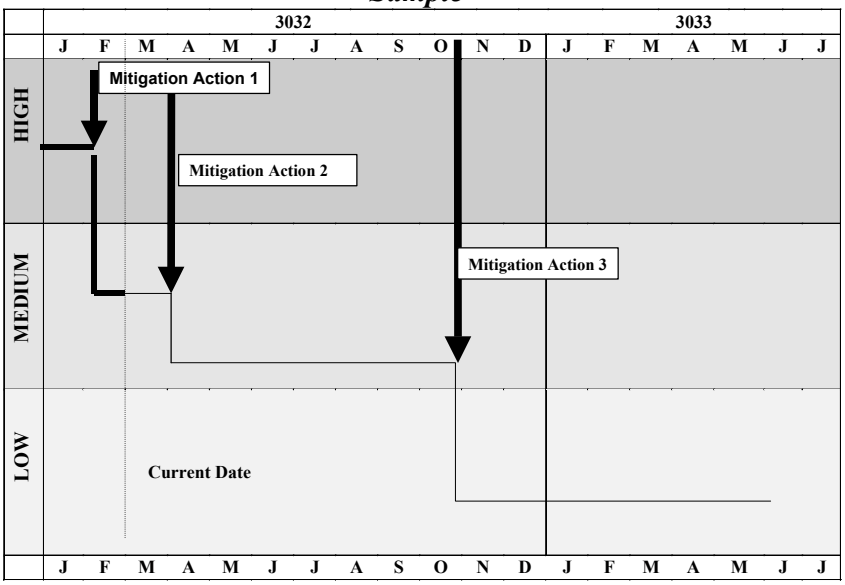



**Figure 4.10-18.**

The risk management plan becomes the basis for monitoring the success in mitigating each risk. The plan includes, but is not limited to, the following:

- A description of the risk for which the plan applies
- The mitigation approaches that detail the specific actions that are planned to reduce the risk or eliminate it. These actions should be event-based, integrated into a schedule, and have associated with each of them:
  - The decision point or trigger, past or future, that initiates the action or group of actions
  - The resources required to execute the actions (including personnel, capital equipment, facilities, procured equipment)
  - The measures of success to be used for the planned actions or group of actions
  - The fallback options (if any)
  - The planned completion dates of the actions
- Risk mitigation metrics
- The Risk Worksheet (Figure 4.10-6)
- Risk Mitigation Waterfall Schedule (Figure 4.10-18)
- The initial Risk Mitigation Plan Summary (Figure 4.10-17)

A risk mitigation plan should be evaluated to determine its effectiveness. This analysis is performed in the same manner as initial analysis for the risk. The set of templates used for analysis of the risk can also be used to determine the mitigation in the risk level following the completion of each major action or group of actions. The regular reassessment of the risk and performance to plan using a fixed set of criteria provides a consistent analysis of the impact to the program.

When a risk mitigation plan has been prepared, it is reviewed and approved by stakeholders based on criteria defined in the Risk Management Plan.

#### 4.10.3.4.Task 4: Implement Risk Mitigation Plan *(Satisfies iCMM BP 13.05 criteria)*

Once a mitigation plan option is selected, the Risk Mitigation Plan Summary chart (Figure 4.10-17) is used as a means of reporting progress in mitigating risks. An example is provided in Figure 4.10-17A. The black, broken line on the stairsteps indicates the planned level of risk versus time; the planned events for reducing the risk are shown as arrows with text boxes. Each major event in the mitigation plan is identified together with how that event mitigates the risk and to what level.

Risk mitigation plans are implemented with the actions and milestones being incorporated in the planning, scheduling, budgeting, and cost accounting systems used on the program. Incorporation of the risk mitigation plans and milestones into these program processes and systems ensures that the risk and its mitigation plans will be monitored and tracked until the risk is eliminated or the risk requires program modification. Risk mitigation plans can be documented starting with the Risk Worksheet shown in Figure 4.10-6 and a Risk Mitigation Waterfall Schedule shown in Figure 4.10-18. All mitigation activities are shared with and communicated to all stakeholders.

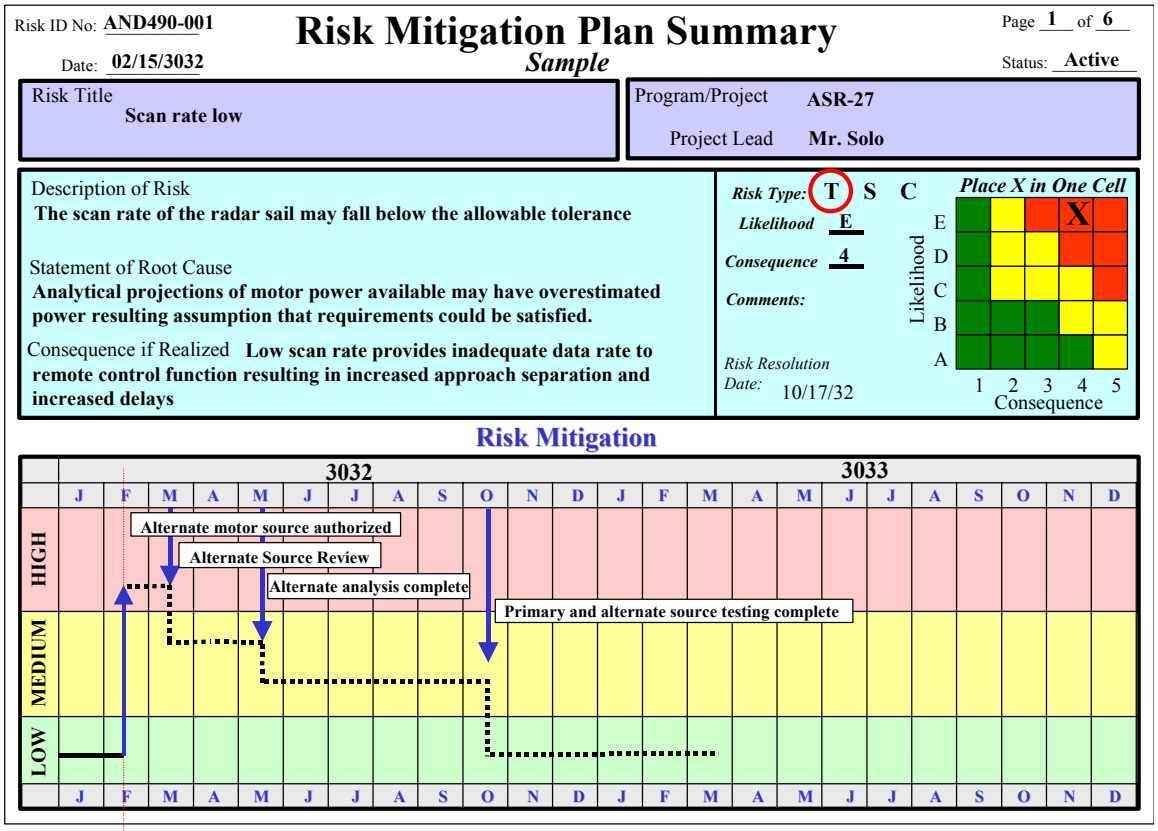



**Figure 4.10-17A.**

### 4.10.3.5.Task 5: Monitor and Track Risks *(Satisfies iCMM PA 14 criteria)*

Reassessment of currently managed risks is done on a periodic and event basis to reflect current status of the issues as well as to identify and quantify new and emerging issues. New, potential risks to the program may be identified at any time. Newly identified risks are analyzed using the same steps described in Section 4.10.3.2.

Steps in the risk-tracking process focus on providing the execution teams, interdependent activities, and program management with program risk trends and status. Actual performance of the planned mitigation actions is compared to the expected performance. The bold line on the Risk Mitigation Plan Summary “waterfall area” (Figure 4.10-17B) indicates progress made to date on the mitigation plan. Detailed cost and schedule tracking is done as part of the program schedule and cost- tracking system.

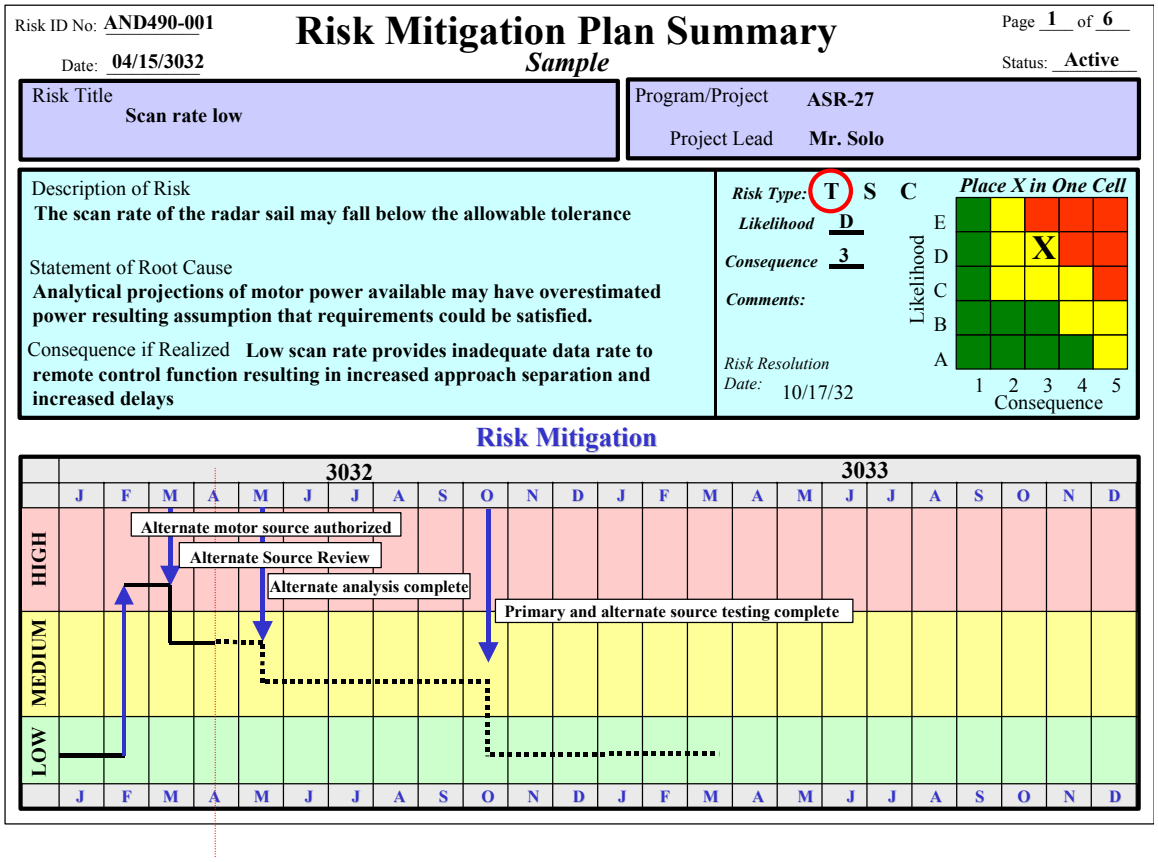



**Figure 4.10-17B.**

### 4.10.3.5.1. Documentation of Risk Analyses

Following the initial risk assessment and each subsequent assessment, the analysis results are documented. The form shown in Figure 4.10-6 is a recommended Risk Worksheet used for documentation. The risk mitigation plan area can be left blank until the risk mitigation plans are completed. When a risk item has been identified, it is incorporated into the risk register (see Figure 4.10-19 and example in Figure 4.10-19A), which includes the program name, revision date, and page sequence.

The risk register is a listing of risk information associated with achieving program objectives. In addition to the risk registers created and maintained by each project, a single composite register of all interdependency risk items must be developed for the program. These registers will be consistently used to monitor and track overall risk status within team meetings, program management reviews, and major program reviews. Immediately following the identification and analysis of a new medium or high risk or when a significant change occurs in a previously identified risk, changes must be incorporated in the register and other documents and the new risk identified to stakeholders. The distribution list will be established by a program's Risk Management Plan. Computer database systems may be needed to manage these outputs for large programs. Smaller programs will often be able to use desktop computer techniques.

# FAA Program Risk Register

**DATE** mm/dd/yyyy
**Page** _____ **of** ____
**Revision**

| Risk # | Likelihood | Consequence | Risk Level/Change | Risk Item/Consequence | Next Milestone date | Risk Resolution Date | Mitigation Status | Risk Type |
|---|---|---|---|---|---|---|---|---|
| 1 | | | | | | | | |
| 2 | | | | | | | | |
| 3 | | | | | | | | |

**Consequence Key**:
1= Minimal impact
2= Minor, able to maintain same approach
3= Moderate shortfalls, workaround exists
4= Unacceptable, workaround exists
5= Unacceptable, no alternative exists

**Risk Level:**
**H** - High **M** - Medium **L** - Low
→= same as last report ↑= up from last report
↓= down from last report

**Risk Type:**
Ⓣ - Technical
Ⓢ - Schedule
Ⓒ - Cost

**Figure 4.10-19.**

# FAA Program Risk Register

*(Example)*

**DATE** 04/15/3032
**Page** __1__ **of** __1__
**Revision**

| Risk # | Likelihood | Consequence | Risk Level/Change | Risk Item/Consequence | Next Milestone date | Risk Resolution Date | Mitigation Status | Risk Type |
|---|---|---|---|---|---|---|---|---|
| 1 | D | 3 | M ↓ | Sweep rate low/ delays-benefits loss | 30320515 | 30321017 | On Track | T |
| 2 | | | | | | | | |
| 3 | | | | | | | | |

**Consequence Key**:
1= Minimal impact
2= Minor, able to maintain same approach
3= Moderate shortfalls, workaround exists
4= Unacceptable, workaround exists
5= Unacceptable, no alternative exists

**Risk Level:**
**H** - High **M** - Medium **L** - Low
→= same as last report ↑= up from last report
↓= down from last report

**Risk Type:**
Ⓣ - Technical
Ⓢ - Schedule
Ⓒ - Cost

**Figure 4.10-19A.**

At minimum, the following information shall be included in the risk register:

**4.10.3.5.1.1. Risk Register Identification and Creation/Update Date**

This is the name of the program risk item. Include the root cause of the risk in this section.

**4.10.3.5.1.1.1. Risk Identification Number**

This number is the code that identifies a unique sequence.

**4.10.3.5.1.1.2. Likelihood**

This is a figure-of-merit indicating the relative likelihood/probability that the identified risk will actually occur (Likelihood Template, Figures 4.10-9 and 4.10-10).

**4.10.3.5.1.1.3. Consequence**

This is a figure-of-merit indicating the relative severity of consequences/impacts that could result if the identified risk did occur (Consequences Templates, Figures 4.10-11, 4.10-12, and 4.10-13, for examples).

**4.10.3.5.1.1.4. Risk Level/Change**

This is a single letter indicating the assessed risk of an item as high, medium, or low (H, M, L) or, red, yellow, or green (R, Y, G), respectively. An arrow that indicates the direction that the risk has moved since the last revision to the risk register demonstrates the risk change.

**4.10.3.5.1.1.5. Risk Consequence Description**

This is a brief, well-stated description of the risk's negative consequences is listed.

**4.10.3.5.1.1.6. Next Milestone Date**

This date is the projected date on which the risk level converts to lower risk. This will be traceable to the Risk Mitigation Plan Summary (See Figure 4.10-17).

**4.10.3.5.1.1.7. Risk Resolution Date**

This is the date of the event that will occur that will either make the risk a real part of the program or eliminate the need to track the risk. Early in the program, it may be difficult to predict an exact date, but a general timeframe needs to be developed. As the program matures, date resolution should occur. These dates should be reviewed regularly and be on the program master schedule.

**4.10.3.5.1.1.8. Mitigation Status**

The currently planned mitigation actions are defined.

**4.10.3.5.1.1.9. Risk Type**

The risk type designates if the risk is a cost risk, a schedule risk, or a technical risk (see paragraph 4.10.3.1.1).

**4.10.3.5.2. Risk Mitigation Plan Status**

The teams regularly update and report the status of the risk mitigation plan for each risk being tracked. Actions are initiated as required where mitigation plan activities are not being accomplished. The risk status is also reviewed with program management on a regular basis. A sample of a brief summary of all risks for a particular program (or team) is shown in a Program Risk Summary (Figures 4.10-20 and 4.10-20A) for use depending on program size. Risk mitigation progress is shown in a Program Risk Mitigation Progress chart (Figures 4.10-21 and 4.10-21A).

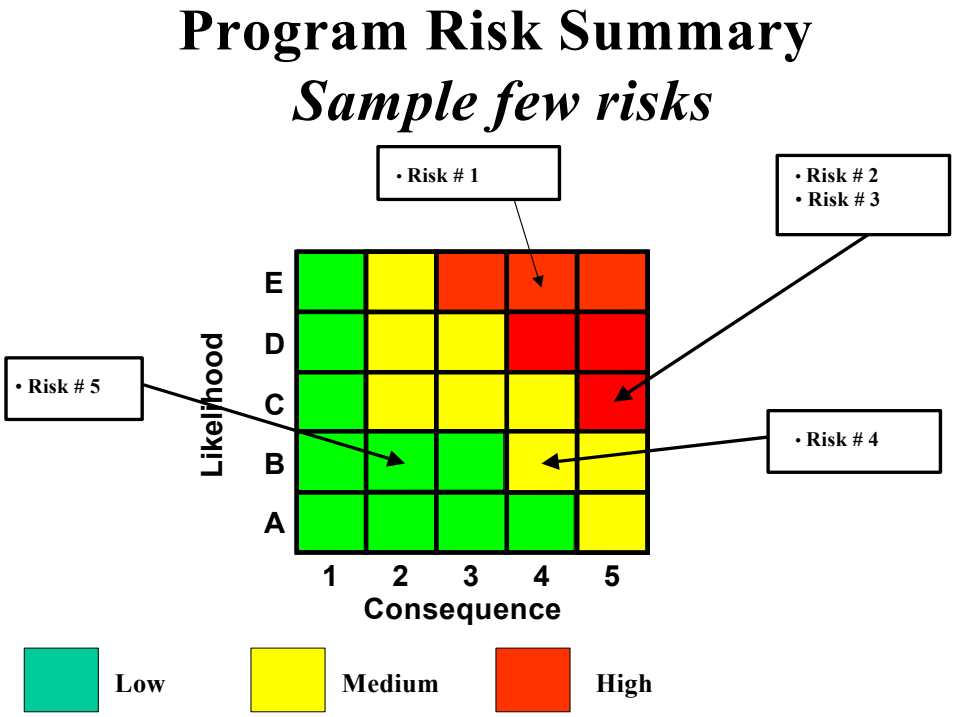



Figure 4.10-20.

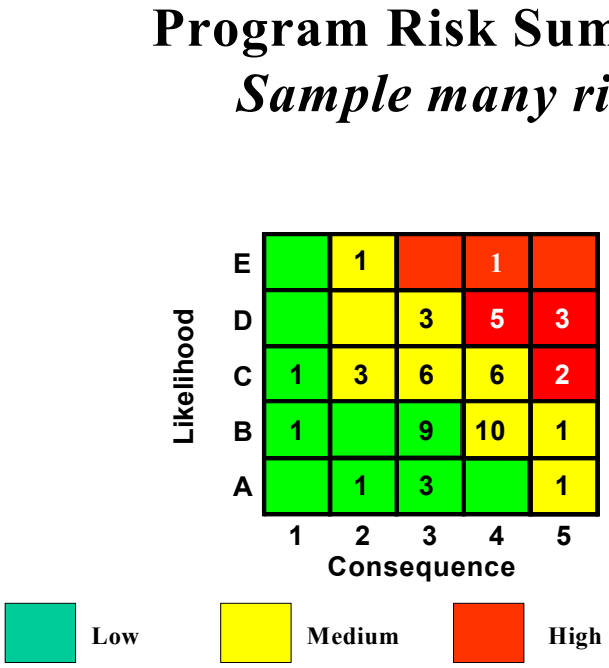



Figure 4.10-20A.

# *Sample*
# Program Risk Mitigation Progress

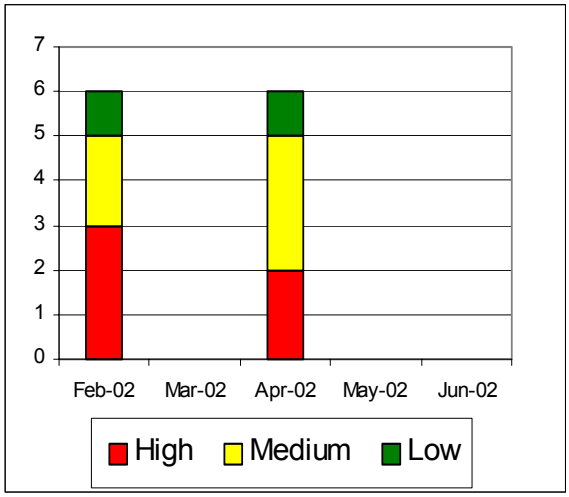
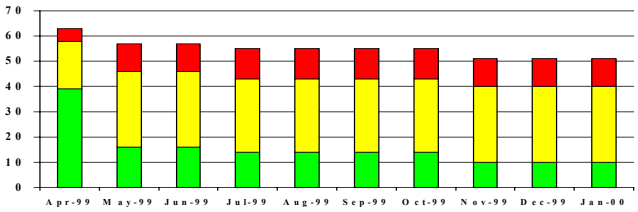



**Figure 4.10-21.**

## FAA Programmatic Risk Management
## NEXCOM Example

# Summary of Risks

Last risk assessment conducted in January 2000.




**Impact:** A subjective ranking from 1 to 5.

1 = very low
2 = low
3 = moderate
4 = high
5 = very high

**Probability**: An estimate that the risk will occur during the impact time. A subjective ranking from 1% (extremely unlikely) to 99% (almost certain)

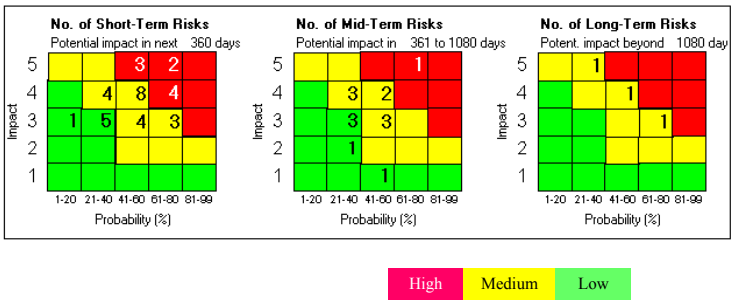



**Figure 4.10-21A.**

### 4.10.4. Outputs of Risk Management (Satisfies iCMM Artifacts criteria)

The five major outputs of this process are:

- Program Risk Register (Figure 4.10-19)
- Risk Mitigation Plans (Section 4.10.3.3)
- Program Risk Summary (Figures 4.10-20 and 4.10-20A)
- Program Risk Mitigation Progress Chart (Figures 4.10-21 and 4.10-21A)
- Risk Mitigation Plan Summary (Figure 4.10-17)

The Program Risk Summary, the Risk Mitigation Plan Summary, and the Program Risk Mitigation Progress charts should be briefed at all regular program reviews. Management decisions are based on the above information. A complete status of a given risk should be briefed when the risk is identified and immediately following the risk resolution date.

