Department of Commerce ● National Oceanic & Atmospheric Administration ● National Weather Service

NATIONAL WEATHER SERVICE INSTRUCTION 80-303

January 17 , 2020

Office of