The Risk Mitigation Plan should be considered an appendix to the IPP and Acquisition Program Baseline. It must be handled as an integral part of program effort.

### 4.10.5. Risk Management Tools

The tools needed to implement this process include:

- Likelihood and consequence templates tailored for the program
- A means to document the results of the process and manage the outputs (databases, spreadsheets, word processors, etc.)
- A means to communicate results across a program (electronic mail, servers, etc.).

Database tools such as “Risk Radar” –(a tool free to the government (http://www.spmn.com))) can be used to generate many of the risk work products (see Section 4.10-7). Analytic tools can be used for probabilistic analysis of schedule uncertainty or technical uncertainty. Critical Path Analysis tools can be used with the program's Integrated Master Schedule to regularly evaluate schedule risk. In a similar fashion, commercial applications (e.g., @RISK) can be applied to technical parameters (such as weight, latency, power, computer throughput) to establish confidence ranges. Results from these probabilistic analyses can support the overall risk analysis task of establishing a likelihood of occurrence. Further details on the use of probabilistic analysis can be found in textbooks and technical papers that cover statistical analysis for risk management.

### 4.10.6. Risk Management Process Metrics (Satisfies iCMM PA 18 criteria)

Metrics for the Risk Management process can include:

- Total risks identified; total high risks, total medium risks
- Percent of risks (Medium and High) with approved mitigation plans
- Percent of mitigation actions that provide expected reduction in risk level
- Percent of total risks reduced to Low in the planned amount of time
- Number of unplanned/unidentified negative consequence events
- Risk mitigation schedule and budget compliance.

Metrics for each program will be defined as part of the Risk Management Plan.

### 4.10.7. References


1. AFMC. "Acquisition Risk Management Guide." USAF Pamphlet 63-101.
2. ANSI/EIA-632-1998. p. 11, 13, 14, 17, 30, 33-4, 45, 49, 52, 67, 75, 77, 81, 96, 109, Requirement 24.
3. Blanchard & Fabrycky. "Systems Engineering and Analysis" (3rd ed.). Prentice Hall, 1998, p. 657-661.
4. Conrow, Edmund H. "Effective Risk Management." American Institute of Aeronautics and Astronautics, Inc., 2000.
5. DoD 4245.7-M. "Transition from Development to Production." Chapter 9-8.
6. DOT H 1350.252. Departmental Guide to Risk Management Planning. May 1999.
7. DSMC. "Risk Management Guide." Available: http://www.dsmc.dsm.mil/pubs/gdbks/risk_management.htm
8. DSMC. "Systems Engineering Management Guide." Chapter 15, 1990.
9. EIA 731.1, FA 2.5, pp. 66-71
10. FAA AMS Paragraph 2.9.14.
11. FAA Acquisition Program Baseline Template.
12. FAA Orders 1900.47, 1050, 1600, 3900, and 1370.82.
13. FAA-P-1810. "Acquisition and Program Management Risk Management Guidance." December 1996.
14. FAA Report No. WP-59-FA7N1-97-2. "Risk Assessment Guidelines for the Investment Analysis Process." July 1999.
15. FAA System Engineering Manual, Section 4.10.
16. GAO/AIMD (Accounting and Information Management Division)-10.1.13. "Assessing Risks and Returns: A Guide for Evaluating Federal Agencies' IT Investment Decision-making." February 1997.
17. GAO/AIMD-99-139. "Information Security Risk Assessment." August 1999.
18. GAO/OCG-00-12. "Determining Performance and Accountability Challenges and High Risks." August 2000.
19. Grady, Jeffery O. "Requirements Analysis." McGraw-Hill, 1993, p. 462-465.
20. Grady, Jeffery O. "System Engineering Planning and Enterprise Identity." CRC Press, 1995, p. 168-177.
21. Grady, Jeffery O. "System Integration." CRC Press, 1994, p. 149.
22. NASA. "NASA Systems Engineering Handbook." 1995, p. 37-44.
23. PMBOK Chapter 11.
24. Ross, J. F. "Living Dangerously." Perseus Publishing, March 2000.
25. US Navy Best Practices. NAVSO P-8871.

# Section 4.11 Configuration Management

## Reserved

## 4.12 Validation and Verification

Validation and Verification is the System Engineering (SE) process that confirms that system requirements are correct and satisfied (Figure 4.12-1). The Validation process confirms that the right system is being built (i.e., that the system requirements are unambiguous, correct, complete, consistent, operationally and technically feasible, and verifiable). The Verification process ensures that the design solution has met the system requirements and that the system is ready for use in the operational environment for which it is intended. This section describes the Validation and Verification process, including the inputs, outputs, and specific tasks of Validation and Verification.

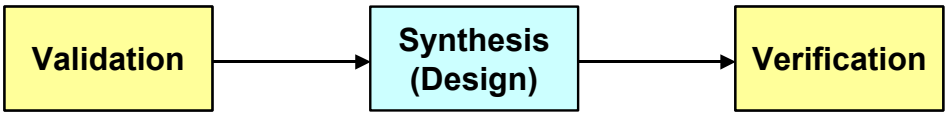



**Figure 4.12-1. Validation and Verification's Role in System Development Process**

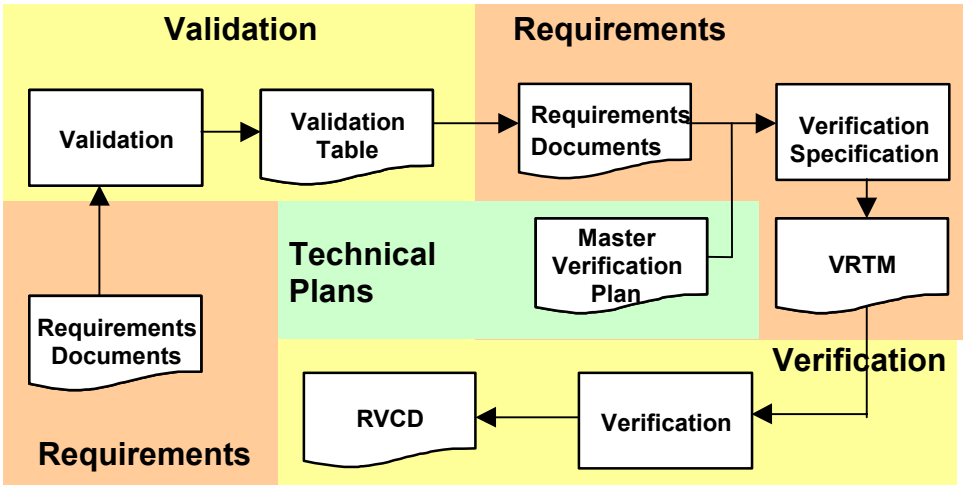



**Figure 4.12-2. Validation and Verification Activities**

The Validation and Verification activities, illustrated in Figure 4.12-2, are summarized below:

- Requirements feed Validation. During Validation activities, a Validation Table is developed that is included in a Validation Report when completed. The Validation Report is an input to the requirements document. The Validation Table becomes the basis for later Verification activities.
- At the same time, work begins on Verification planning and is documented in a "living" joint SE and Test and Evaluation (T&E) Master Verification Plan (MVP)(described and developed under Integrated Technical Planning (Section 4.2)).

- After Verification planning is completed, a specification/approach for verifying each requirement is developed in Requirements Management (Section 4.3) and documented for each requirement in the Validation Table. This update to the Validation Table transforms it to a Verification Requirements Traceability Matrix (VRTM), which becomes the foundation for the next activity and is included in the MVP as an update.
- After Verification activities are performed, the VRTM is updated with evidence of completion of activities. Using the updated VRTM, the Verification team develops the Requirements Verification Compliance Document (RVCD) to record completion of the Verification effort. The RVCD also identifies system compliance or noncompliance with the set of requirements used for the Verification activities. Program management uses this information for the Risk Management process (Section 4.10).

### 4.12.1 Validation

As stated earlier, the Validation process (Figure 4.12-3) confirms that the right system is being built (i.e., that the system requirements are unambiguous, correct, complete, consistent, operationally and technically feasible, and verifiable). The process is conducted in order to demonstrate that the requirements for a system are clearly understood and that it is possible to satisfy the requirements through design work using available state-of-the-art technology, funding, and schedule.

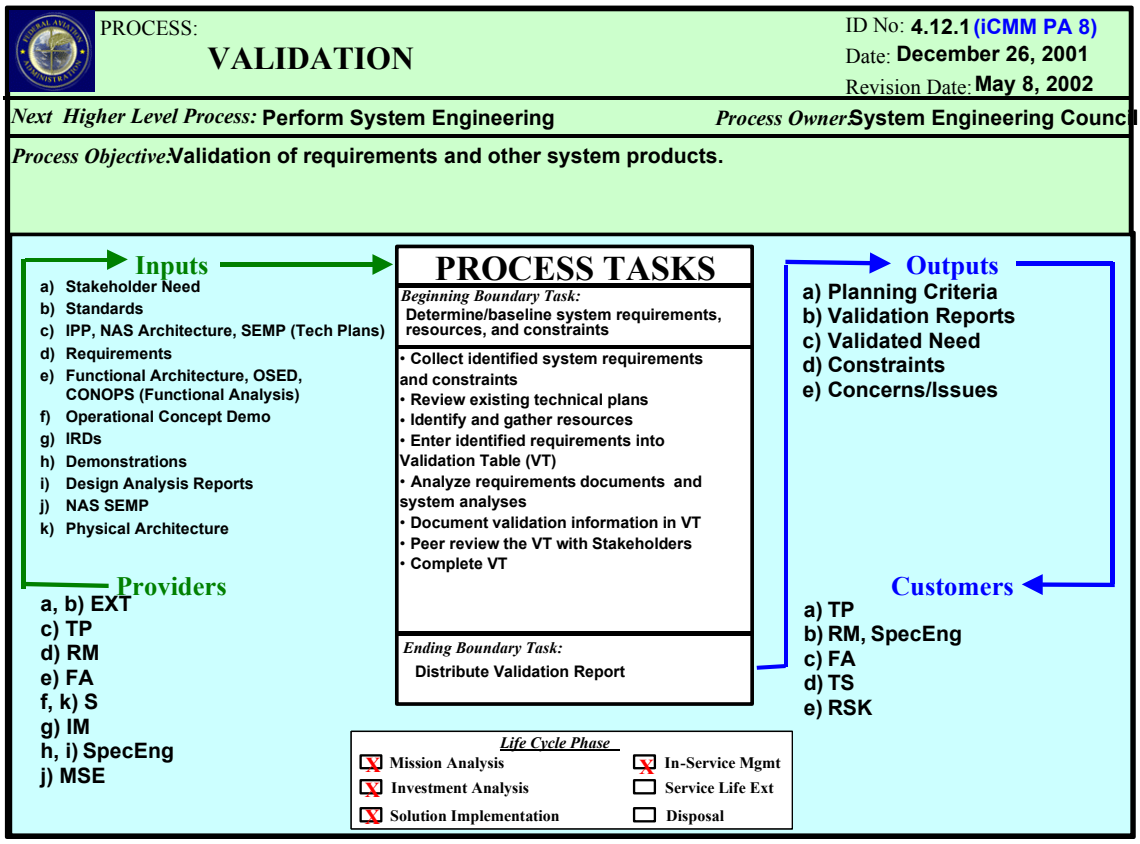



**Figure 4.12-3. Validation Process-Based Management Chart**

The Validation process is repeated incrementally at all stages of requirements development to ensure that the design at all levels is consistent with the intended mission. Validation follows the development of system requirements. Since these requirements are hierarchical in nature and developed in increasing detail as the lifecycle progresses, Validation is a staged process (Figure 4.12-4). Thus, as each level of requirements is developed, the requirements at that level undergo Validation, after which each validated requirement undergoes Verification.

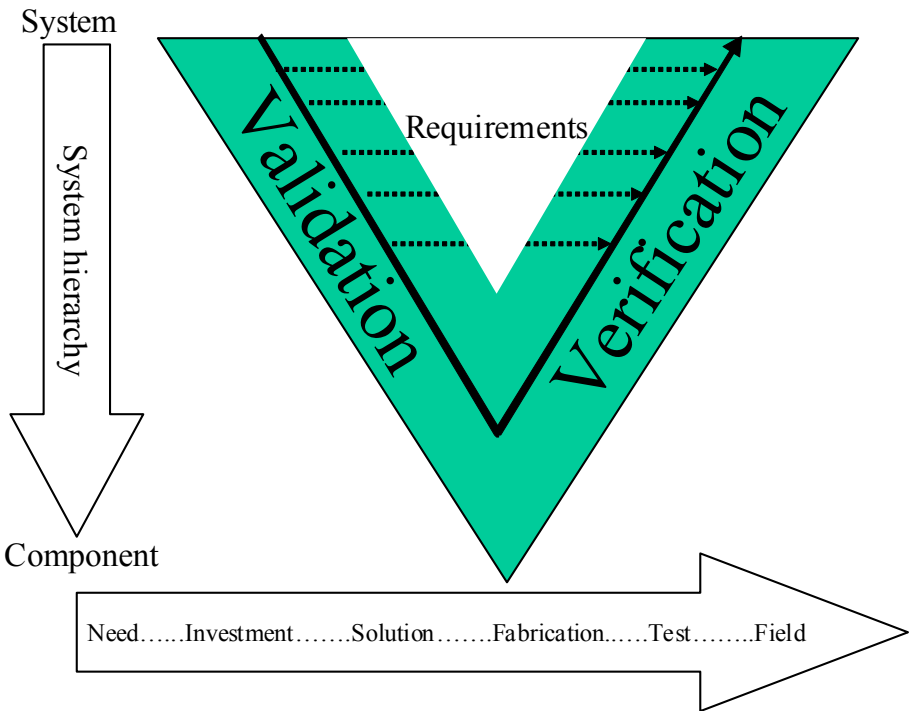



**Figure 4.12-4. System Engineering "V" Diagram**

A large part of this SE activity is challenging the requirements need and the requirements' associated values before development of solutions. This activity helps to ensure that an economy of effort exists on the project and that resources are not wasted on developing solutions for unnecessary requirements. At each stage, the Validation process provides increasing confidence of the correctness and completeness of system requirements.

#### 4.12.1.1 Definition of Validation

There are multiple definitions of the Validation process, but, for the purposes of this manual and the Federal Aviation Administration (FAA), the accepted definition of the Validation process is:

> *"the determination that the requirements for a product are sufficiently correct and complete." (SAE ARP 4761, 1996)*

#### 4.12.1.2 Objective of Validation

The primary objective of the Validation process is to ensure that requirements are correct and complete. In addition, the Validation process ensures that requirements defined for a system

are consistent with the characteristics listed in Requirements Management (Section 4.3). Successful Validation confirms that the identified requirements are justified, relevant, and logically correct in terms of the customer's needs and operating environment. In addition, the Validation process also ensures that the identified set of requirements is complete (i.e., containing all essential elements). To achieve Validation's objective, Validation activities are performed as early as possible in the development phase after requirements are identified; thus, Validation follows requirements development and precedes design solution.

The Validation process is conducted in order to find and correct poor requirements, which stem from three sources:

- Ambiguous requirements statements
- Incorrect (including unnecessary) requirements statements
- Incomplete (or omitted) requirements statements

### 4.12.1.3 Interfaces With Other System Engineering Processes

The SE elements that interface with the Validation process appear in Figure 4.12-5 and are described in "Inputs to Validation" (Paragraph 4.12.1.4).

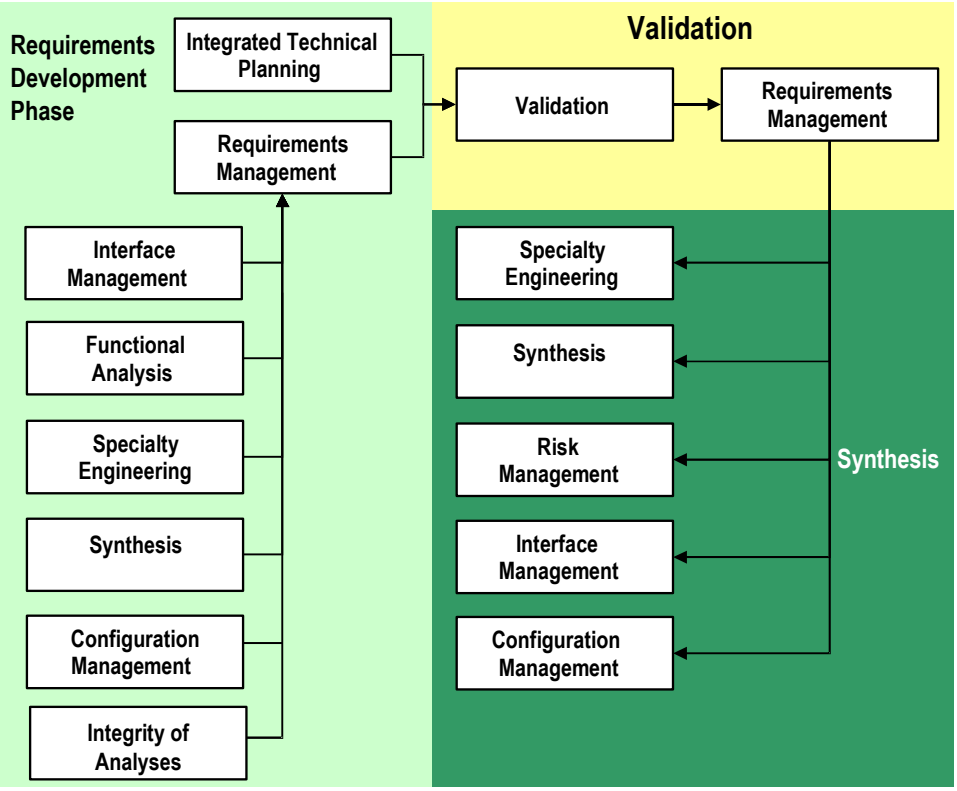



**Figure 4.12-5. How Validation Interfaces with Other System Engineering Processes**

### 4.12.1.4 Inputs to Validation

The inputs to the Validation process include:

- Stakeholder Needs
- Standards
- Technical Plans (Integrated Program Plan (IPP), National Airspace System (NAS) Architecture, and program System Engineering Management Plan (SEMP))
- Requirements
- Functional Analysis (Functional Architecture, Operational Services and Environmental Description (OSED), and Concept of Operations (CONOPS))
- Operational Concept Demonstrations
- Interface Requirements Document(s) (IRD)
- Demonstrations
- Design Analysis Reports (DAR)
- NAS SEMP
- Physical Architecture

#### 4.12.1.4.1 Stakeholder Needs

The original Stakeholder Need generated from a NAS stakeholder (or stakeholders) to identify a capability shortfall requires Validation. Once a Stakeholder Need is validated, SE continues to ultimately provide a balanced solution to the need.

#### 4.12.1.4.2 Standards

Industry and government standards are additional inputs to the Validation process. These documents often contain information required to validate the Requirements of a system not found in higher-level requirements documents. They include publications and standards from the Society of Automotive Engineers (SAE) and/or the International Organization of Standards (ISO), as well as U.S. Government advisory circulars and FAA regulations.

#### 4.12.1.4.3 Technical Plans

Technical plans are an output of the Integrated Technical Planning process (Section 4.2). These plans define the program's tailored tasks for conducting Validation and Verification for a specific program. The IPP lays out the overall program and details the program's planned activities. The FAA Acquisition System Toolset (FAST) (http://fast.faa.gov/ams/ippdesc.htm) supplies a complete description of the IPP, and Integrated Technical Planning discusses SE's role in producing the IPP. In addition to the IPP, the program's SEMP and the NAS Architecture shall be used as inputs to the Validation process. The NAS Architecture is considered a part of the technical plans package in that it defines the FAA framework for future systems in the NAS. This architecture is a useful resource for validating the Requirements for systems developed for NAS Modernization.

#### 4.12.1.4.4 Requirements

Requirements documents are outputs from the Requirements Management process (Section 4.3). These documents include the initial Requirements Document (iRD) and final

Requirements Documents (fRD) (as they become available), as well as supporting documents such as:

- Program and technical requirements
- Customer operational requirements, including the Mission Need Statement (MNS)
- Regulatory, agency, and statutory requirements

The Requirements are classified under several categories described in "Requirements Category" (Paragraph 4.3.3.2.1.4.3). The two major categories are (1) program requirements and (2) technical requirements. Program requirements are imposed on vendors through contracts, not specifications. Technical requirements apply to the system or service under acquisition, and they are described in requirements documents, system specifications, and IRDs.

#### 4.12.1.4.5 Functional Analysis

The Functional Analysis process (Section 4.4) is an SE tool that provides a functional (what the system does, not how) description of a system that becomes a framework for synthesis and requirements development. It is recommended that the output of this process be used to validate Requirements. The outputs of this process are:

- Functional Architecture(s)
- OSED; RTCA/DO-264, Appendix C, System Handbook (SSH), Sections 4.1.1 and 3.8
- CONOPS

#### 4.12.1.4.6 Operational Concept Demonstrations

Operational Concept Demonstrations ("Demonstrations" (Paragraph 4.8.0.4.8)) are conducted to determine and validate high-risk Requirements associated with an unvalidated CONOPS.

#### 4.12.1.4.7 Interface Requirements Documents

IRDs are another example of system design information. These documents, which are outputs of the Interface Management process (Section 4.7), provide a deeper understanding of the underlying interfaces, functions, and reasons for the Requirements. These descriptions include the system-level interface definitions. Part of the Validation of a system is the assurance that the Requirements for these interfaces are correct.

#### 4.12.1.4.8 Demonstrations

Specialty engineers, as deemed necessary, often conduct Demonstrations ("Demonstrations" (Paragraph 4.8.0.4.8)) as part of analysis efforts (e.g., maintainability demonstration or human factors demonstrations). These Demonstrations provide useful feedback on the effectiveness and value of various design alternatives. Additionally, the Demonstrations may generate information for use while validating Requirements are being validated.

#### 4.12.1.4.9 Design Analysis Reports

DARs are outputs of the Specialty Engineering process (Section 4.8). These reports document the results of the Specialty Engineering analyses, which may contribute to the identification, Validation, and Verification of Requirements.

#### 4.12.1.4.10 National Airspace System System Engineering Management Plan

The NAS SEMP defines the overall plan for SE in the Acquisition Management System (AMS). This plan details *who, what, when,* and *why* SE tasks are performed in support of AMS programs. The System Engineering Manual (SEM), on the other hand, defines how the SE processes are performed.

#### 4.12.1.4.11 Physical Architecture

The Physical Architecture is essentially the engineering design of the system that is produced via the Synthesis process (Section 4.5). This information may vary in detail, depending on the phase of the program. This input is essential so that the persons responsible for the Validation process understand the product Requirements and configuration (if available). Information includes:

- Drawings (if updating current systems, and if they exist in the Validation phase)
- Design descriptions
- System descriptions

### 4.12.1.5 The Validation Process

The following sections describe the purpose, general outcomes/expectations, and tasks of the Validation process.

#### 4.12.1.5.1 Validation Process Purpose

Validation is primarily performed to ensure the correctness and completeness of the requirements that define a system. ARP 4754, Paragraph 7.1, defines correctness and completeness as follows:

- Correctness of a requirements statement means the absence of ambiguity or error in its attributes
- Completeness of a requirements statement means that no attributes have been omitted and that those stated are essential

System requirements are analyzed to ensure that the defined set of Requirements is consistent with the operational need defined in the CONOPS, Specialty Engineering analyses, and MNS. The Validation process is conducted to provide objective evidence that the services provided by the system, as defined in the requirements document, comply with the Stakeholder Needs, as defined in the analyses, CONOPS, and MNS. When variances are identified, they are recorded and used to guide corrective actions. Because Validation is a comparative assessment of Requirements against needs, it also results in confirmation that Stakeholder Needs are correctly identified and requested. Stakeholders normally ratify Validation of Requirements at the system level.

"Task 5: Analyze Requirements Documents and System Analyses" (Paragraph 4.12.1.5.3.5) describes the desired attributes of Requirements. The Requirements Management (Section 4.3) also describes the desired attributes of individual Requirements.

#### 4.12.1.5.2 Validation Process Objectives

The general objectives of the Validation process include:

- Development of the Validation Table and inclusion of the Validation Table in a Validation Report
- Appending to or referencing by the existing requirements documents of the Validation Report
- Confirmation that the system services required by stakeholders are properly documented in the Requirements
- Confirmation that the stakeholder requirements faithfully describe the required system services
- Reporting of nonconformance, which is used to guide corrective actions
- Traceability of all requirements to higher-level Requirements
- Documentation of the program's concerns/issues and constraints

#### 4.12.1.5.3 Validation Process Tasks

All Requirements in all categories are required to be validated. In general, the Validation of higher-level Requirements serves as a basis of Validation for lower-level Requirements. The tasks involved in the Validation process are conducted in three phases: planning, evaluation, and documentation. The recommended process tasks for validating Requirements are shown in Figures 4.12-3 and 4.12-6 and are described in the following paragraphs.

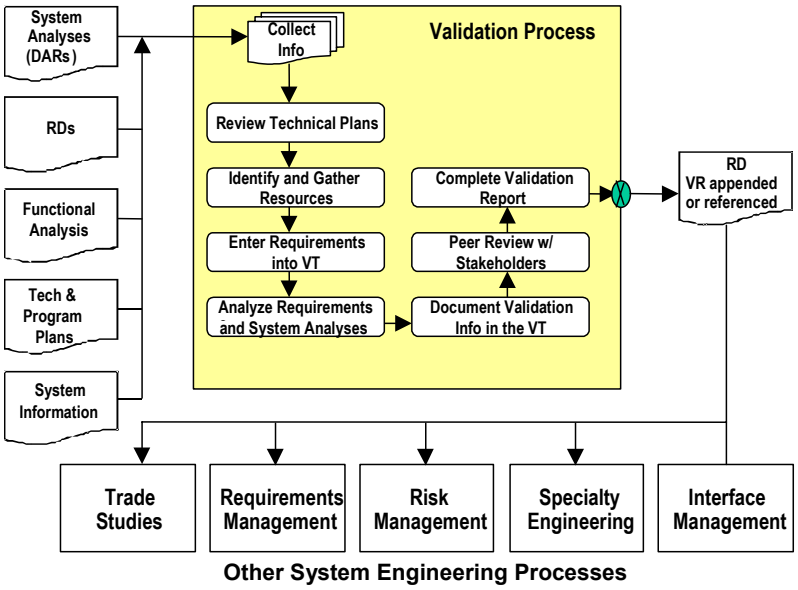



**Figure 4.12-6. Overall Validation Process and Outputs**

#### 4.12.1.5.3.1 Task 1: Collect Identified System Requirements and Constraints

The initial step in the Validation process is to accept the set of Requirements to be validated from the Requirements Management process (Section 4.3). In addition, the information required for Validation is gathered, which documents the baseline system requirements, resources, and constraints. These documents are described in "Inputs to Validation" (Paragraph 4.12.1.4) and include the requirements documents, technical plans, and system description information.

#### 4.12.1.5.3.2 Task 2: Review the Existing Technical Plans

The next step is to review the program and acquisition plans, such as the IPP and the MVP, if it exists. These plans include the Validation tasks to be performed; allocation of responsibility to organizations; schedule; and costs. The objective is to define the strategy for validating the system's services in its operational environment and achieving customer satisfaction in accordance with these plans. This strategy depends on the lifecycle stage (e.g., whether a model, prototype, or actual product is being verified); on risks (e.g., novelty, safety, technical, and commercial criticality issues); and on the agreement and organizational constraints of the stakeholder requirements. It is required that, where appropriate, Validation steps (e.g., various operational states, scenarios, and missions) be defined that progressively build confidence in compliance of the installed system and assist diagnosis of any noncompliance.

**NOTE**

*Where Stakeholder Needs are unable to be specified in advance or change frequently, repeated Validation of (often rapidly developed) increments in system evolution may be employed to refine stakeholder requirements and mitigate risks in the correct identification of need. For example, ISO 13407 describes an iterative lifecycle that involves users.*

#### 4.12.1.5.3.3 Task 3: Identify and Gather Resources

At this stage, the Validation resources are formed from the appropriate SE resources. These resources include tools, information, and organizations, including the execution teams, stakeholders, and SE.

#### 4.12.1.5.3.4 Task 4: Enter the Identified Requirements Into a Validation Table

This step involves entering or copying the Requirements from the requirements document into a table, spreadsheet, database, or other SE tool appropriate to managing the Validation of Requirements. Table 4.12-1 shows an example of a typical Validation Table. Each Requirement and specification that defines a system, at all levels, shall be listed in a Validation Table.

**Table 4.12-1. Example Validation Table**

| (PUI) | Requirement | Requirements Document or specification? | Validated? Y/N | Source(s) | Location in Source | Conformance information | Corrective Action Owner |
|---|---|---|---|---|---|---|---|
| Program Unique Identifier. Enter a unique number here to ID the Requirement. This ID is the paragraph number from the requirements document. | Copy the Requirement here verbatim from the requirements document and specification. | Identify where the Requirement is found. | Indicate whether the Requirement was validated. | …of Validation. Explain the source of the Validation, (e.g., a safety analysis or other means). | Where specifically in the source the Requirement is validated. | State conformance basis. If nonconformance is found, state recommended or required corrective action. | Organization or individual that owns the conformance or corrective action |
| 3.2.1.1.1 | The ADS-B system shall continue to operate normally in icing conditions up to heavy icing, as defined in 14 CFR FAR 25. (example only) | iRD | YES or check | IRD, ADS-B, OSA | IRD: Paragraph 3.2.1.1.1<br>OSA: Paragraph 2.5.5 | System safety confirms that icing is expected in the operational environment description. | AND-710 |

### 4.12.1.5.3.5 Task 5: Analyze Requirements Documents and System Analyses

During Task 5, a review of the existing requirements documents is performed. Also during this task, the set of Requirements that is being evaluated for validity is compared to the existing higher-level requirements documents. The Validation of higher-level Requirements may serve as the basis for Validation of lower-level Requirements, if traceability is demonstrable. If the existence of a validated Requirement in a higher-level requirements document is shown, then lower-level Requirements that are traced from the validated Requirement may be partially validated on this basis. The lower-level Requirements still need to meet the characteristics listed in “Validation Process Purpose” (Paragraph 4.12.1.5.1). For example, assume that a Requirement is listed in a validated MNS and the current task is to validate the functional requirements. If the functional requirement is traceable to a Functional Architecture based on the MNS (higher level), then the functional requirement (lower level) is considered partially validated by virtue of this traceability. However, the functional requirement in this example still requires evaluation of the characteristics listed in “Validation Process Purpose.” Once complete, the Requirement is considered validated.

If a Requirement is not contained in a higher-level requirements document, then it is evaluated by detailed review of Functional Analyses, results of prototype evaluations, Specialty Engineering analyses in documented DARs, specified design guides, CONOPS, the NAS Architecture, and other industry and government standards that describe the system and assess the system’s needs and capability shortfalls. These documents often contain information needed to validate Requirements not found in higher-level requirements documents. In these

documents, the Verification team looks for candidate requirements, recommendations, functional requirements, and other information that confirm the need for the stated Requirement.

The following Validation principles shall be employed when performing Validation activities:

- Ensure that stakeholders and testers are an integral part of the Validation process
- Perform research and analysis to find information and/or related Requirements that confirm the need for a particular Requirement (e.g., a set of related Requirements may confirm the need and validity of a derived Requirement)
- Note Requirements that are unable to be confirmed; these Requirements are noted as nonconforming[1] and evaluated for removal in the Requirements Management process (Section 4.3)
- Conduct Validation activities to detect (in the system or services) the existence of random and systematic nonconformance to stakeholder requirements
- Ensure that the Validation process is undertaken in a manner consistent with defined and documented organizational practices to minimize uncertainty in the replication of Validation actions, conditions, and outcomes
- Maintain objective and authenticated records of Validation actions and outcomes
- Conduct fault resolution of a nonconformance in the Requirements Management process to a level of resolution consistent with cost-effective remedial action, including revalidating following defect correction and/or organizational quality improvement actions
- Conduct Validation activities to determine the correctness and completeness of the Requirements

When Validation is performed, the following correctness and completeness checks (may be tailored by expansion) shall be completed at each level of the Requirements hierarchy:

**Correctness**

1. Requirements correctly stated:
    - What is required (design independent)
    - Unambiguous
    - Statements lead to appropriate design
    - Achievable with current or emerging technology
    - Requirement is verifiable
    - Stated for appropriate environmental conditions (ambient and operational)

[1] Nonconformance means that a needed Requirement is missing, or an existing Requirement is unable to be validated. In accordance with agreement terms or organizational objectives, Validation is conducted to isolate the part of the system that gives rise to a nonconformance, which may result in the need for corrective action and/or changes in quality management policy. “Objective of Validation” (Paragraph 4.12.1.2) discusses the sources of nonconformance.

- Stated for normal and abnormal operations
- Derived Requirements supported by analyses
- Each Requirement has an identified source

2. Requirements correctly reflect the analyses:
   - Appropriate analyses completed correctly
   - System hazards correctly identified and classified according to risk
   - System characteristics in DARs correctly identified and classified
   - Reliability, availability, fault detection, and tolerances identified
3. Functions correctly identified:
   - Requirements based on functions
   - Functions significant to Requirements
   - Documented
   - Traced to higher functions
   - Constrained by higher-level Requirements

**Completeness**

1. Requirements traceable to an identified source:
   - Functional Analysis
   - Higher-level requirements documents
   - Safety assessments
   - Reliability, maintainability, and availability (RMA) analyses (Failure Modes and Effects Analysis (FMEA), Failure Modes and Effects Criticality Analysis (FMECA))
   - Requirements identified in DARs (Specialty Engineering" (Paragraph 4.12.2.3.4))
   - Derived Requirements
   - Regulations, standards, or statutory requirements
   - OSED
   - Integration requirements
2. Constraints defined, substantiated, and addressed:
   - State of the art
   - Safety
   - Environment
   - Industry and FAA standards
   - Specify system implementation

3. System implementation specified:
    - Functional analysis completed
    - Requirements allocated to systems
    - Architecture defined at each functional level
    - Interfaces (internal and external) defined—human, hardware, software, physical, functional, procedural, and environmental (ambient and operational)
4. All prohibited behaviors and characteristics explicitly stated
5. All technical performance measures explicitly stated

**4.12.1.5.3.6 Task 6: Document the Validation Information in the Validation Table**

During this task, Validation data is collected, classified, and collated in the Validation Table described in "Task 4: Enter the Identified Requirements Into a Validation Table" (Paragraph 4.12.1.5.3.4) and in accordance with criteria defined in the program and acquisition plans. This process categorizes conforming and nonconforming Requirements according to their source and corrective action owner. The Validation data is then analyzed to detect essential features, such as trends and patterns of failure, evidence of systemic failings, and emerging threats to system services.

**4.12.1.5.3.7 Task 7: Peer Review the Validation Table With Stakeholders**

During this task, the stakeholders of the system's Requirements are identified. Once the Validation Table is filled, the stakeholders review it. Stakeholder comments are incorporated into the table, and the table is finalized.

**4.12.1.5.3.8 Task 8: Document the Requirements Validation Analysis in the Validation Table and Include the Validation Table in a Validation Report**

The results of the Validation analysis are documented in the Validation Table, and the Validation Table is included in a Validation Report. The Validation Report is transmitted to Requirements Management (Section 4.3). This report is appended to or referenced by the requirements document.

The Validation Report summarizes the Validation effort and results and communicates the Validation Table to other SE processes. The following format shall be used as a guide for the contents and organization of a Validation Report.

Validation Report format:

I. Summary of Validation efforts and results
    a. Summarize the Validation results when locating conforming and nonconforming Requirements

II. System and program description

III. Methodology used

IV. Unvalidated Requirements

a. List of nonconforming Requirements

b. Recommendations for correction of nonconforming Requirements

V. Validation Table

VI. Discussion of trends and patterns of failure, evidence of systemic failings, and emerging threats to system services.

#### 4.12.1.6 Tailoring of Validation Activities

Tailoring of a program's Validation activities is limited to the following:

- The specific means of Validation may include the techniques and tools employed and described in SAE ARP 4754, Section 7.7, if desired by the program
- The specific contents of the Validation Report may be tailored to include additional information as specified in "Task 8: Document the Requirements Validation Analysis in the Validation Table and Include the Validation Table in a Validation Report" (Paragraph 4.12.1.5.3.8)

### 4.12.2 Verification

The Verification process ensures that the design solution has met the system requirements and that the system is ready for use in the operational environment for which it is intended. This description means that a verified system is able to demonstrate (show evidence) that it complies with mission need; functional, performance, allocated, derived, and interface requirements; and design and allocated constraints that achieve stakeholder needs. The Verification process (Figures 4.12-4 and 4.12-7) supports system evolution at all levels of the system's lifecycle, from concept to advanced studies and preliminary analyses to design and development, culminating in the production, product acceptance, operational, and disposal phases.

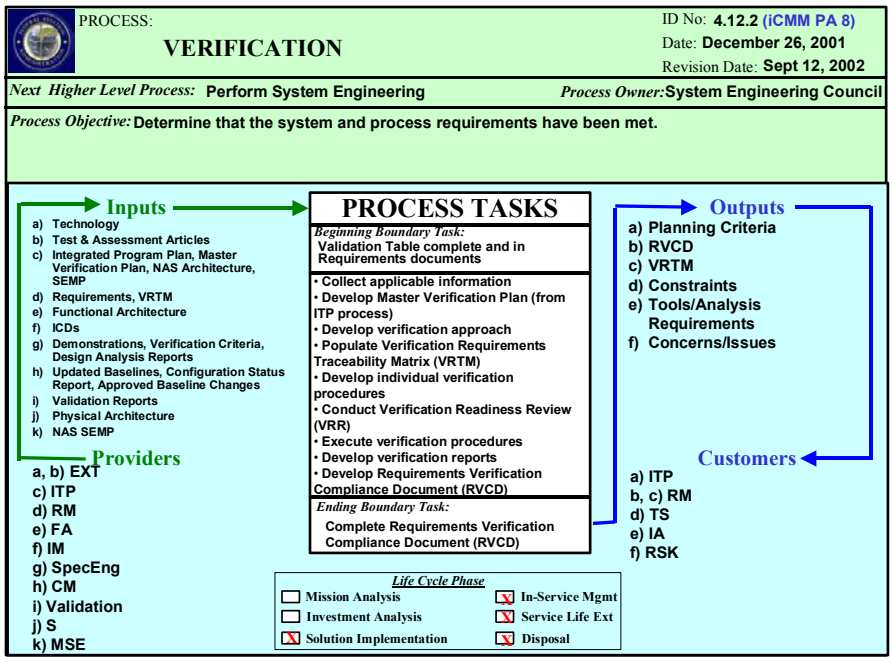



**Figure 4.12-7. Verification Process-Based Management Chart**

### 4.12.2.1 Objectives of Verification

The major objectives of the Verification process are to confirm that:

- Intended functions are correctly implemented and that the system is operationally ready and acceptable to the users
- Requirements are satisfied
- Specialty Engineering analyses, including lifecycle, remain valid for the system as implemented

Successful Verification confirms that the development process has provided a system consistent with stakeholder needs and compliant with the system's validated requirements. It is a basic principle to verify all requirements in the system's requirements documents. This principle does not imply that a test is required for every requirement, but it does imply the need to conduct some form of Test and Evaluation (T&E) and/or SE Assessment at an appropriate level to ensure that all requirements are satisfied.

The broad range of product development cycles and levels of product development complexity require that the Verification process be tailored to each project.

The expected outcomes of Verification are the development of:

- MVP (from the Integrated Technical Planning process (Section 4.2))
- VRTM
- Individual T&E and SE Assessment plans
- T&E procedures

- Verification Readiness Reviews (VRR) (if applicable)
- T&E and SE Assessment reports, which detail specific test results and assessments
- RVCD, which provides documentation that the system product conforms to system requirements and includes nonconformance reports

#### 4.12.2.2 Definition of Verification

The accepted definition of verification for this manual and the FAA is:

> *"the evaluation of an implementation [system] to determine that applicable requirements are met." (SAE ARP 4761, 1996)*

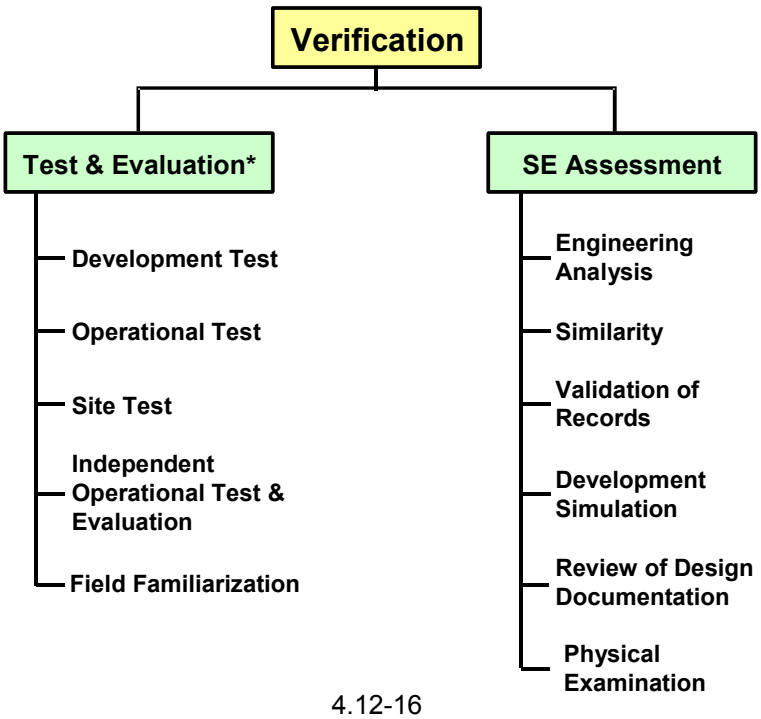
Verification is the composite of all tasks, actions, and activities performed on system elements that are required in order to evaluate the progress and measure the effectiveness of evolving system products and processes in meeting system requirements. There are two basic and complementary methods of Verification: T&E and SE Assessment, as shown in Figure 4.12-8.

##### 4.12.2.2.1 Test and Evaluation Verification

It is recommended that T&E programs be structured to:

- Provide essential information to support decisionmaking
- Provide essential information to assess technical and acquisition risk
- Verify the attainment of technical performance specifications and objectives
- Verify that a system is operationally effective and suitable for its intended use

It is also recommended that T&E objectives for each AMS lifecycle phase be designed to mitigate potential operational risks and to demonstrate system performance appropriate to that phase. Quantitative criteria provide substantive evidence for analysis of hardware, software, and system maturity and readiness to proceed through the acquisition management process.






* For more information, refer to the Test and Evaluation Process Guidelines Document on the FAA AMS Toolset (http://fast.faa.gov).

**Figure 4.12-8. Components of Verification**

It is recommended that each T&E phase have specific milestones (entrance and exit criteria) that are satisfied before the next T&E phase is entered. Parallel testing is encouraged when it is more efficient and at least as effective as serial testing.

It is highly desirable that system performance be established by test under actual (or simulated) operating conditions; however, these conditions may not be possible until the system is deployed. Problems uncovered at deployment are costly to correct; therefore, a combination of inspection, analysis, and test often is employed during program development to detect problems early, thereby reducing risk and helping to ensure a successful, cost-effective program.

Compliance with each requirement in a specification shall be verified by one or more of the methods described in this manual and as indicated in the VRTM.

T&E methods include:

- **Verification by Demonstration**. This method includes Verification accomplished by operation, adjustment, or reconfiguration of items performing their design functions under specific scenarios. The items may be instrumented and quantitative limits of performance monitored; however, only check sheets are required rather than recordings of actual performance data. This method is used when actual demonstration techniques may be used to verify compliance with a requirement. Observations made by engineers or instrumentation are compared with predetermined responses based on the requirements. An example of this Verification method is the demonstration of installing and uninstalling an aircraft engine in a specified amount of time . Demonstration is often used to verify compliance with requirements in servicing, reliability, maintainability, transportability, and human factors engineering.
- **Verification by Test**. This method is accomplished through systematic exercising of the application item under appropriate conditions, with or without instrumentation, and the collection, analysis, and evaluation of quantitative data.
- **Verification by T&E Analysis**. This method is accomplished by technical or mathematical evaluation, mathematical models or simulation, algorithms, charts, circuit diagrams, and representative data.
- **Verification by Inspection**. This method is accomplished by visually examining the item, reviewing descriptive documentation, and comparing the appropriate characteristics with predetermined standards to determine conformance to requirements without the use of laboratory equipment or procedures. Inspection is generally nondestructive and uses the senses of sight, hearing, smell, touch, and taste; simple physical manipulation; mechanical and electrical gauging and measurement; and other means of investigation. Inspection often verifies the physical design features of a system as well as construction features, workmanship, dimensions, quality, and physical conditions, such as cleanliness, installation, and finishing. Inspection may include reviews of documentation, system descriptions, and other materials to compare the actual system with predetermined standards.

The Test and Evaluation section of the FAST (http://fast.faa.gov/toolsets/index.htm) provides specific guidelines to conduct T&E.

#### 4.12.2.2.2 Verification by System Engineering Assessment

It is recommended that Verification by SE Assessment be conducted to support the development of products, services, and processes necessary to verify that system end-items satisfy their requirements. Verification assessment addresses Verification requirements and criteria for solution alternatives; definition of Verifications to demonstrate proof of concept; and development, qualification, acceptance, pertinent operational, and other testing. The assessment may also consider the requirements and procedures needed to verify critical Verification methods and processes (e.g., Verification of key methods and assumptions and the data used in Verification by analysis).

It is suggested that Verification assessment be initiated when a design concept is established. The Verification assessment is drawn from the MVP and the results of the Validation effort. According to the Integrated Technical Planning process (Section 4.2), the objective of the MVP is to define all Verification activities that demonstrate the system's capability to meet the requirements of its specification. These activities shall be fully integrated to ensure that adequate data is provided at minimum cost within the allotted timeframe. A continuing feedback of Verification data throughout product development, test, and evaluation is necessary to reduce risk and to detect problems early. The goal is to completely verify the system's capability to meet all requirements before production and operational use.

SE Assessment methods include:

- **Verification by Engineering Analysis**. This process includes the techniques of SE analysis, Specialty Engineering, statistic and qualitative analysis, simulations, and modeling. Engineering analysis is used when testing is not feasible, similarity is nonapplicable, and inspection is inadequate.
- **Verification by Similarity**. This process assesses compliance with requirements by reviewing a similar system's test data, configuration, and applications. This method is only used when the systems are similar in design and manufacturing, and the prior system was qualified to equivalent or greater specifications. Great care is taken to ensure that the intended application environment of the emerging system is identical or less rigorous than the environment of the previous system testing.
- **Validation of Records**. This process reviews manufacturing records at end-item acceptance to verify features and requirements of the system.
- **Simulation**. This process verifies design features, system behavior, and performance using simulated models of the system.
- **Review of Design Documentation**. This process uses the disciplined review of design documentation, such as reports and drawings from Acquisition Reviews, Design Reviews (preliminary and critical), and other evaluations.
- **Physical Examination**. This process assesses compliance with requirements by visually inspecting a physical item or configuration according to preestablished criteria.

### 4.12.2.3 Interfaces With Other System Engineering Processes

Verification has multiple interfaces with other SE elements. These interfaces are shown in Figure 4.12-9 and described in the following paragraphs.

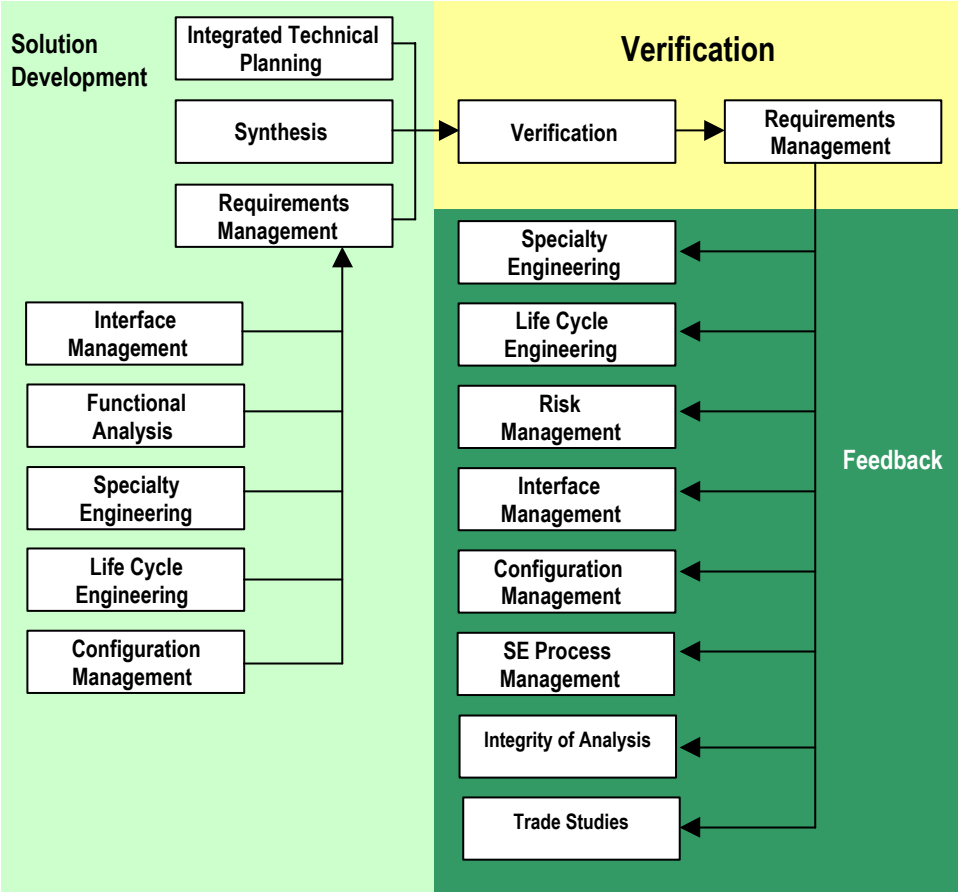



**Figure 4.12-9. Verification Interfaces With Other System Engineering Elements**

#### 4.12.2.3.1 Requirements Management

Requirements documents are outputs from the Requirements Management process (Section 4.3). These documents include the iRD and fRD, as well as underlying documents, such as customer operational requirements, system and technical requirements, and regulatory, agency, and statutory requirements. These documents also include the MNS and any Verification specification documents. The execution teams manage these documents.

#### 4.12.2.3.2 Synthesis

System, subsystem, component, and procedural designs comprise the outputs of the Synthesis process (Section 4.5). The information contained in these designs and, in some cases, test articles and/or prototypes is required for Verification.

#### 4.12.2.3.3 Integrated Technical Planning

Technical plans are an output of the Integrated Technical Planning process (Section 4.2). They define the program's tailored tasks for a specific program. The IPP lays out the overall program; The MVP comes from the Integrated Technical Planning process but is a separate plan.

#### 4.12.2.3.4 Specialty Engineering

Specialty Engineering (Section 4.8) both feeds and is fed by the Verification process. Specialty Engineering often is a source of requirements and design constraints that require Validation and Verification. In addition, Specialty Engineering analyses often are used to assist in the Verification of requirements as part of assessment. Specialty Engineering DARs are the major outputs of the Specialty Engineering process. These reports document the results of the Specialty Engineering analyses, which may result in the identification and Validation and Verification of requirements. Once Verification is complete, the verified requirements are checked to ensure that the Specialty Engineering DARs reflect the Verification.

#### 4.12.2.3.5 Risk Management

Risk Management (Section 4.10) is another SE element that both feeds and is fed by the Verification process. Risk Management is able to drive the Verification of high-risk requirements. In addition, all requirements that fail to meet verification criteria are considered a risk to the program. These requirements become inputs to the Risk Management process for mitigation.

#### 4.12.2.3.6 Interface Management

Results of the Interface Management process (Section 4.7) provide a deeper understanding of the underlying physical and functional interfaces of the system requirements. The interface documentation includes the system-level interface definitions.

##### 4.12.2.3.6.1 Lifecycle Engineering

Lifecycle Engineering (Section 4.13) is another SE element that both feeds and is fed by the Verification process. This element provides supportability, deployment and transition, real estate and disposal requirements, and design constraints. These requirements and design constraints undergo the Verification process to ensure compliance.

### 4.12.2.4 Inputs to Verification

There are four major input categories to Verification:

- Technology
- Technical Plans
  - IPP
  - MVP
  - Program SEMP
  - NAS Architecture
- Requirements
  - Requirements documents and associated Validation Reports
  - VRTM templates populated with Requirements
- Design information and Test and Assessment articles

- Functional Architecture
- Physical Architecture
- Interface Control Documents (ICD)
- Demonstrations
- Verification Criteria
- DARs
- Updated Baselines
- Configuration Status Report
- Approved Baseline Changes

#### 4.12.2.4.1 Technology

State-of-the-art Technology constrains the means of Verification. Therefore, it is critical that this factor be considered in the development of the Verification approach.

#### 4.12.2.4.2 Technical Plans

These plans, developed via the Integrated Technical Planning process (Section 4.2), detail the overall vision for executing the program, including the timing and sequence of Verification. The plans that need to be collected to properly conduct Verification include the IPP, the MVP, and program SEMP. The NAS Architecture is also a valuable input in that it defines the FAA framework in which the system being verified eventually operates.

#### 4.12.2.4.3 Requirements

Requirements documents are an output of the Requirements Management process (Section 4.3). These documents include customer operational requirements, as well as regulatory agency and statutory requirements. With Validation Reports (and associated Validation Tables) and Verification specifications included, these documents are the primary source of information for the Verification process. Phase-specific implementation teams maintain requirements documents. It is recommended that these documents include the most up-to-date information from interfaces, Functional Analyses, Specialty Engineering analyses, and system configuration.

#### 4.12.2.4.4 Design Information and Test and Assessment Articles

This input is essential to understanding the product configuration. (Configuration Management (Section 4.11) supplies a complete description of this process.) To develop the MVP and the individual test plans, the system engineer needs any available design information, including Physical Architectures, drawings, interface documents, system design specifications, functional specifications, product specifications, and test equipment designs. This information also includes Specialty Engineering DARs used for the assessment. In addition, Functional Architectures and their associated analyses need to be available. The results of the Functional Analyses provide a deeper understanding of the underlying functions and reasons for the Requirements. ICDs, if they exist at the time of Verification, are also required. These documents provide detailed information on the interfaces involved in system operation. Part of the Validation and Verification of a system is the assurance that the Requirements for these

interfaces are correct and satisfied. The Test and Assessment Articles are the constituent pieces of the system, or the system in its entirety, on which Verification is performed.

#### 4.12.2.5 The Verification Process

Verification is accomplished through a combination of T&E and SE Assessment. The general Verification process tasks are grouped into three distinct phases: planning, Verification activities, and documentation. Planning and documentation are common to both T&E and SE Assessment. Planning includes determination of the resources required, sequence and timing of activities, data and documentation to be produced, and establishment of the assessment criteria. The results of the planning effort are documented in the MVP. The documentation phase includes those tasks taken to ensure that evidence of completion is recorded and collated. The activity phase includes the processes or tasks in which the actual Verification methods are employed, whether they are T&E or SE Assessment. These processes are described below.

##### 4.12.2.5.1 Process for Verification by Test and Evaluation

Specific guidelines for planning and conducting a T&E process are included in the FAA AMS Test and Evaluation Process Guidelines located under Test and Evaluation in the index of the FAST (http://fast.faa.gov/toolsets/index.htm).

##### 4.12.2.5.2 Process for Verification by System Engineering Assessment

Verification by the SE Assessment is accomplished simultaneously and is fully coordinated with other SE processes—Integrated Technical Planning (Section 4.2); Requirements Management (Section 4.3); Interface Management (Section 4.7); Specialty Engineering (Section 4.8); and Risk Management (Section 4.10)—and test functions to ensure project costs, schedules, and risk implications are managed efficiently. The program plan for the Validation and Verification process is documented in specific detail in the MVP and in general in the IPP. Figure 4.12-10 depicts the overall Verification process.

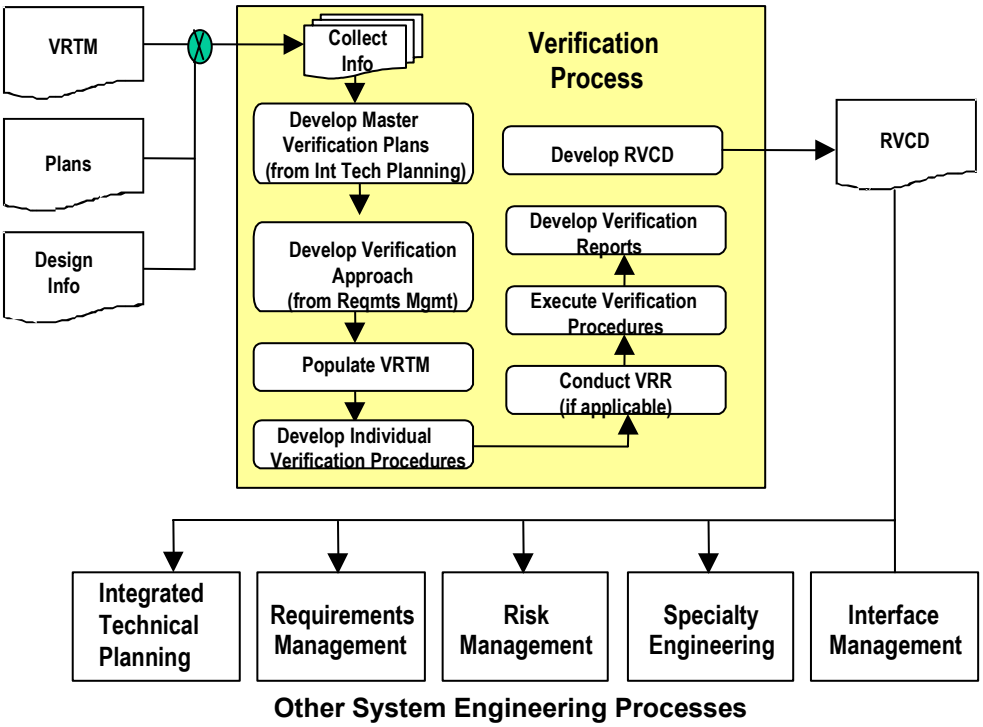



**Figure 4.12-10. General Verification Process and Outputs**

#### 4.12.2.5.2.1 Verification Process Purpose

Through assessment of the system product, the Verification process demonstrates that system behavior and characteristics comply with the specified Requirements. Verification provides the information required to effect the remedial actions necessary to correct nonconformance in the realized system or the processes that act on it.

#### 4.12.2.5.2.2 Verification Process Tasks

The recommended process tasks for conducting Verification of Requirements by SE Assessment are shown in Figure 4.12-7 and are described in the following paragraphs.

##### 4.12.2.5.2.2.1 Task 1: Collect Applicable Information

At minimum, the inputs discussed in Paragraph 4.12.2.4 shall be collected and reviewed for impacts on the Verification process. For instance, the DARs generated by Specialty Engineering (Section 4.8) may have identified special Verification procedures or needs.

##### 4.12.2.5.2.2.2 Task 2: Obtain Master Verification Plan From Integrated Technical Planning or Develop It Now

As the Verification approach is refined, the facilities, budget, schedules, personnel, test articles, instrumentation, and data necessary to accomplish the Verification events are also identified, coordinated, and approved with the appropriate decision authorities, resulting in an approved Verification plan for the program. This strategy and overall plan for the Verification process is documented in the MVP, which is delivered from the Integrated Technical Planning process (Section 4.2) to “Task 7: Execute Verification Procedures” (Paragraph 4.12.2.5.2.2.7). The MVP

is required to provide the content and depth of detail necessary for understanding the Verification activities. Each major activity is defined and described in detail. The MVP covers all qualification, acceptance, predevelopment, operational, and disposal Verification activities for hardware, software, and procedures. The MVP provides a general schedule and sequence of events for major Verification activities. It also describes test hardware and software, support equipment, and facilities required to support Verification activities. The MVP is developed by design, system, and test engineers with a thorough understanding of the requirements document, segment requirements and specifications, and Validation Table.

It is recommended that the following activities be completed during the planning stage:

- Identify the system and system configuration, including definition of test equipment and telemetry, facilities, and support equipment
- Identify and collate all Requirements appropriate to the (level of) Verification
- Define the specific Verification method employed for each Requirement
- Define the criteria used to evaluate the evidence from each Verification for each Requirement

#### 4.12.2.5.2.2.3 Task 3: Develop Verification Approach

Simply put, the Verification approach is how the Requirements are going to be verified. This approach is developed in Requirements Management (Section 4.3) and documented in the VRTM. This task includes the activities of receiving, updating, analyzing, decomposing, and summarizing Requirements to ensure that they are economically and efficiently measurable and are able to be appropriately distributed for Verification planning. The purpose of the Develop Verification Approach activity is to determine and document the Verification approach to ensure that the product is compliant with the identified Requirements.

In this step, the Verification specification (from Requirements Management) is used to develop a Verification approach for each Requirement documented in the Validation Table. The Validation Table is further refined into a VRTM. The VRTM is the heart of the Verification process. The strategy or method used to verify each Requirement is specified in a Verification Requirement, and the Verification Requirements are listed in the VRTM. The VRTM defines how each Requirement (functional, performance, design, etc.) is to be verified, the stage in which the Verification is to occur, and the applicable Verification levels. The VRTM essentially establishes the basis for the Verification program. SE and the Verification team develop the VRTM together. The T&E and the SE Assessment methods available for use are discussed in detail in Paragraphs 4.12.2.2.1 and 4.12.2.2.2. Table 4.12-2 is an example VRTM. Specific guidelines for the VRTM are included in the Test and Evaluation section of the FAST (http://fast.faa.gov/toolsets/index.htm).

**Table 4.12-2. Sample Verification Requirements Traceability Matrix**

| Section 3 Requirements Paragraph Reference for Specification SCN (Paragraph No./Title) | Requirement Description | Verification Method (Test, Demonstration, Analysis, Inspection, Engineering Analysis, Similarity, Validation of Records, Simulation, Documentation) | Verification Plan (Indicate which plan describes the Verification of the requirement) | Remarks |
|---|---|---|---|---|
| 3.1.1.1 Aircraft I.D. | | T = Test | | |
| 3.1.1.2 | | D = Demonstration | | |
| 3.1.1.3 | | A = Analysis | | |
| 3.2.1.1 System Alignment | | I = Inspection | | |
| 3.3.1.1 Transmit Time | | EA = Engineering Analysis | | |
| 3.3.1.2 Receive Time | | SY = Similarity | | |
| 3.3.1.3 Process Time | | VR = Validation of Records | | |
| 3.3.1.4 Display Time | | SM = Simulation | | |
| 3.3.1.5 System Check | | DC = Documentation | | |

#### 4.12.2.5.2.2.4 Task 4: Populate the Verification Requirements Traceability Matrix

Verification is performed at all levels in a system. Each Requirement is verified either by test, SE Assessment, or both, as appropriate. As mentioned earlier, the strategy or method used to verify each Requirement is specified in a Verification Requirement, and the Verification Requirements are documented in the VRTM. It is recommended that a description of the test or SE Assessment and the criteria used to determine conformance and disposition of each Verification Requirement be included in the VRTM.

#### 4.12.2.5.2.2.5 Task 5: Develop Individual Verification Procedures

This process is the detailed development of Verification procedures and resources that achieve specified Verification objectives using approved agency and regulatory procedures. Specific guidelines on content and format are included in Sections 6 and 7.1 of the FAST and Test and Evaluation Guidelines (http://fast.faa.gov/toolsets/index.htm).

The product Verification procedures consist of step-by-step directions to conduct the actual product Verification at any level. Traceability to all Requirements in the VRTM shall be shown as an integral part of these procedures. The procedure is tailored to the Verification activity that is to be performed to satisfy Requirements and may be a test, SE Assessment, or a combination of both. The as-run and certified copy of the procedure is maintained as part of the project's archives as test or SE Assessment plans.

All Verification procedures for both hardware and software include development of test plans, procedures, and test cases. The process includes performing timing and sizing analysis Verification at the subsystem and system levels. The results of these analyses are maintained in the test or SE Assessment plans.

The process also performs abnormal and erroneous condition testing at the subsystem and system levels. The process includes the use of regression test procedures for hardware and software integration, subsystem test, and integration and system test, including the use of a core test process, if planned.

The Verification process incorporates any commercial-off-the-shelf (COTS) software or hardware in the system integration and test planning.

#### 4.12.2.5.2.2.6 Task 6: Conduct Verification Readiness Review

A VRR or equivalent is held before each major Verification or groups of smaller Verifications with common elements. The VRR is conducted to ensure that all SE considerations are satisfied and that the readiness of all support, test, and operational systems is in order to perform the Verification process. The VRR includes a detailed review of the status of the facilities, ground support equipment, Verification design, software, procedures, and Verification Requirements. In addition, Verification activities and schedules are outlined, and organizational/personal responsibilities are identified. Emphasis is on ensuring that all Verification Requirements identified for each Verification method or technique are included in the Verification design and procedures.

A key feature of the Verification approach is the non-advocate aspect (i.e., it is a principle of the Verification process that the person or group performing the design not execute the Verification activities). The same principle applies to planning and conducting the Verification design itself. The VRR is conducted to ensure that Verification activities are planned adequately and that risks are controlled. It is recommended that the VRR be chaired by senior personnel not associated with the program but who possess some expertise in the systems and operations under evaluation. The program implementation teams manage the VRR.

#### 4.12.2.5.2.2.7 Task 7: Execute Verification Procedures

This task is the actual product of the Verification process (i.e., the conduct of tests or SE Assessment). The process of product Verification confirms through documented evidence of Verification activities that production-representative hardware and software are in compliance with functional, performance, and design requirements.

The Verification team is responsible for performing product Verification, which consists of preparation for product Verification, execution of product Verification activities, and product post-verification and documentation. Specific guidelines for the test process are found in the Test and Evaluation Guidelines in the FAST (http://fast.faa.gov/). When performing test Verification, the Verification team shall consult this document for specific instructions. Specialty Engineering (Section 4.8) supplies specific guidelines on conducting system (specialty) engineering assessments.

Responsibilities of the Verification team during the preparation phase of a Verification program using testing and demonstration may include:

- Design, fabrication, and/or preparation of the Verification setup

- Verification facility
- Verification fixture and/or stations
- Data acquisition, reduction, and archive system
- Verification control system
- Instrumentation system
- Design and fabrication of Verification article hardware/software
- Conduct of make-or-buy analyses for Verification setup hardware and software
- Coordination of Verification article delivery
- Coordination of Verification setup hardware/software delivery
- Coordination of support equipment and special Verification
- Preparation of Verification safety, hazard, and environmental compliance plans
- Assembly and installation of the Verification article, fixture, and setup
- Implementation of serial numbered component installation/removal records
- Installation of Verification instrumentation
- Preparation of instrumentation installation drawings
- Implementation of instrumentation installation/removal records
- Management of Verification configuration control
- Verification articles
- Instrumentation and measurements
- Data acquisition and reduction system
- Verification support software
- Checkout and maintenance of the Verification setup hardware and software
- Coordination of Verification article configuration buyoff and/or conformity approval inspections
- Conduct of preverification conference or VRR (or equivalent)
- Management and status reporting of Verification preparation activities

During the preparation phase, quality-control members of the Verification team establish/verify conformity of Verification articles, establish/verify conformity of the Verification methods, and check/verify systems and operations.

Responsibilities of the Verification team during the product Verification execution may include:

- Maintenance of detailed Verification notes/logs, including all deviation from the MVP
- Management of Verification configuration control
- Verification facility
- Verification fixture and/or stations

- Verification article
- Instrumentation and measurements (if required)
- Data acquisition and reduction system (if required)
- Verification support software
- Coordination of Verification article configuration and/or conformity approval inspections (if required)
- Coordination of Verification witnessing
- Checkout and maintenance of the Verification setup hardware and software
- Management of calibrated equipment (if required)
- Execution of Verification in accordance with approved MVP
- Validation, collection, reduction, archive, and delivery of Verification data
- Management and status reporting of Verification activities
- Conduct of post-verification inspections
- Identification of readiness criteria for formal and informal system and subsystem test
- Conduct of unit tests on software code changes before they are incorporated; review of software code changes for correctness and the avoidance of undesired impact on other software and system variables and components

#### 4.12.2.5.2.2.8 Task 8: Develop Verification Reports

When product Verification is complete, the Verification team is responsible for conducting a post-verification review and preparing a report to disseminate the results. The purpose of the Verification report is to determine compliance with the Verification Requirements.

Documentation of product Verification is completed by the Verification team and distributed to all interested parties. This documentation includes reports that detail the Verification results, including nonconformances, failure analyses, and other findings.

It is recommended that a Verification report be provided for each test and SE Assessment and, at minimum, for each major Verification activity. If testing occurs over long periods of time or is separated by other activities, Verification reports may be required for each individual Verification activity. It is recommended that Verification reports be completed within a few weeks following a test and include evidence of compliance with the Verification Requirements for which it was conducted. The Verification report documents the steps that were taken to ensure that the Verification process was followed and that the Verification decisions were sound.

Guidelines for developing and formatting specific types of T&E reports are specified in the Test and Evaluation section (specifically, Section 6) of the FAST (http://fast.faa.gov). For Verification by SE Assessment, it is recommended that the Verification report be documented as a DAR, as defined in Specialty Engineering (Section 4.8).

#### 4.12.2.5.2.2.9 Task 9: Develop Requirements Verification Compliance Document

The RVCD provides the evidence of compliance for each Requirement at all levels and to each VRTM Requirement. The flow down from the requirements documents to the VRTM completes the full Requirements traceability. Compliance with all the Requirements ensures that the system-level Requirement have been met.

The RVCD defines, for each Requirement, the methods of Verification and corresponding compliance information. The results of the Verification activity, including evidence of completion, are recorded and documented in the RVCD. The RVCD contains information regarding the results of each Verification activity and a description and disposition of conformance, nonconformance, conclusions, and recommendations. The compliance information provides either the actual data or a reference to the location of the actual data that shows compliance with the Requirement. The document also includes a section that details any noncompliance; this section specifies appropriate reverification procedures. The RVCD is an input to the Requirements Management process (Section 4.3); decisions regarding what to do with noncompliant Requirements are made during this process.

The specific compliance information may reference a test or SE Assessment report, automated test programs, or any other data generated in the Verification process. These inputs usually occur over a lengthy period of time and may be continuous on large programs.

Up-to-date information shall be maintained in the compliance document (RVCD) for the VRR for elements already verified. The RVCD is not baselined because it is updated throughout the program's lifecycle.

The purpose of this process is to analyze the data and results from "Task 7: Execute Verification Procedures" (Paragraph 4.12.2.5.2.2.7). If the Requirements have not been satisfied, coordination shall occur (with customer/stakeholder involvement, as necessary) to determine the impacts on the Requirements, design, and Verification approach. As a result of the impact analysis, compliance reports are generated, and the appropriate action(s) regarding the noncompliance are taken. This activity is iterative and shall be performed each time "Task 7: Execute Verification Procedures" is initiated. It is recommended that compliance reports include Requirements' identification information, compliance status, and Verification approach information.

The Validation and Verification process is completed when the information in the RVCD documents that all identified Requirements have been addressed by Verification activities and the product is compliant. When product Verification is completed, SE is responsible for completing/updating the RVCD.

#### 4.12.2.6 Disposal of Resources

This process obtains formal direction or consent for shipment, contract transfer, sale, scrap, donation, or abandonment of Verification activity resources. Disposition ensures the safe deactivation and disposal of all system products and processes and that Verification necessary to establish compliance with disposal requirements are finished.

Once product Verification is completed, accepted, and documented by SE and the Verification team, the Verification team is responsible for identifying unused, excess, or obsolete Verification resources. Depending upon resource ownership, required disposal documentation is submitted, and resource disposal is accomplished. All resource disposal actions are documented and filed or archived, as required.

### 4.12.3 Outputs of Validation and Verification

The major outcomes of the Validation and Verification process are:

- Planning criteria for the Integrated Planning (Section 4.2) process to develop and complete the MVP (as well as the IPP and program's SEMP)
- Constraints that may affect Trade Studies activities (Section 4.6)
- Concerns/issues (Appendix D) for the Risk Management process (Section 4.10) to analyze
- Outputs unique to the Validation process
  - Validated Need
  - Validation Table documented in the Validation Report
- Outputs unique to the Verification process
  - VRTM populated with Verification results
  - RVCD
  - Tools/Analysis Requirements for conducting planned Verification approach(es)
  - T&E and SE Assessment plans (internal to Validation and Verification)
  - VRRs (internal to Validation and Verification)
  - Verification documentation, including Verification reports (internal to Validation and Verification)

### 4.12.4 Validation and Verification Tools

There are several dedicated tools available to assist in managing the relationship between requirements, their validity, and their verification method. The selection of tool(s) shall ensure that the data is transportable and able to be integrated with other related SE results. A list of tools that may be used to facilitate this process is available on the International Council on System Engineering Web site ( www.incose.org). Smaller projects may successfully manage these relationships with a simple spreadsheet or database application instead of a dedicated tool. (The Validation Table (Table 4.12-1) and the VRTM (Table 4.12-2) further illustrate this topic.)

### 4.12.5 Unique Tailoring Guidance

The Verification team of a specific project may select the particular means of Verification for that project. For small projects, the project team may perform the function of the Verification team. The project team may perform both the SE and the Verification team functions. Regardless of the scope of the project and depending on the required or desired visibility into the Validation and Verification process, the project team may consider merging the Validation table, VRTM, and compliance data into one consolidated table. Such a consolidated view may be readily produced with any of the following: a simple spreadsheet application (e.g., Microsoft Excel), a robust requirements traceability application (e.g., DOORS), or a relational database application (e.g., Oracle or Microsoft Access). These tools or similar tools may be used to produce this macro-level view with the capability to filter to some lesser view as needed. Table 4.12-3 illustrates this overarching consolidation view.

**Table 4.12-3. Sample Validation and Verification Traceability and Compliance Table**

| Validation | | | | | | | | Verification Traceability | | | | | | Verification Compliance | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Source Doc (*) | PUI | Reqmt | Valid (Y/N) | Valid Source(s) | Location in Source | Corr Action | Actionee | Method | | | | Level | Verif Reqmts Traceability | Verif Task | Plan Ref | Report Ref | Verif Status |
| | | | | | | | | Test | Anal | Demo | Exam | | | | | | |
| | | | | | | | | | | | | | | | | | |

### 4.12.6 References


1. Blanchard, B. "SE Management." 2nd ed. New York: John Wiley & Sons, Inc., NY.
2. Blanchard, B., and W. Fabrycky. "Systems Engineering and Analysis." 2nd ed. Englewood Cliffs, New Jersey: Prentice Hall.
3. Defense Systems Management College. "Systems Engineering Fundamentals." Fort Belvoir, Virginia: Defense Systems Management College Press, 1999.
4. Grady, J.O. "A Grand Systems Development Course, Version 6.0." San Diego, California: JOG SE, Inc.
5. International Council on Systems Engineering. "Systems Engineering Handbook." Version 2.0. 2000.
6. National Aeronautics and Space Administration. "NASA SE Handbook." June 1995.
7. RTCA, Inc. "Guidelines for the Approval of the Provisions and Use of Air Traffic Services Supported by Data Communications." RTCA DO-264.
8. Society of Automotive Engineers. "Certification Considerations for Highly-Integrated or Complex Aircraft Systems." Society of Automotive Engineering, Aerospace Recommended Practice ARP-4754, 1996.
9. Society of Automotive Engineers. "Guidelines and Methods for Conducting the Safety Assessment Process on Civil Airborne Systems and Equipment." Society of Automotive Engineering, Aerospace Recommended Practice ARP-4761, 1996.
10. Test and Evaluation Process Guidelines. FAA Acquisition Management System. July 2001 <http://fast.faa.gov/toolsets/index.htm>.

# Section 4.13 Life Cycle Engineering

## Reserved

# Section 4.14 Maintain System Engineering

# Reserved

# APPENDIX A: ACRONYMS

## A

| | |
|---|---|
| AC | Alternating Current |
| ADS-B | Automatic Dependent Surveillance – Broadcast |
| AMS | Acquisition Management System |
| AND | Associate Administrator for NAS Development |
| ANSI | American National Standards Institute |
| APB | Acquisition Program Baseline |
| AR | Acquisition Review |
| ARP | Aerospace Recommended Procedure |
| ASD | Office of Architecture and System Engineering |
| ASR | Airport Surveillance Radar |
| ASY | Office of System Safety |
| ATM | Air Traffic Management |
| ATM | Automated Teller Machine |

## C

| | |
|---|---|
| CCB | Configuration Control Board |
| CDR | Critical Design Review |
| CDRL | Contract Data Requirements List |
| CDRL | Critical Design Review List |
| CFD | Control Flow Diagram |
| CFE | Contractor-Furnished Equipment |
| CFR | Code of Federal Regulations |
| CI | Configuration Item |
| CM | Configuration Management |
| CNS | Communications, Navigation, and Surveillance |
| COCOMO | COnstructive COst MOdeling |
| CONOPS | Concept of Operations |
| COTS | Commercial-Off-the-Shelf |
| CSA | Comparative Safety Assessment |
| CSC | Computer Software Component |
| CSCI | Computer Software Configuration Item |
| CSU | Computer Software Unit |

## D

| | |
|---|---|
| DAR | Design Analysis Report |
| DC | Direct Current |
| D/CFD | Data/Control Flow Diagram |
| DD | Data Dictionary |
| DER | Designated Engineering Representative |
| DFD | Data Flow Diagram |
| DID | Data Item Description |
| DoD | Department of Defense |
| DOORS | Dynamic Object Oriented Requirements System |
| DOT | Department of Transportation |

**E**

| | |
|---|---|
| $E^3$ | Electromagnetic Environmental Effects |
| EIA | Electronic Industries Association |
| EMC | Electromagnetic Compatibility |
| EMI | Electromagnetic Interference |
| EPA | Environmental Protection Agency |

**F**

| | |
|---|---|
| FAA | Federal Aviation Administration |
| FAR | Federal Acquisition Regulations |
| FAR | Federal Aviation Regulations |
| FAR | Functional Architecture Referencing |
| FAST | Federal Aviation Administration Acquisition System Toolset |
| FBR | Functional Baseline Review |
| FCA | Functional Configuration Audit |
| FFD | Functional Flow Diagram |
| FHA | Functional Hazard Analysis |
| FMEA | Failure Modes and Effects Analysis |
| FMECA | Failure Modes and Effects Criticality Analysis |
| FRACAS | Failure Reporting Analysis and Corrective Action System |
| fRD | final Requirements Document |

**G**

| | |
|---|---|
| G | Green |
| GFE | Government-Furnished Equipment |
| GPS | Global Positioning System |
| GSE | Ground Support Equipment |

**H**

| | |
|---|---|
| H | High |
| HHA | Health Hazard Assessment |
| HTS | Hazard Tracking System |
| HW | Hardware |
| HWCI | Hardware Configuration Item |

**I**

| | |
|---|---|
| IA | Investment Analysis |
| IAR | Investment Analysis Review |
| IAW | In Accordance With |
| IC | Interface Control |
| ICAM | Integrated Computer-Aided Manufacturing |
| ICAO | International Civil Aviation Organization |
| ICD | Interface Control Document |
| iCMM | Integrated Capability Maturity Model |
| ICN | Interface Control Notice |

| | |
|---|---|
| ICP | Interface Control Plan |
| ICR | Interface Change Request |
| ICWG | Interface Control Working Group |
| IDEF | Integrated Definition for Function Modeling |
| IEEE | Institute of Electrical and Electronics Engineers, Inc. |
| I/F | Interface |
| ILSP | Integrated Logistics Support Plan |
| IMS | Integrated Master Schedule |
| IOT&E | Independent Operational Test and Evaluation |
| IPDS | Integrated Product Development System |
| IPP | Integrated Program Plan |
| IR | Infrared |
| iRD | initial Requirements Document |
| IRD | Interface Requirements Document |
| ISO | International Organization for Standardization |
| ISRR | Initial System Requirements Review |
| ISSP | Integrated Safety System Program |

**J**

| | |
|---|---|
| JRC | Joint Resource Council |
| JRC2 | Joint Resource Council Second Meeting |

**L**

| | |
|---|---|
| L | Low |
| LCC | Life Cycle Cost |
| LCP | Life Cycle Plan |
| LSSI | Large-Scale System Integration |

**M**

| | |
|---|---|
| M | Medium |
| MA | Mission Analysis |
| MASPS | Minimum Aviation System Performance Standards |
| MND | Mission Need Decision |
| MNR | Mission Need Review |
| MNS | Mission Need Statement |
| MOE | Measures of Effectiveness |
| MOPS | Minimum Operational Performance Standards |
| MOS | Measures of Suitability |
| MRS | Mature Requirements Statement |
| MSAW | Minimum Safe Altitude Warning |
| MVP | Master Verification Plan |

**N**

| | |
|---|---|
| $N^2$ | N-squared |
| NAS | National Airspace System |
| NCP | NAS Change Proposal |
| NDI | Nondevelopmental Item |

**O**

| | |
|---|---|
| ODBC | Open Database Connectivity |
| OMB | Office of Management and Budget |
| OOA | Object-Oriented Analysis |
| ORD | Operational Requirements Document |
| OSA | Operational Safety Assessment |
| OSED | Operational Services and Environmental Description |
| OSHA | Occupational Safety and Health Administration |
| O&SHA | Operating and Support Hazard Analysis |
| OSI | Open System Interconnection |

**P**

| | |
|---|---|
| PBM | Process-Based Management |
| PCA | Physical Configuration Audit |
| PCS | Proposal Completion Schedule |
| PDR | Preliminary Design Review |
| PHA | Preliminary Hazard Analysis |
| PIN | Personal Identification Number |
| POC | Point of Contact |
| PRS | Primitive Requirements Statement |
| P-Spec | Process Specification |
| PUI | Program-Unique Identifier |

**Q**

| | |
|---|---|
| QFD | Quality Function Deployment |

**R**

| | |
|---|---|
| R | Red |
| RAA | Responsibility, Authority, and Accountability |
| RAM | Requirements Allocation Matrix |
| RCM | Reliability-Centered Maintenance |
| RDGT | Reliability Development Growth Testing |
| REVIC | Revised Intermediate COCOMO |
| RF | Radio Frequency |
| RFE | Request for Estimate |
| RFP | Request for Proposal |
| RMA | Reliability, Maintainability, Availability |
| ROM | Rough Order of Magnitude |
| RPG | Radar Product Generator |
| RRD | Risk Resolution Date |
| RVCD | Requirements Verification Compliance Document |

**S**

| | |
|---|---|
| SAE | Society of Automotive Engineers |
| SARP | Standards and Recommended Practices |

| | |
|---|---|
| SCM | Software Configuration Management |
| SCN | Specification Change Notice |
| SDP | Software Development Process |
| SDR | System Design Review |
| SDRL | Subcontractor Data Requirements List |
| SDRL | System Design Review List |
| SE | System Engineering |
| SEM | System Engineering Manual |
| SEMP | System Engineering Management Plan |
| SEMS | System Engineering Master Schedule |
| SFRA | System Functional Requirements Analysis |
| SHA | System Hazard Analysis |
| SI | Solution Implementation |
| SMP | Software Management Plan |
| SMP | System Management Plan |
| SOO | Statement of Objectives |
| SOW | Statement of Work |
| SRR | System Requirements Review |
| SRVT | Safety Requirements Verification Table |
| SSDD | System/Segment Design Document |
| SSE | System Safety Engineering |
| SSH | System Safety Handbook |
| SSHA | Subsystem Hazard Analysis |
| SSMP | System Safety Management Plan |
| SSPP | System Safety Program Plan |
| SSR | Software System Review |
| SSWG | System Safety Working Group |
| STD | Standard |
| STD | State Transition Diagram |
| SW | Software |

**T**

| | |
|---|---|
| T&E | Test and Evaluation |
| TLS | Time Line Sheet |
| TPM | Technical Performance Measure |
| TRB | Technical Review Board |
| TRR | Test Readiness Review |

**U**

| | |
|---|---|
| U | Unacceptable |
| UML | Unified Modeling Language |
| URL | Uniform Resource Locator |

**V**

| | |
|---|---|
| VFR | Visual Flight Rules |
| VOR | Very High Frequency Omnidirectional Range |
| VRR | Verification Readiness Review |
| VRTM | Verification Requirements Traceability Matrix |

**W**

| | |
|---|---|
| WARP | Weather and Radar Processor |
| WBS | Work Breakdown Structure |
| WMSCR | Weather Message Switching Center Replacement |
| WSP | Weather System Processor |

**Y**

| | |
|---|---|
| Y | Yellow |

# APPENDIX B: SYSTEM ENGINEERING MANUAL GLOSSARY

| TERM | DEFINITION |
|---|---|
| Acceptance Criteria | Various criteria that a system or component must satisfy in order to be accepted by a user, customer, or other authorized entity. |
| Allocation | Top-down distribution of system-level requirements to the subsystem, element, component, or to the project team that delegated to meet the requirement. This approach tends to promote a top-down "system approach" in helping to establish specific design requirements for all levels of the system hierarchy as appropriate. The allocation process can be properly applied with reliability, maintainability, supportability, lifecycle cost, and related characteristics in mind. Allocation is also the assignment of performance requirements to functions. |
| Analysis | Logical examination or study of a system to determine the nature, relationships, and interaction of its parts and environment. Analysis emphasizes baseline system performance and/or compares development, production, or usage alternatives. Analysis is concerned with understanding the existing system and establishing the system requirements.<br><br>Analysis is also a type of verification. It can be any kind of mathematical, computational, or logical task performed to verify a requirement that cannot be verified in any other manner, including simulation and similarity analyses. |
| Article | Any product, including systems, subsystems, elements, components, or parts. |
| Behavior Diagram | Graphical representation of system dynamics that incorporates system responses to inputs. A type of functional flow diagram. The behavior diagram differs from functional flow diagrams in general in that behavior diagrams contain data flow and control elements. (See Functional Flow Diagram.) |
| Compliance | Determination that the requirements have been met. |
| Component | A part of the product being designed or produced. |
| Concept of Operations | Description of what is expected from the system, including its various modes of operation and time-critical parameters. |
| Configuration Item | Aggregation of hardware, software, processed materials, services, or any of its discrete parts that is demonstrated for configuration management and treated as a single entity in the configuration management process. (ISO) |
| Configuration Management | A basic system engineering element. A management process for establishing and maintaining consistency of a product's performance, functional, and physical attributes with its requirements, design, and operational information through out its life. The CM program consists of |

| TERM | DEFINITION |
|---|---|
| | CM functions associated with the following program elements: program management, design requirements, document control, change control, and assessments. (See Section 4.11, Configuration Management.) |
| Critical Design Review (CDR) | Formal technical review of detail design documentation to establish compatibility with applicable requirements and interfaces and to identify specific engineering documentation required for release to production. |
| Data Dictionary | A definition of all system data representations in the system models that binds the models together. |
| Data Flow Diagram (DFD) | Graphical means for modeling the processes that transform data in a system. |
| Decomposition | Partitioning/dividing a requirement into its lower-level discrete elements or parts. |
| Demonstration | Type of verification. Similar to test except that it does not require instrumentation. |
| Derived Requirements | Any requirement that is not explicitly identified by the Customer. For example:<br>· Decision to select a separate power supply for equipment performing a specific function leads to derived safety requirements.<br>· Architectural choices, such as selecting hydraulic versus electrical power, would have different consequences and different requirements for achieving the same objective.<br>· Hardware-software interfaces. |
| Deviation | Specific, written authorization to depart from a particular requirement(s) of an item's current approved configuration documentation for a specific number of units or a specified period of time, and to accept an item that is found to depart from specified requirements, but nevertheless is considered suitable for use "as is" or after repair by an approved method. (A deviation differs from an engineering change in that an approved engineering change requires corresponding revision of the item's current approved configuration documentation, whereas a deviation does not allow a revision of the item's current approved configuration documentation.) |
| DOORS (Dynamic Object-Oriented Requirements System) | Commercial tool licensed to the FAA for capturing and managing requirements. |
| Effectivity | Designation defining the point in time, an event, or a product range (e.g., serial, lot number, model, date) at which changes or variances to specific products are to be effected. The authorized and documented point of usage for a specific configuration of a part/ assembly/installation, etc. |

| TERM | DEFINITION |
|---|---|
| Environment | Natural and induced conditions experienced by a system, including its people, product, and processes. |
| Function | Characteristic task, action, or activity that must be performed to achieve a desired system objective (or customer need). |
| Functional Analysis | One of the basic elements of system engineering. A process for examining a system need to identify all the functions and subfunctions necessary to accomplish the system's operation or mission. (See Section 4.4, Functional Analysis.) |
| Functional Architecture | Hierarchical arrangement of functions and interfaces providing a complete representation of the system from a performance and behavioral perspective, as captured in the requirements set. |
| Functional Baseline | Set of functions, functional interfaces, timelines, and requirement allocations established for a particular system. |
| Functional Configuration Audit (FCA) | Review to verify the functionality of subsystems. These reviews are also part of the reviews designed to accomplish certification. |
| Functional Decomposition | Approach to reducing functional complexity by allocating functionality and interfaces to sublevel functions, which are more readily understood and managed. |
| Functional Flow Diagram (or Functional Flow Block Diagram) | Multi-tier, time-sequenced, step-by-step diagram of the system functional flow. (See also Behavior Diagram.) |
| Functional Interface | Logical or physical association between functions that allows transmission of a quantity across a boundary. Quantities may include electrical, hydraulic, and pneumatic power; mechanical forces and torques; gases; heat; vibration, shock, and loads; data; and other quantities. |
| Functional Requirements | Requirements necessary to obtain the desired performance of a system under the conditions specified. |
| High-Level Requirements | Requirements applicable to the highest tier of the system architecture. |
| "ilities" | Specialty functions that contribute to the design, manufacture, and acceptable performance of the product (e.g., elements of specialty engineering: reliability, maintainability, human engineering, safety, supportability, etc.). |
| Inspection | Type of verification method. Verification of a requirement by visual examination. |
| Integrity of Analyses | One of the basic elements of system engineering. A disciplined process applied throughout a program to ensure that analyses provide the required levels of fidelity, accuracy, and confirmed results in a timely manner. Integrity is ensured by competent users iteratively applying a validated set of tools to a clearly defined data set. |

| TERM | DEFINITION |
|---|---|
| Integration | Bottom-up process of system buildup. The task of ensuring that all items work together individually and collectively as a group or as a whole system. |
| Interface | Functional and physical connection at a boundary. (See Section 4.7, Interface Management.) |
| Interface Control Document (ICD) | Document that provides basic information about interfaces between two elements, including type of interface (electrical, pneumatic, hydraulic, etc.) and the interface characteristics (functional or physical). |
| Interface Requirements Document (IRD) | Document that provides FAA interface requirements between two elements, including type of interface (electrical, pneumatic, hydraulic, etc.) and the interface characteristics (functional or physical). |
| Lifecycle | Entire spectrum of activity for a given system, commencing with the identification of a need and extending through system design and development, production and/or construction, operational use, sustaining support, and system retirement and phaseout. |
| Master Verification Plan (MVP) | Plan describing the overall verification program. (See Section 4.12, Validation and Verification.) |
| Mission Need Statement | Documentation of needs that is approved at JRC-1. |
| Model | Representation of an actual or conceptual system that involves mathematics, logical expressions, or computer simulations that can be used to predict how the system might perform or survive under various conditions or in a range of hostile environments. (See Simulation.) |
| $N^2$ Diagram | Visual matrix representing functional or physical interfaces between system elements. |
| Nonconformance | Failure of a unit or product to conform to specified requirements. |
| Operational Requirements Document (ORD) | Top-level requirements document normally provided by the customer. It is the intent of ORD to specify the requirements for all the operational aspects of the system. |
| Part | One, two, or more pieces joined together to make a component; these pieces are not normally subject to disassembly without destruction or impairment of designed use. |
| Performance | Quantitative measure characterizing a physical or functional attribute relating to the execution of an operation or function. Performance attributes include quantity (how many or how much), quality (how well), coverage (how much area, how far), timeliness (how responsive, how frequent), and readiness (availability, mission/operational readiness). Performance is an attribute for all systems, people, products, and processes, including those for development, production, verification, |

| TERM | DEFINITION |
|---|---|
| | deployment, operations, support, training, and disposal. Thus, supportability parameters, manufacturing process variability, reliability, and so forth are all performance measures. |
| Physical Architecture | Hierarchical arrangement of hardware and/or software components along with associated interfaces depicting the physical definition of the system. |
| Physical Configuration Audit (PCA) | Review to determine whether the aircraft was built in accordance with the drawings reviewed at the CDR. In addition, the PCA fulfills the requirements of the audit requirements of certification. |
| Preliminary Design Review (PDR) | Formal technical review of initial design concepts and documentation to establish compatibility with applicable requirements and to further define physical and functional interface requirements. |
| Product | Whole system or process being designed or produced. |
| Quality Function Deployment (QFD) | Method for capturing and delineating requirements based on identifying what is desired by the customer or stakeholder, along with how that desire will be satisfied. |
| Record | Information or data on a particular subject that is collected and input into a system for electronic storage. |
| Reliability | Ability of a system and its parts to perform its mission without failure, degradation, or demand on the support system. |
| Requirements Analysis | Basic element of the system engineering process. (See Section 4.3, Requirements Management.) |
| Requirements Document | Collection of requirements and related information/attributes presented in a user-defined format. These documents, when output from the Requirements Management process, are called requirements documents. Examples of requirements documents are an initial Requirements Document (iRD), final requirements document (fRD), discrete performance and procurement specifications, and requirements traceability matrices. |
| Risk | Undesirable situation or circumstance that has a realistic probability of occurring and an unfavorable consequence. (See Section 4.10, Risk Management.) |
| Rule | Standard procedure that governs a task or record through its lifecycle. |
| Similarity | Type of verification by analysis. Applicable to components and subsystems similar in characteristics and usage to those on previous systems. In principle, there are no parts of the subject subsystem more at risk (due to environment or installation), and operational stresses are no more severe than on previous systems. |
| Simulation | Type of verification by analysis. The verification of a system requirement by a computer simulation or other technique. Simulation also includes hardware-in-the-loop simulations. |

| TERM | DEFINITION |
|---|---|
| | Execution of a system model to examine the response of the system to injected inputs, usually performed before development of system hardware and software. |
| Stakeholder | Entity (e.g., person, team, or product) that is responsible for or in some way has a vested interest in the requirement or product under consideration. |
| State Transition Diagram | Graphical means of modeling the dynamic behavior of a system by depicting the legal states that the system may assume. |
| Structured Analysis | Disciplined approach to defining a system using a graphical box-and-arrow diagramming language. |
| Synthesis | A basic element of the system engineering process. A process for identifying one or more physical solutions or embodiments of functionality identified in the Functional Analysis process and associated requirements set. (See Section 4.5, Synthesis, and Section 4.6, Trade Studies.) |
| System | An integrated set of constituent pieces that are combined in an operational or support environment to accomplish a defined objective. These pieces include people, hardware, software, firmware, information, procedures, facilities, services, and other support facets. |
| System Design Review (SDR) | A review of the overall system configuration in lieu of or in addition to individual reviews of equipment items, software, and other system components.<br>The system design review covers:<br>1) Functional analysis and allocation of requirements<br>2) Development, process product, and material specifications<br>3) Design data defining the overall system (layouts, drawings, parts/material lists, supplier data)<br>4) Analyses, reports predictions, tradeoff studies, and related design documentation<br>5) Assessment of the proposed system design configuration in terms of technical performance measures<br>6) Individual program/design plans |
| System Environment | All elements and interfaces external to a system from which the system receives inputs and to which the system delivers outputs. |
| System Engineer | Individual who concentrates on the design and application of the whole (system), as distinct from the parts, and who looks at a problem in its entirety, taking into account all the facets and all the variables and relating the social to the technical aspects. |
| System Requirements | A review to verify that all the top-level requirements are correct; that is, |

| TERM | DEFINITION |
|---|---|
| Review (SRR) | that they meet with customer approval. Another SRR function is to present to the customer those "assumed" requirements that have been developed throughout the requirements development. |
| Technical Performance Measurement (TPM) | Continuing verification of the degree of anticipated and actual achievement of technical parameter growth toward expected values. |
| Test | Type of verification that requires instrumentation. Includes both laboratory and flight tests. |
| Thread | A system input, system output, description of the transformations to be performed, and the conditions under which these transformations should occur. |
| Time-Critical Functions | Functions that affect reaction time, downtime, or system availability. |
| Time-Critical Requirement | An identified, temporal constraint on or characteristic of the system. |
| Time Line Analysis | Graphical representation that considers functional duration and provides a description of the functional sequences for operation, test, and maintenance functions. |
| Time Line Sheet | Used to perform and record the analysis of time-critical functions and functional sequences. |
| Traceability | Characteristic by which requirements at one level of design may be related to requirements at another level. Traceability also encompasses the relationship between a performance requirement and the function from which the performance requirement was derived. |
| Trade Study | Analysis conducted to determine the preferred option, given two or more options. Trade studies can be either top-level or subsystem-level. |
| Validation | Determination that the requirements for a product are sufficiently correct and complete. (See Section 4.12, Validation and Verification.) |
| Variance | Specific, written authorization to depart from a particular requirement(s) of a product's current approved configuration documentation for a specific number of units or a specified period of time. (A variance differs from an engineering change in that an approved engineering change requires corresponding revision of the product's current approved configuration documentation, whereas a variance does not.) |
| Verification | Evaluation of an implementation [system] to determine that applicable requirements are met (See Test, Demonstration, Analysis, and Inspection. Verification for a given requirement can include one or more of these methods. See Section 4.12, Validation and Verification.) |
| Verification Requirements Traceability Matrix (VRTM) | Matrix correlating requirements and the associated verification method(s). (See Section 4.12, Validation and Verification.) |
| Waiver | Written authorization to accept an item, which during manufacture, or |

| TERM | DEFINITION |
|---|---|
| | after having been submitted for inspection or acceptance, is found to depart from specified requirements, but nevertheless is considered suitable for use "as is" or after repair by an approved method. |
| Working Groups | Cross component groups chartered with the task of working process, design, and development tasks for any common system. |

# APPENDIX C: INITIAL SYSTEM REQUIREMENTS REVIEW CHECKLIST

The purpose of the Initial System Requirements Review (ISRR) Checklist is to ascertain the adequacy of the efforts in defining system requirements and to determine initial direction and progress of the System Engineering (SE) management effort and the convergence to an optimum and complete system configuration. The ISRR assesses the understanding of the contract requirements documents.

**Checklist:**

1. Has the Joint Resources Council (JRC) approved the Mission Need Statement (MNS)?
2. Is the MNS validated and in accordance with the Acquisition Management System (AMS) SE guidance?
3. Is the Concept of Operations (CONOPS) complete and in accordance with AMS SE guidance?
4. Is the Functional Analysis complete to a level equivalent to requirements development?
5. Is the Functional Analysis in accordance with the AMS SE guidance?
6. Are all requirements traceable to a function or functions defined in The Functional Architecture?
7. Have all requirements been validated in accordance with the AMS SE guidance?
8. Are all requirements written in accordance with AMS SE guidance?
9. Did the Specialty Engineering disciplines participate in the validation of requirements?
10. Were the candidate requirements identified in Specialty Engineering Design Analysis Reports (DAR) considered in the validation of requirements?
11. Are all interfaces identified in accordance with AMS SE guidance?
12. Is each interface traceable to an associated set of requirements?
13. Has the Master Verification Plan (MVP) been developed in accordance with AMS SE guidance?
14. Has the MVP been coordinated and approved by the appropriate approval authority?

# APPENDIX D: CONCERNS/ISSUES

While performing any System Engineering process, the specialist may encounter issues and concerns to surface. These issues and concerns may take many forms, but they usually consist of potential risks to the program. Risk Management (Section 4.10) addresses this topic and is consulted when problems arise. The issues and concerns are collected in a form to use during the Risk Management process to determine if they are a threat to program success. At a minimum, the following information concerning each issue or concern shall be derived or collected:

- Title of issue or concern
- Problem statement
- Causes
- Potential effect on the program
- Who is identifying the issue or concern
- Contact information, such as telephone number or e-mail address

**Problem Statements**

If a problem arises, a problem statement of one to two sentences shall be generated that succinctly identifies the problem and answers the following questions:

- What is the problem?
- What is the scope of the problem?

The following tips may be used as a guide to develop a problem statement:

- Avoid using jargon
- Focus on the specific problem or the issue; problems are characterized by a need, a shortfall in capability, or a threat
- Avoid confusing symptoms or causes with the problem. Focusing on symptoms or causes diverts resources from solving the real or entire issue
- When possible, use data to support the existence of the problem or issue
- Make a connection between the issue and the organization
- Carefully read and analyze the problem statement. Discuss the problem within a peer group to enhance the overall understanding of the problem. A peer group effort is more effective in identifying the key factors in this type of problem-solving situation. The peer group actively searches for the information necessary to solve the problem
- List what is known. Start a list to record everything known about the situation. Begin with the information contained in the problem statement and add the knowledge that the peer group brings
- Record information that people think that they know but are unsure
- List what is needed. Prepare a list of questions that need to be answered to solve the problem. Record them under a second list: “What do we need to know?” Several types of questions may be appropriate. Some may address concepts or principles that need to be learned in order to address the situation. Other questions may be in the form of

requests for more information. These questions will guide future searches for information

- List possible actions, such as recommendations, solutions, or hypotheses under the heading “What should be done?”

# APPENDIX D: CONCERNS/ISSUES

While performing any System Engineering process, the specialist may encounter issues and concerns to surface. These issues and concerns may take many forms, but they usually consist of potential risks to the program. Risk Management (Section 4.10) addresses this topic and is consulted when problems arise. The issues and concerns are collected in a form to use during the Risk Management process to determine if they are a threat to program success. At a minimum, the following information concerning each issue or concern shall be derived or collected:

- Title of issue or concern
- Problem statement
- Causes
- Potential effect on the program
- Who is identifying the issue or concern
- Contact information, such as telephone number or e-mail address

**Problem Statements**

If a problem arises, a problem statement of one to two sentences shall be generated that succinctly identifies the problem and answers the following questions:

- What is the problem?
- What is the scope of the problem?

The following tips may be used as a guide to develop a problem statement:

- Avoid using jargon
- Focus on the specific problem or the issue; problems are characterized by a need, a shortfall in capability, or a threat
- Avoid confusing symptoms or causes with the problem. Focusing on symptoms or causes diverts resources from solving the real or entire issue
- When possible, use data to support the existence of the problem or issue
- Make a connection between the issue and the organization
- Carefully read and analyze the problem statement. Discuss the problem within a peer group to enhance the overall understanding of the problem. A peer group effort is more effective in identifying the key factors in this type of problem-solving situation. The peer group actively searches for the information necessary to solve the problem
- List what is known. Start a list to record everything known about the situation. Begin with the information contained in the problem statement and add the knowledge that the peer group brings
- Record information that people think that they know but are unsure
- List what is needed. Prepare a list of questions that need to be answered to solve the problem. Record them under a second list: “What do we need to know?” Several types of questions may be appropriate. Some may address concepts or principles that need to be learned in order to address the situation. Other questions may be in the form of

requests for more information. These questions will guide future searches for information

- List possible actions, such as recommendations, solutions, or hypotheses under the heading “What should be done?”

# Appendix E Integrated Technical Planning Details

## E.1 Integrated Technical Planning – Reserved

## E.2 Requirements Management Planning

This planning specifies the tasks, products, responsibilities, and schedule for managing requirements throughout product development. The planning begins in the early stages of Investment Analysis and SEMP development and is baselined at the JRC-2B and is updated as necessary at subsequent exit reviews.

The planning section details the total effort in managing requirements. The work includes identifying and capturing requirements (Paragraph 4.3.3.1), analyzing and decomposing requirements (Paragraph 4.3.3.2), and allocating requirements (Paragraph 4.3.3.3).

### E.2.1 Inputs to Requirements Management Planning

The following inputs are normally required for the planning section:

- Internal and external requirements as defined in Paragraph 4.3.1
- Component-specific program guidelines
- Program-specific organizational constraints and assumptions to be used in the program
- Program-specific schedule constraints and events
- Top-level conceptual alternatives, functional analyses, design support alternatives, and initial system evaluations
- Technology availability or constraints
  - Captures those technologies for which requirements necessary to meet requirements and the resulting derived requirements from them
  - Constraints identify the envelope of the technology operation
  - Inputs may include identification of key technologies, performance, maturity, cost, and risks, as well as technology breakthroughs and forecasts
- Derived requirements, which are developed through trade studies and are not provided by external sources, such as the stakeholder or government policies
- Outputs from each stage of the program life cycle
- Concepts of the product (e.g., operational, maintenance, support, logistics)
- New or revised directions and limitations established by the acquisition decision authority
- Records of meetings, conversations, and agreements with stakeholders, and internal functions relating to documented changes

### E.2.2 Requirements Management Planning Steps

Following are the steps in producing a planning section, which is normally coordinated and written by an SE group.

#### E.2.2.1 Step 1: Collect Inputs

All program organizations that develop and manage requirements are responsible for providing planning section inputs to the planning coordinator.

#### E.2.2.2 Step 2: Prepare Planning Section

The planning coordinator prepares the planning section. Although no standard format exists for developing the section, it is recommended that the section contain the key elements of tasks, deliverables, responsibilities, and schedule. Developing a standard format may be included in this step. The section provides for deviations from the Requirements Management process (Section 4.3).

#### E.2.2.3 Step 3: Coordinate and Baseline

The planning coordinator provides drafts of the planning section to all stakeholders for review, and the version approved at the JRC-2B becomes the baseline planning section.

#### E.2.2.4 Step 4: Maintain Planning section

The planning coordinator monitors the program's progress continually throughout the life of the program, and any program changes in the program are reflected in the planning section.

#### E.2.2.5 Step 5: Provide Current Planning Section

The planning coordinator provides the planning section to all stakeholders (including, at a minimum, the program manager, users, and project leaders) required to manage by the planning section.

### E.2.3 Outputs of Requirements Management Planning

The following outputs are normally required for the planning section.

#### E.2.3.1 Requirements Management Planning Tasks

It is recommended that the tasks to be described in the planning section reflect the processes detailed in Requirements Management (Section 4.3).

The other two subprocesses in the Requirements Management Process—Develop Verification Approach and Analyze Verification Data—are the subjects of the Verification process in Section 4.12.

#### E.2.3.2 Requirements Management Planning Products

A key function of the planning section is to define the products of the Requirements Management process. Another key function of the planning section is to assign responsibilities to various subprocesses within the Requirements Management process (Section 4.3).

#### E.2.3.3 Requirements Management Planning Schedule

A function of the planning section is to provide a schedule of the requirements management tasks. See Section 4.3 for a description of the schedule considerations.

### E.2.4 Requirements Management Planning Metrics

The primary planning metric is the publication and approval of the planning section at the Investment Analysis, phase one, exit review and the updating at subsequent reviews. A metric of the requirements process is the number of requirements identified after SDR. This metric may also apply to the planning section as well, since it reflects the quality of the program planning.

### E.2.5 Requirements Management Planning Tools

A word processing tool and DOORS are needed.

A sample outline for a requirements management planning section appears in Table 8-1. Also it is recommended that, the planning section be developed in accordance with the Requirements Management process described in Section 4.3 and reflect the principles reflected in government and industry standards, such as MIL-STD-961 or -490 for specifications, EIA 632 for the SE process, and ARP 4754 for commercial aircraft development. Although QFD is a tool for requirements development, it is also a tool for planning, since it incorporates a unique set of steps. The outline (Table 8-1) depicts the recommended contents of the Requirements Management planning section.

**Table 8-1 Table of Contents Requirements Management Section of SEMP**

| Requirements Management Planning Section Example Outline | | |
|---|---|---|
| 1 | SCOPE | |
| 1.1 | Document Overview | |
| 1.2 | Process Overview | This section contains a diagram showing the interrelationship between the various process elements, including the requirements management tool, if any. |
| 2 | APPLICABLE DOCUMENTS | |
| 3 | TASKS | The tasks described are tied to the specific organizational and program requirements in accordance with Section 4.3. |
| 3.1 | Identify and Capture Requirements | |
| 3.2 | Analyze and Decompose Requirements | |
| 3.3 | Allocate Requirements | |
| 3.4 | Derive Requirements | |
| 3.5 | Manage Requirements Changes | |
| 4 | PRODUCTS | This section describes the various program requirements documents. The section describes what organizational entity is the recipient of the product; for example, the product team, stakeholder, other project teams, company management, or outside organizations, such as |

| Requirements Management Planning Section Example Outline | | |
|---|---|---|
| | | manufacturing, product support, test and evaluation, or supplier management. |
| 4.1 | Requirements Documents | This section enumerates and describes the various program requirements documents to be produced. |
| 4.2 | Requirements Allocation Matrices | This section describes the characteristics of the requirements allocation sheets (RASs) to be produced on this program. |
| 5 | RESPONSIBILITIES | This section details responsibilities of the various organizational entities to accomplish the tasks of Section 3. The responsibilities are to be tied to the tasks of Section 3. |
| 6 | SCHEDULE | The schedule shown in this section is to be tied to the milestones of the IPP. |
| 7 | AUTOMATED REQUIREMENTS TOOL | This section describes the planned use of the requirements management tool, if any. |
| 8 | NOTES | |
| | APPENDICES | |

## E.3 Functional Analysis Planning

The Functional Analysis planning section of the SEMP specifies the tasks, products, responsibilities, and schedule for functional analysis throughout the development of the product. Because there is no program level SEMP in the early phases of the program (i.e., phase 1 of Investment Analysis), Functional Analysis in these phases is guided by the NAS-level SEMP. When the IPP is developed, the Functional Analyses is guided by the program's tailored SEMP. The planning section is baselined at the JRC-2B and is updated as necessary at subsequent exit reviews. This planning section details the total effort for managing functional analysis. This work includes analysis of the concept of operations and environment, the decomposition of functions into subfunctions, decomposing and allocating requirements to functions, evaluating alternative decompositions, defining functional sequences and timelines, defining functional interfaces, and documenting the functional baseline. The outline (Table 8-2) depicts the recommended contents of the FA planning section.

### E.3.1.1 Inputs to Functional Analysis Planning

The following inputs are normally required for planning:

- MNS and fRD, which detail the system's expected operational environments
- Component-specific program guidelines
- Program-specific constraints and assumptions, such as nature of the program's project teams
- Program-specific schedule constraints and events

- NAS SEMP, which provides the overall plan for conducting SE as part of NAS modernization

#### E.3.1.2 Functional Analysis Planning Steps

The planning section is normally coordinated and written by an SE group. Following are the steps in producing this section.

##### E.3.1.2.1 Step 1: Collect Inputs

All program organizations developing and managing requirements are responsible for providing planning inputs to the planning coordinator.

##### E.3.1.2.2 Step 2: Prepare Planning Section

The planning coordinator prepares the planning section. No standard format exists for developing the section; however, it is recommended that the section contain the key elements of tasks, deliverables, responsibilities, and schedule. The plan provides for justification and deviations from the Functional Analysis process (Section 4.4).

##### E.3.1.2.3 Step 3: Coordinate and Baseline

The plan coordinator provides drafts of the plan for review. The version approved at JRC-2A becomes the baseline plan.

##### E.3.1.2.4 Step 4: Maintain Planning Section

The plan coordinator maintains continuous cognizance of the program progress throughout the life of the program, and changes in the program are reflected in the planning section.

##### E.3.1.2.5 Step 5: Provide Current Planning Section

The plan coordinator provides the planning section to all parties required to manage this section. At a minimum, these organizations include the program manager, the stakeholders, and project leaders.

#### E.3.1.3 Outputs of Functional Analysis Planning

The following outputs are normally required for the planning section.

##### E.3.1.3.1 Functional Analysis Planning Tasks

It is recommended the tasks described in the planning section reflect the processes described in Functional Analysis (Section 4.4). These processes are as follows:

- Define the operational mission, environment, and requirements
- Define top-level functions
- Organize functions into logical relationships
- Decompose functions Into subfunctions
- Define internal and external interfaces
- Evaluate alternative decompositions

- Define sequences and timelines
- Complete functional architecture

#### E.3.1.3.2 Functional Analysis Planning Products

The products of the functional analysis plan are the (a) functional architecture, (b) Concept of Operations (CONOPS), and (c) Issues and Concerns.

##### E.3.1.3.2.1 Functional Architecture

The functional architecture primarily is in the form of functional flow diagrams and/or timeline sequences produced in accordance with the directions contained in Section 4.4. The architecture contains a description of the system's functions and their inter-relationships, as well as a description of the functional interfaces and the functional sequences or timelines.

Functional architecture development is conducted in relation to requirements. As requirements are developed in increasing detail, the functional architecture is also developed in like detail. This means that as long as requirements are being developed, so is the functional architecture. Though the functional analysis continues in detail, the functional architecture is baselined at the Internal System Requirements Review or JRC-2B, whichever occurs first. This baseline is a functional description of the proposed system as it is described in the system-level specification.

#### E.3.1.3.3 Functional Analysis Planning Responsibilities

A key function of the planning is to assign responsibilities to various subprocesses within the Functional Analysis process. In general, one organization (or person) executes the process. In addition, within each process, one organization (or person) is responsible for specific tasks or decisions within that process. The discussion below gives guidance in assigning responsibilities to the various subprocesses. In the end, these assignments may vary greatly according to the product and the organization.

In general, SE assisted by design, support, program management, and stakeholders, normally performs functional analysis. SE also normally has ownership of the electronic tool with the functional analysis capability.

##### E.3.1.3.3.1 Responsibility for the Define the Operational Mission, Environment, and Requirements Subprocess

SE normally has overall responsibility for this process; however, the process is to be conducted in close cooperation with the stakeholder.

##### E.3.1.3.3.2 Responsibility for the Define Top-Level Functions Subprocess

In this process, the operational mission, environment, and existing requirements (including needs) are transformed into the required functions, which are listed.

##### E.3.1.3.3.3 Responsibility for the Organize Functions Into Logical Relationships Subprocess

The functions listed (see Paragraph 4.3.4.4.1.3.3.2) are organized into logical (input-function-output) and/or sequence relationships.

#### E.3.1.3.3.4 Responsibility for the Decompose Functions Into Subfunctions Subprocess

This process decomposes functions into subfunctions to a level at which the requirements associated with a specific function may be allocated to specific elements of equipment, software, personnel, procedures, and facilities. The process is normally an SE responsibility with assistance from designers.

#### E.3.1.3.3.5 Responsibility for the Define Internal and External Interfaces Subprocess

System engineers, assisted by designers, normally conduct this subprocess also. The process shall also include participation of the Interface Control Working Group (ICWG), discussed in Interface Management (Section 4.7).

#### E.3.1.3.3.6 Responsibility for the Evaluate Alternative Decompositions Subprocess

In this subprocess, the functions are broken down further in increasing detail consistent with the further development of requirements.

#### E.3.1.3.3.7 Responsibility for the Define Sequences and Timelines Subprocess

SE also normally conducts this subprocess, assisted by design personnel, operations, and the stakeholders.

#### E.3.1.3.3.8 Responsibility for the Complete Functional Architecture Subprocess

System engineers, who own the electronic tool on which it is produced, publish the functional architecture. The output of this subprocess shall support interaction between the Requirements Management process (Section 4.3) and this Functional Analysis (Section 4.4). Design personnel within each project team are normally responsible for assigning performance requirements to specific functions and subfunctions. SE records these allocated requirements in the requirements electronic database.

### E.3.1.3.4 Functional Analysis Planning Schedule

A planning function is to provide a schedule of the functional analysis tasks. It is recommended that the schedule show the delivery dates of the product, namely, the functional architecture. The schedule, which provides the necessary sequence of events, needs to identify the task start dates and end dates and key them to the events outlined in the IPP template of Figure 4.2-3. Functional analysis is normally accomplished at specific levels in the AMS phases discussed below.

#### E.3.1.3.4.1 Mission Analysis Through Define Mission Need

It is recommended that, prior to the IA, modes of operation and the top-level functional architecture be established and functional interfaces with other systems identified. The top-level functional architecture is developed during this phase.

#### E.3.1.3.4.2 Investment Analysis—Alternatives Assessment

It is recommended that, prior to the ISRR, the functional architecture be established to the second level of the system architecture by decomposing the top-level functions. Prior to the SDR, the functional architecture is developed to the lowest level of the functional architecture by further decomposition.

#### E.3.1.3.4.3 Investment Analysis—Requirements Baseline

Prior to the JRC-2B, the functional architecture is developed to the third (or lower) level of the system architecture by further decomposition of the functions.

#### E.3.1.3.4.4 Solution Implementation

The contractor or vendor, with government assistance as directed by the contract, is to continue the functional decomposition in support of defining increasingly detailed requirements and specifications.

### E.3.1.4 Functional Analysis Planning Metrics

The primary planning metric is the publication and approval of the planning at the JRC-2B phase exit review and the updating at subsequent reviews.

### E.3.1.5 Functional Analysis Planning Tools

No templates or standards currently exist for this planning. However, the planning section is developed in accordance with the Functional Analysis process (Section 4.4).

**Table 8-2 Table of Contents Functional Analysis Planning Section of SEMP**

| Functional Analysis Planning Section Example Outline | | |
|---|---|---|
| 1 | SCOPE | |
| 1.1 | Document Overview | |
| 1.2 | Process Overview | This section contains a diagram showing the interrelationship between the various process elements, including tools, if any. |
| 2 | APPLICABLE DOCUMENTS | |
| 3 | TASKS | The tasks described are tied to the specific organizational and program requirements in accordance with Section 4.4. |
| 4 | PRODUCTS | This section describes the various functional analysis outputs in accordance with Paragraph 4.4. The section describes what organizational entity is the recipient of the product; for example, the product team, stakeholder, other project teams, company management, or outside organizations, such as manufacturing, product support, test and evaluation, or supplier management. |
| 5 | RESPONSIBILITIES | This section details responsibilities of the various organizational entities to accomplish the tasks of Section 3. The responsibilities are to be tied to the tasks of Section 3. |
| 6 | SCHEDULE | The schedule shown in this section is to be tied to the milestones of the IPP. |
| 7 | AUTOMATED REQUIREMENTS TOOL | This section describes the planned use of the requirements management tool, if any. |

| Functional Analysis Planning Section Example Outline | | |
|---|---|---|
| 8 | NOTES | |
| | APPENDICES | |

## E.4 Synthesis Planning – Reserved

## E.5 Trade Studies Planning

The Trade Study planning documents the formal management planning regarding how alternative solutions to a problem or design issue associated with a program/project product development is to be assessed in a fair and impartial manner.

Trade study planning:

- Provides the formats for how trade study results and information are to be presented to management at design reviews
- Identifies the organization or person designated to be the trade study leader
- Identifies any tools that are to be used in performing of the trade study (i.e., cost models, computer simulations, test articles and fixtures, analytical tools)
- Provides the criteria (including constraints) under which the trade study is to be conducted
- Provides instructions on where trade study results and data are to be stored for future reference and which organization is responsible for maintaining the data

The outline (Table 8-3) depicts the recommended contents of the TS planning section.

### E.5.1 Inputs to Trade Study Planning

The following inputs are typically required in preparing the trade study planning section. Other program/project-unique inputs may exist and be considered as appropriate.

- Definition of the problem that is to be studied
- Program/project schedule
- Program/project requirements
- Document preparation tools

### E.5.2 Trade Study Planning Steps

#### E.5.2.1 Step 1: Collect Inputs

Coordinate with the program technical groups to obtain input information, including source data.

#### E.5.2.2 Step 2: Analyse Inputs

Review and organize input data.

#### E.5.2.3 Step 3: Define Activities and Effort

Work with the technical experts to document trade study activities.

#### E.5.2.4 Step 4: Lay Out and Baseline Section

Develop and coordinate the draft planning section, obtain necessary approvals (program management, senior technical experts, etc.), and release the baseline version of the SEMP.

#### E.5.2.5 Step 5: Interface With Other Processes

Coordinate and interface with other processes throughout planning.

### E.5.3 Outputs of Trade Study Planning

The output is a trade study planning section that includes the items described in the following paragraphs.

#### E.5.3.1 Problem Definition

A clear statement of the problem to be solved by a trade study is required to properly focus the efforts of participants. The problem is usually associated with meeting a specific requirement. The requirement needs to be defined to a level of detail that is appropriate for the project's current product development phase. In addition, it is recommended that any related requirements be listed that may be affected by the trade study. Stakeholder agreement is to be established on all high-level performance or mission requirements before the trade study is conducted.

#### E.5.3.2 Evaluation Criteria

A key step in eliminating or minimizing bias in trade studies is to define a consistent set of evaluation criteria before the trade study is started. Technical evaluation criteria are to reflect all technical requirements, and effective evaluation criteria shall:

- Differentiate meaningfully between alternatives without bias
- Relate directly to the purpose of the trade study (i.e., they are requirements-based)
- Be broadly based to ensure coverage of all decision factors
- Be independent of each other as much as possible
- Be universally understood by all trade study participants

#### E.5.3.3 Alternative Solutions

A broad set of alternative solutions needs to be developed prior to any evaluation is conducted. It is recommended that the affected disciplines conduct brainstorming sessions to develop a large number of alternatives for the trade study and that the trade study leader provide background information to all trade study participants before the brainstorming sessions.

#### E.5.3.4 Trade Study Tools

It is recommended that tools compatible with the problem under study be selected before the trade study is conducted.

#### E.5.3.5 Trade Study Schedule

It is recommended that a schedule be developed that identifies personnel responsible and due dates for completing each task associated with the trade study. The schedule is designed to support the overall program/project integrated master schedule.

### E.5.4 Trade Study Planning Metrics

The metric for measuring the product of this process is completion of the planning section. Also, the cost to produce and update the section may be measured.

### E.5.5 Trade Study Planning Tools

A word processing tool is needed.

**Table 8-3 Table of Contents Trade Studies Planning Section of SEMP**

| Functional Analysis Planning Section Example Outline | | |
|---|---|---|
| 1 | SCOPE | |
| 1.1 | Document Overview | |
| 1.2 | Process Overview | This section contains a diagram showing the interrelationship between the various process elements, including tools, if any. |
| 2 | APPLICABLE DOCUMENTS | |
| 3 | TASKS | The tasks described are tied to the specific organizational and program requirements in accordance with Section 4.6. |
| 4 | PRODUCTS | This section describes … |
| 5 | RESPONSIBILITIES | This section details responsibilities of the various organizational entities to accomplish the tasks of … |
| 6 | SCHEDULE | The schedule shown in this section is to be tied to the milestones of the IPP. |
| 7 | AUTOMATED REQUIREMENTS TOOL | This section describes the planned use of tools… |
| 8 | NOTES | |
| | APPENDICES | |

## E.6 Interface Management Planning

Interface management (IM) planning ensures establishment of the formal management system of interface (I/F) controls that enable physical and functional compatibility between interfacing hardware, software, personnel, and facilities. This planning:

- Provides the means for identifying, defining, documenting, and controlling the interfaces at all levels of the system
- Provides the means for changing the interfaces as required by the evolution of the design and for resolving interface incompatibilities

- Guides management, control, and documentation of all system functional and physical interfaces
- Establishes the Interface Control Working Group (ICWG) and its policies and procedures
- Contains requirements and templates for preparing, revising, and processing the interface documentation; identifies products
- Establishes the participants of the I/F management process and their responsibilities
- Establishes the interface management schedule

The ICWG Chairman drafts the IM planning policies and procedures in the early phase of Investment Analysis concurrent with the IPP Schedule. The ICWG Chairman updates and reviews the IM planning section of the SEMP to reflect the system functional and physical architectures developed in later phase of Investment Analysis.

### E.6.1 Inputs to Interface Management Planning

There are several inputs typically required to prepare the IM planning section. A description of each input follows along with a short justification and the sources of the input. Other unique program inputs may exist that are relevant to the preparation of the IM planning section. As appropriate, it is recommended that these be included:

- **IPP and IMS**
  - SE and Program Management prepare the IPP and IMS. The IPP and IMS are required to enable preparation of the I/F management schedule and to ensure coherent, complete, consistent and timely I/F design at all levels of the system.
- **SEMP**
  - SE prepares the SEMP. IM planning is an important adjunct to the SEMP. The IM planning section depends on products defined and scheduled by the SEMP and is therefore partially driven by it.
- **SE Schedule, which is prepared by SE**
  - The IM planning schedule and products are SEMP-driven.
- **System Requirements Documents**
  - SE generally prepares these documents, but they also are prepared occasionally by stakeholders or outside agencies. The documents define the system external interfaces and the (internal) interfaces between the system segments. They are therefore an important point of departure and basis for planning and controlling the system interfaces.
- **System Functional and Physical Architecture**
  - The system architectures are prepared early in the Investment Analysis through comprehensive trade studies of alternative configuration studies. The architectures determine where the system/segment interfaces exist and are the point of departure for the detailed identification and definition of the interfaces. The architectures also are the basis for allocating responsibilities of interfaces.
- **Design Review Plans**

- These plans are to be used as the bases for conducting reviews and audits of the interfaces. The corporate design review plans specify the required status of development of interfaces at the various prescribed design review milestones.

### E.6.2 Interface Management Planning Steps

Following are the major steps required to develop IM planning.

#### E.6.2.1 Step 1: Appoint ICWG Chairman

The program management generally appoints the ICWG Chairman, who is the key person in the I/F definition and control process. This individual is identified early in the program because he/she is chartered with the responsibility of developing and establishing the policies and process for identifying, defining, documenting, auditing, and controlling interfaces.

#### E.6.2.2 Step 2: Collect Inputs

Collect the inputs identified in Paragraph E.6.1.1.

#### E.6.2.3 Step 3: Analyze Inputs

Review, analyze, and organize the inputs collected. The interfaces and constraints embedded in the requirement documents and the system architectures are to be evaluated and assimilated and used as bases for establishing interfaces and responsibilities, as well as to determine if there are program-peculiar interfaces that need special treatment/attention. The planning sections and schedules are to be used as bases for constructing the interface management schedule.

#### E.6.2.4 Step 4: Define Activities and Effort

Establish the ICWG policies and procedures; delineate and coordinate the processes to be applied for identifying, defining, documenting, changing, auditing, and controlling interfaces; identify the responsibilities of participants; and identify standard formats to be used for documenting interfaces and their change process.

#### E.6.2.5 Step 5: Lay Out and Baseline

Prepare the IM planning section, which captures the processes, formats, schedule, and responsibilities. The processes and formats embedded in the IM planning section of the SEMP shall be consistent with the IPP. Using the IPP and IPS, and the SEMP and SE Schedule as bases, prepare an interface management schedule. The schedule may include all significant control and audit milestones defined by the corporate design review processes.

#### E.6.2.6 Step 6: Interface With Other Processes

The IM planning section of the SEMP shall be coordinated with the IPP and SE schedule and the design review planning sections.

#### E.6.2.7 Step 7: Update/Maintain the Planning Section

The ICWG Chairman shall review the IM planning section of the SEMP at the beginning of each of the AMS phases to determine if adjustments to the processes and schedules are required to ease or ensure effective fulfillment of the objectives of that phase.

### E.6.3 Outputs of Interface Management Planning

The principal output is an IM planning section of the SEMP delineating the I/F identification, definition, documentation, approval, change, and control and audit process. In addition, the IM planning section establishes the ICWG and its policy and procedures, constituents, and constituents' responsibilities.

### E.6.4 Interface Management Planning Metrics

The IM planning section is to be reviewed to ensure completeness and cohesiveness. The interface management schedule and products are to be reviewed for consistency with the rest of the SEMP and SE schedule.

### E.6.5 Interface Management Planning Tools

A word processing tool is needed.

To facilitate preparation of the IM planning section of the SEMP, refer to all applicable sections of the System Engineering Manual. The outline (Table 8-4) depicts the recommended contents of the IM planning section.

**Table 8-4. Interface Management Planning Section Outline of SEMP**

| Interface Management Planning Section Outline | |
|---|---|
| 1 | SCOPE |
| 1.1 | Document Overview |
| 1.2 | System Overview |
| 2 | APPLICABLE DOCUMENTS |
| 3 | INTERFACE CONTROL WORKING GROUP |
| 3.1 | ICWG Policy and Procedures |
| 3.2 | ICWG Membership and Responsibilities |
| 3.2.1 | ICWG Chairman |
| 3.2.2 | Interface Custodian |
| 3.2.3 | Interface Participant |
| 4 | INTERFACE CONTROL PROCESS |
| 4.1 | Establishing Interfaces |
| 4.1.1 | Identifying Interfaces |
| 4.1.1.1 | Scope Sheet |
| 4.1.1.2 | Documenting ICDs |
| 4.1.1.3 | Coordinating Interfaces |
| 4.1.1.4 | Auditing, Statusing, and Controlling ICDs |
| 4.1.1.4.1 | Authorized ICD List |
| 4.1.1.4.2 | Review at SRR |

| Interface Management Planning Section Outline | |
|---|---|
| 4.1.1.4.3 | Review at SDR |
| 4.1.1.4.4 | Review at Preliminary Design Review (PDR) |
| 4.1.1.4.5 | Review at CDR |
| 4.1.1.4.6 | Review at FCA/PCA |
| 5 | REVISING INTERFACES |
| 5.1 | Change Request Preparation |
| 5.1.1 | Review/Coordinate Change Request |
| 5.1.2 | Change Approval and Documentation |
| 6 | INTERFACE MANAGEMENT SCHEDULE |
| 7 | NOTES |
| Appendices | |

## E.7 Specialty Engineering Planning

### E.7.1 System Safety Management Planning

System safety is the application of engineering and management principles, criteria, and techniques to optimize safety within constraints of operational effectiveness, time, and cost throughout all program life cycle stages. The NAS Modernization System Safety Management Plan (SSMP) governs system safety efforts conducted in the AMS. The SSMP requires each program to develop, as part of the IPP, an Integrated System Safety Program (ISSP) tailored to the program's safety needs. The ISSP calls for contractors or vendors to develop and maintain a System Safety Program Plan (SSPP) that details the planned safety activities. The SSPP describes safety assessments, tasks, and activities of system safety management and system safety engineering required to support the design process and to identify, evaluate, and eliminate or control hazards throughout the system life cycle.

Government System Safety engineers in the program are responsible for generating the ISSP, and, typically, the System Engineering Council (SEC) approves it as the first step in the system safety program. Contractor System Safety engineers in the program are responsible for generating the SSPP; the Program Manager approves the document internally, and the SEC approves it externally. System safety is an integral element of system engineering applicable to all design stages. Consequently, the stakeholder typically requires the SSPP as early as possible in the program life cycle, usually within 60 to 90 days after contract award. Updates to the SSPP are necessary from stage to stage. Significant program changes may also warrant an update.

A comprehensive, approved SSPP provides value to the overall program. Misunderstandings are avoided regarding the safety definitions, scope of safety analysis, and risk-resolution procedures. The SSPP serves to increase safety awareness within the integrated team, building system safety into the product. The SSPP is tailored guidance for the System Safety Manager or engineer. Finally, the SSPP serves as an important audit trail, justifying the safety work performed and the methodology for safety decisions made. The program shall use the format

and content guidelines for the SSPP documented in the SSMP. The SSMP is available on the Web (http://fast.faa.gov/).

#### E.7.1.1 Inputs to System Safety Management Planning

- Requirements for the System Safety effort detailed in the plan may come from stakeholders' requirements, which flows out of the Requirements Management Process (Section 4.3). Compliance shall be with the FAA NAS Modernization SSMP in the AMS FAST Toolset.
- Available system safety evaluation tools shall be used to determine, validate, and verify requirements in accordance with this manual and the SSMP.
- Inputs typically come from the engineer implementing the SE process. These include, potentially, all design groups and, depending on the program structure, either other specialty engineering groups or SE representatives on design teams. Among others, ensure coordination with Human Factors, Reliability, Maintainability, Quality, and Test and Evaluation.
- Lessons learned from previous programs, incidents, and accidents are to be included.
- The program shall form a program-specific System Safety Working Group (SSWG) that works with the FAA's NAS Modernization SSWG in managing risk.
- Programmatics are made available from the "Manage Program" process.

#### E.7.1.2 System Safety Management Planning Steps

##### E.7.1.2.1 Step 1: Collect Inputs

Coordinate with the program technical groups to obtain input information, including source data, tasks to be delineated in the plan, and other information.

##### E.7.1.2.2 Step 2: Analyze Inputs

Review and organize input data.

##### E.7.1.2.3 Step 3: Define Activities and Effort

Work with the technical experts to document as specifically as possible system safety assessment activities.

##### E.7.1.2.4 Step 4: Lay Out and Baseline Plan

Develop and coordinate the draft plan, incorporating revisions; obtain necessary approvals (lines of business, program management, senior technical experts, stakeholders), and release the baseline version of the plan.

##### E.7.1.2.5 Step 5: Interface With Other Processes

Coordinate and interface with other processes throughout plan deployment.

##### E.7.1.2.6 Step 6: Update/Maintain the Plan

Repeat this process to produce updates to the plan during the course of the program.

#### E.7.1.3 Outputs of System Safety Management Planning

Output is the System Safety Program Plan, which contains details on the intent, procedures, requirements, techniques, and criteria of the system safety program. The program shall use the format and content guidelines for the SSPP documented in the SSMP. The SSMP is available on the Web (http://fast.faa.gov/).

#### E.7.1.4 System Safety Management Planning Metrics

The metric for measuring the product of this process is completion of the plan in accordance with the SSMP. Additionally, the cost to produce and update the plan may be measured.

#### E.7.1.5 System Safety Management Planning Tools

Refer to the NAS Modernization SSMP (http://fast.faa.gov/).

### E.7.2 Human Factors Engineering Planning – see AMS FAST

### E.7.3 Quality Engineering Planning – Reserved

### E.7.4 Reliability, Maintainability and Availability Planning – Reserved

### E.7.5 Electromagnetic Environmental Effects ($E^3$) Planning – Reserved

### E.7.6 Hazardous Materials Management/Environmental Engineering Planning – Reserved

## E.8 Integrity of Analyses Planning

### E.8.1 Analysis Management Planning

The Analysis Management planning section of the SEMP is compiled following JRC-1 approval. It supports the objective of that process: "to create high likelihood that the program's analyses are credible, useful, and sufficient." Analysis Management planning defines the analyses to be performed throughout the program and the operational criteria for the analytic tools to be used, as well as the users and the requirements for verifying that the results are correct and sufficient. As a part of the IPP, this section is reviewed with any other plans at the JRC-2B. The outline (Table 8-5) depicts the recommended contents of the Integrity of Analysis planning section.

**Table 8-5 Table of Contents Integrity of Analysis Planning Section of SEMP**

| Functional Analysis Planning Section Example Outline | | |
|---|---|---|
| 1 | SCOPE | |
| 1.1 | Document Overview | |
| 1.2 | Process Overview | This section contains a diagram showing the interrelationship between the various process elements, including tools, if any. |
| 2 | APPLICABLE DOCUMENTS | |
| 3 | TASKS | The tasks described are tied to the specific organizational and program requirements in accordance with Section 4.9. |
| 4 | PRODUCTS | This section describes the various … |

| Functional Analysis Planning Section Example Outline | | |
|---|---|---|
| 5 | RESPONSIBILITIES | This section details responsibilities of the various organizational entities to accomplish the tasks of Section … |
| 6 | SCHEDULE | The schedule shown in this section is to be tied to the milestones of the IPP. |
| 7 | AUTOMATED REQUIREMENTS TOOL | This section describes the planned use of the requirements management tool, if any. |
| 8 | NOTES | |
| | APPENDICES | |

#### E.8.1.1 Inputs to Analysis Management Planning

To prepare Analysis Management planning, the program team members with a need to perform or to have performed one or more analyses provide inputs. Often in this phase of planning a program, there is an iteration in which initial requests to have each analysis authorized and funded are seen as too extensive and costly for the program. Occasionally, program management (PM) determines that other analyses be performed or that analyses may replace tests or improve confidence; however, history shows that usually more analyses are initially requested than are approved. Negotiations then take place between the proponents of the analyses and the PM until a balanced set of analyses are defined. These negotiations may involve such compromises as reducing the scope of simulations and analyses —and possibly relaxing the precision, which the analyst may wish—to a level that management believes is adequate. Ultimately, each analysis earns its way into the integrated program plan by improving the management-balanced program metrics of cost, performance, and time/schedule. For a more in-depth treatment, see " Integrity of Analyses" (Section 4.9).

The kinds of input data that analysts provide include:

- Title or brief description of the analysis
- Description of programmatic benefit to be gained from the successful performance of the analysis; (i.e., the role the analysis plays in the program)
- Relative place in the project schedule:
  - What tasks may be precursors
  - Which tasks are successors and directly depend upon the analysis (i.e., the interfaces of the analysis to the program)
- The inputs required typically include:
  - System requirements
  - Available technology unique to the analysis (both as used in the system being analyzed and as used to perform or support a part of the analysis)
  - Data sets, as possibly updated by precursor task(s)—a program generally maintains a configuration-controlled set of data (environmental factors (atmospheric models, extent of corrosion conditions, etc., some of which may mature or change through

the course of a program)); trade study parameters (range penalty per pound of weight added, at current design state, etc.); and material properties, etc., to be used in analyses

- The inputs from the planning of successor tasks, which essentially define:
  - The reasons for the analysis to be done
  - System/subsystem/component description, as it is involved in analysis
  - The precision, scope, timing, and quality of results that they may get from the analysis; the nature of the deliverable product of the analysis to each using successor task will be defined

- Analytical tool(s) selected and basis/justification of selection (is it from an approved list of tools available or did the analyst create it?)
- Process and plan for ensuring competence of the analyst (credentials, training, certification, testing, etc.)
- Process for ensuring the integrity of the results (analyst's say-so, cross-check by independent analysis, detailed review by expert, or test validation within specified accuracy, etc.)
- Subtasks to be performed to begin, perform, and validate the analysis
- Estimate of duration and resources required; resources may include labor hours, charged computer runtime, lab support charges, and similar programmatic cost and schedule burdens

#### E.8.1.2 Analysis Management Planning Steps

##### E.8.1.2.1 Step 1: Collect Inputs

Coordinate with program technical groups on analysis needed.

##### E.8.1.2.2 Step 2: Analyze Inputs

Review and organize the data; check for conflicts in precursor/successor relationships among different analyses; and prepare management summaries of resource needs (cost, equipment, facilities, and talent).

##### E.8.1.2.3 Step 3: Coordinate With Interfacing Program Functions

Determine details of configuration control of tool and skill inventories, data sets, scheduling, and so that specific and correct references may be made in the Analysis Management planning section.

##### E.8.1.2.4 Step 4: Lay Out and Baseline the Planning Section

Coordinate the draft planning section; support management/analyst/user negotiations; incorporate revisions; obtain necessary approvals; and release the baseline version of the planning section in the SEMP.

### E.8.1.3 Outputs of Analysis Management Planning

The output of this process *is* the Analysis Management planning section of the SEMP, which typically consists of these elements:

- **Introduction**. This section covers scope and purpose. It is recommended that this section include any analysis that involves separate task management and control, or which has stakeholders from the analyst's sub organization, or which is deemed to have a significant influence on the program product. On the other hand, minor analyses that merely fill in details of work within a single sub organization and are small in scope are not intended to be formally controlled by this planning section (although the precepts of the process "Integrity of Analyses" always apply as a best practice).

- **Specific comments on the role of Configuration Management (CM) as it applies to Analysis Management.** It is recommended that approved analytic tools (including special or proprietary procedures, computer programs, networks, and workstations; and physical, computational, and hybrid models) be under CM, as well as rosters of analysts with expertise annotated. It is recommended that data sets especially be under CM, and the AMP requires use of configured data in managed analyses. (Several analyses using conflicting data leads to faulty conclusions that confuses a program.) Within the planning section, it is also recommended that some special notation (like **{CM}**) be appended to any reference of name, tool, or data that is configuration controlled.

  - Abstract of the programmatic approach(es) to ensure the competence of the analysts. This may range from merely listing credentials within each analysis to a rigorous testing and validation program of analysts doing certain work. With the various options chosen by the program, the reference in any one of the analysis coverages will be simplified.

- **Tailoring.** This section provides tailoring of specific documentation requirements, where applicable. Coordination with the procuring authorities is recommended so that agreement is reached on what tailoring needs to be done to minimize any delay in getting the planning approved.

- **Organization.** This subsection discusses the organizational aspects of analysis management and typically, is a product of SE. The analyses may be performed in any sub organization or by contractors; if so, a separate contracting plan will supplement the Analysis Management planning section. When there is more than one stakeholder for an analysis, the analysis coverage shall deal with possibly conflicting needs. Thus, a hierarchical ranking of precision, scope, timing, and quality of the analysis product will be established, and a single set of requirements levied on the analysis. Analysis Management planning development, deployment, and maintenance are the responsibility of SE within the program. The data to be presented (see the "Inputs to Software/Development Planning" (Paragraph 4.2.4.4.3.1)) for each analysis is the responsibility of an analyst assigned to that analysis. This responsibility covers acquisition, interpretation, analysis, and transmittal of the data to the Analysis Management planning section author.

- **Specific Analyses.** This subsection covers each of the various analyses that qualify for inclusion in the Analysis Management planning. The format follows and addresses the items identified in Paragraph 4.2.4.4.3.1. The final subsection for each analysis will be the connectivity (precursor and successor tasks) of the analysis, and the duration and level of effort required.

#### E.8.1.4 Analysis Management Planning Metrics

The metrics for the process of preparing and maintaining the Analysis Management planning section of the SEMP are the completion of the planning, the readiness of the planning section to support management/analyst/stakeholder negotiations, and the costs of the first draft, release, and maintenance of the planning section.

#### E.8.1.5 Analysis Management Planning Tools

Analysis Management planning is typically prepared using a program-standard word processing tool. Interfacing tools may be noted, to include the business-control and scheduling tools, and the CM tools, as well as any program-unique tools identified.

## E.9 Risk Management Planning

Risk is inherent in every program. Stakeholders know this and expect contractors to address risks in program plans. SE addresses three facets of risk: technical, schedule, and cost. Technical risks include all events that may prevent the program from satisfying contractual requirements, including performance, supportability, maintainability, and regulatory requirements. Schedule risks are events that may prevent timely execution of tasks identified in the IPP. Cost risks are events that may cause actual expenditures to exceed estimated costs.

Risk management is a key process within SE. The program and functional managers implement it by ensuring appropriate resources are applied to reduce risk to acceptable levels. Risk management consists of five essential components: identify risks, analyze risks, identify mitigation options, implement risk-reduction plan, and monitor risks.

The risk management planning section describes the approach, methods, procedures, and criteria for risk management and its integration into the program decision process. It is continually updated throughout the program life with the SEMP.

### E.9.1 Inputs to Risk Management Planning

Inputs include program goals, constraints, IPP/IPS, Rough Order Magnitude/Basis of Estimate.

The risk management process is tailored according to the complexity and criticality of each specific project. The program manager weighs mission goals with the potential benefits and costs and in determining the acceptable level of risk for a program. Stakeholders and regulatory directives may also affect determination of acceptable risk levels.

### E.9.2 Risk Management Planning Steps

Risk Management planning guides the program and functional managers in ensuring that adequate risk management is applied at the key decision points of a program.

#### E.9.2.1 Step 1: Establish Risk Review Team

The team should include at least the project task leaders. It is recommended that all affected specialty support groups be identified and consulted throughout the risk management process. In addition, it is recommended that independent non-advocate experts and stakeholders, if appropriate, be identified for participation during formal risk reviews.

#### E.9.2.2 Step 2: Define Risk Management Process

It is recommended that the Risk Management process, or a specially tailored version that is followed by the program, be documented, as well as justification for modification of the process provided. It is further recommended that the process contain the key steps of identifying risk, assessing risk, and mitigating risk, as well as the procedure for implementing contingency plans and risk monitoring. It is also recommended that appropriate tools to implement each step be identified if available.

#### E.9.2.3 Step 3: Define Risk Assessment Criteria

The risk categories (technical, schedule, and cost) and risk levels defined in the Risk Management process may not be appropriate for every program. Technical risks may be subdivided into such categories as Performance, Supportability, and Software, to emphasize key requirements based on program goals. Acceptable schedule or cost risks may also require adjustment based on program goals or constraints. It is recommended that programmatic risks be added if appropriate; justifications for process modification documented; and criteria for closing a risk item defined.

#### E.9.2.4 Step 4: Identify Key Decision Points

Risks reside in any technology development program. Risk management is an essential tool used by program managers to assess the adequacy of the integrated program plan in achieving program goals. At each program review, the decision to proceed with a program shall be based on recognition of identifiable risks and adequacy of contingency plans. It is recommended that risks be identified and assessed and mitigation options identified before each review.

#### E.9.2.5 Step 5: Define Risk Documentation Procedure

It is recommended that all risks identified, assessed, and mitigated be included in a program's documentation. The risk management planning section includes a risk identification worksheet and instruction for submitting risks. It also provides means of documenting steps taken in the risk management process for each risk until closure of the risk.

#### E.9.2.6 Step 6: Define Monitoring Procedure

When a risk is identified, immediate action may be taken to reduce or eliminate the risk. This would result in a change to the SEMP and possible closure of the risk. Alternatively, action may be deferred until a specific predetermined trigger event occurs. It is recommended that the procedure and forms for identifying the trigger events and resulting contingency action be documented. It is also recommended that the forum for reviewing risks and status of trigger events be identified.

#### E.9.2.7 Step 7: Update this Section as Needed or With Any Updates of the Integrated Program Plan

It is recommended that the program progress be periodically reviewed against the Risk Management Planning section.

### E.9.3 Risk Management Planning Outputs

The following is the general outline (Table 8-6) to be used for the Risk Management Planning section (or as a separate plan if considered necessary).

**Table 8-6 Table of Contents Risk Management Planning Section of the SEMP**

| Risk Management Planning Section Outline | |
|---|---|
| 1 | SCOPE |
| 1.1 | Document Overview |
| 1.2 | System Overview |
| 2 | RISK REVIEW TEAM |
| 3 | RISK MANAGEMENT PROCESS |
| 3.1 | Process |
| 3.2 | Risk Assessment Criteria and Mitigation Requirements |
| 3.3 | Key Decision Points |
| 3.4 | Documentation Requirements |
| 4 | RISK MONITORING PROCEDURE |
| 5 | RISK MANAGEMENT SCHEDULE |
| 6 | NOTES AND REFERENCES |
| 7 | APPENDICES |
| 7.1 | Documentation Forms |
| 7.2 | Risk Management Tools |

### E.9.4 Risk Management Planning Metrics

Completion (or revision as needed) of the Risk Management planning section before each AMS phase exit review and approval of this section at the review are the primary metrics of success.

### E.9.5 Risk Management Planning Tools

Risk Management Planning is typically prepared using a word processing tool. Refer to the appropriate sections of this manual to ensure that the activities described in the Risk Management Planning section are consistent with the SE planning process. This comparison ensures that risk management is injected into the progressive and iterative SE process steps for this program.

## E.10 Configuration Management Planning

Configuration Management planning documents the formal management system of CM to ensure that the integrity and continuity of the design, engineering, and cost tradeoff decisions made between technical performance, producibility, operability, testability, and supportability are recorded, communicated, and controlled by program and functional managers. CM planning provides the means for the:

- Configuration Identification process that identifies the functional and physical characteristics of selected system components, designated as configuration items (CI), during the system's acquisition life cycle
- Configuration Control process that controls the changes to CIs during the system's acquisition life cycle

- Configuration Status Accounting process that records/reports change processing and implementation status
- Configuration Audits process that supplies current descriptions of developing hardware configuration items (HWCI), computer software configuration items (CSCI), and the system itself

The Configuration Management Organization typically owns this planning section. The planning section may be initiated by inputs from the SE process as early as the Investment Analysis, phase one, but formally starts at Investment Analysis, phase two, and continues throughout the program life cycle as the system develops and is modified.

### E.10.1 Inputs to Configuration Management Planning

Following are the two categories of CM planning:

- **Concepts (initial, baseline).** This data identifies the functional and physical characteristics of selected system components and CIs to be controlled and managed.
- **Integrated Program Plan Requirements.** This data identifies contractual and non-contractual constraints, such as program deliverables, cost, and schedule.

### E.10.2 Configuration Management Planning Steps

#### E.10.2.1 Step 1: Collect Input Data

The beginning task is to collect all input data.

#### E.10.2.2 Step 2: Define Configuration Items

The planner determines what is to be controlled and managed by identifying the CIs from the initial and/or baseline concept.

#### E.10.2.3 Step 3: Identify Means for Configuration Change Management

The planner needs to determine how to control and manage each of the identified CIs.

#### E.10.2.4 Step 4: Identify Means for Configuration Status Accounting

This step determines when and how to document the change processing and implementation status and encompasses establishing the frequency and format of the record and report documents.

#### E.10.2.5 Step 5: Identify Means for Configuration Verification and Audit

Identify methods to supply current descriptions of the CIs and means to trace all changes back to the baseline configuration.

### E.10.3. Outputs of Configuration Management Planning

The output shall be the Configuration Management Planning section that outlines all the tasks with corresponding completion dates and personnel responsible for task completion.

### E.10.4 Configuration Management Planning Metrics

The metric for measuring the product of the CM Planning process is completion of the planning section within cost and schedule.

### E.10.5 Configuration Management Planning Tools

The CM Planning section is typically prepared using a word processing and drawing tools.

## E.11 Validation and Verification Planning

The Master Verification Plan (MVP) contains both validation and verification planning. Validation is the process of proving that the right system is being built (i.e., that the system requirements are unambiguous, correct, complete, consistent, traceable to needs, operationally and technically feasible, and verifiable). The validation planning process is conducted to demonstrate that the requirements for a system are clearly understood and that it is possible to satisfy them through design work using available state-of-the-art technology, funding, and schedule. Verification is the process (tasks, actions and activities) of confirming that evolving system solutions comply with functional, performance, and design requirements that spell out stakeholder (internal and external) expectations of capabilities, as well as performance and characteristics of the developed system. Product verification may occur during any phase of a product development cycle, but is more likely to occur after the product Preliminary Design Review (PDR). Verification is the process that ensures that system requirements have been met by the design solution and that the system is ready for use in its operational environment. This means that a verified system may demonstrate that it complies with mission need and meets functional, performance, allocated, derived, and interface requirements, as well as design and allocated constraints that achieve customer needs.

The MVP objective is to define all verification activities that demonstrate the system's capability to meet the specification requirements.

### E.11.1 Inputs to Master Verification Plan

The inputs required for preparing the master verification plan are:

- Existing requirements and specifications documents
- Risk-mitigation plan
- Existing Functional analyses, including CONOPS
- MNS, fRD, and program SOW
- NAS-Level SEMP
- Program-specific schedule constraints and milestones provided by the SEMP and SEMS
- IPP, including the T&E plans and schedules, safety, and quality sections; and the IPS
- Existing product performance/objectives and physical specifications supplied through system/design engineering, including the results of trade studies, baseline product modeling, CM planning and changes, system specifications (SSs), system/segment design document (SSDD), interface control planning and documents (IC, IRDs/ICDs) and technical performance measures (TPMs)
- External standards and conditions required by government regulatory agencies

#### E.11.1.1 Master Verification Plan Steps

The MVP for the component through the system-level is normally written by SE in conjunction with T&E and includes coordinating with program multidisciplinary project teams. The following major steps are required for developing of the master verification plan(s).

##### E.11.1.1.1 Step 1: Collect and Review Inputs

The inputs provided by each responsible organization are required to be collected and reviewed for acceptability and completeness by the MVP coordinator.

##### E.11.1.1.2 Step 2: Develop Master Verification Plan

Using the inputs collected and reviewed in Step 1, the MVP coordinator prepares the master verification plan(s) using the format described in Figure 4.12-12.

##### E.11.1.1.3 Step 3: Review Plan(s)

The master verification plan(s) are reviewed both before and during program critical milestones (normally starting at the PDR or equivalent). The master verification plan(s) are baselined upon initial program approval.

##### E.11.1.1.4 Step 4: Maintain Plan(s)

The MVP plan coordinator maintain continuous cognizance of program progress throughout the life of the program. Changes to the program are reflected in the master verification plan(s).

##### E.11.1.1.5 Step 5: Distribute Plan(s)

The MVP plan coordinator provides the master verification plan(s) to all stakeholders, who manage by the master verification plan(s). These stakeholders include the program manager, stakeholders, project teams and leads, system engineering, test and evaluation, quality assurance, and safety, as a minimum.

#### E.11.1.2 Outputs of Master Verification Plan

The output of the MVP planning is the MVP and includes planning that supports development of the following products.

##### E.11.1.2.1 Master Verification Plan

The MVP describes the overall verification program. It provides the content and depth of detail necessary for full visibility of all verification activities. Each major verification activity is defined and described in detail. The plan provides a general schedule and sequence of events for major verification activities. It also describes test software (including code and documentation), Ground Support Equipment (GSE), and facilities necessary to support verification activities. The systems engineer and verification engineer develop the plan with design and test organizations, with all having a thorough understanding of the verification program concept, program requirements at all levels, and the methods identified in the VRTM for verification.

##### E.11.1.2.2 Verification Requirements Traceability Matrix

The VRTM is that portion of a requirements document that defines how each requirement is to be verified, the plan that describes the verification activity, and the results (including traceability

to the test of verification report). The VRTM is based on the Validation Table documented in the Validation Report. The design, test, SE, and verification team members jointly develop the VRTM. The VRTM establishes the basis for the verification program.

#### E.11.1.2.3 Requirements Verification Compliance Document

The RVCD provides the evidence of compliance for each requirement at all levels and to each VRTM requirement. The flow down from the requirements documents to the VRTM completes the full requirements traceability. Compliance with all requirements ensures that the system-level requirements have been met.

The RVCD defines for each requirement the methods of verification and corresponding compliance information. The results of the verification activity, including evidence of completion, are recorded and documented in the RVCD. It is recommended that the RVCD contain information regarding the results of each verification activity and a description and disposition of conformance, nonconformance, conclusions, and recommendations. The compliance information provides either the actual data, or a reference to the location of the actual data, that shows compliance with the requirement. The document also includes a section that details any noncompliances; it is recommended that this section also specify appropriate reverification procedures. The RVCD is an input into the Requirements Management process (Section 4.3). Decisions regarding what to do with noncompliant requirements are made in Requirements Management.

### E.11.1.3 Master Verification Plan Metrics

Three fundamental metrics exist to help measure and improve the verification plan:

- Timeliness of developing and reviewing the verification plan
- Quality of developing the verification plan
- Cycle Time to complete development and distribution of the verification plan regarding collecting and reviewing the inputs for verification plan development

### E.11.1.4 Master Verification Plan Tools

The MVP shall be completed in accordance with the guidelines documented and tools described in this section and Validation and Verification (Section 4.12).

**E.12 Integrated Planning Lifecycle– Reserved**

**E.12.1 Real Property Management Planning – Reserved**

**E.12.2 Deployment and Transition Planning – Reserved**

**E.12.3 Integrated Logistics Support Planning – Reserved**

**E.12.3.1 Maintenance Planning – Reserved**

**E.12.3.2 Maintenance Support Facility – Reserved**

**E.12.3.3 Direct-Work Maintenance Staffing – Reserved**

**E.12.3.4 Supply Support – Reserved**

**E.12.3.5 Support Equipment – Reserved**

**E.12.3.6 Training, Training Support, and Personnel Skills – Reserved**

**E.12.3.7 Technical Data – Reserved**

**E.12.3.8 Packaging, Handling, Storage, and Transportation (P,H,S&T) – Reserved**

**E.12.3.9 Computer Resources Support – Reserved**

**E.12.4 Sustainment/Technology Evolution – Reserved**

**E.12.5 Disposal – Reserved**

**E.13 Maintain System Engineering – Reserved**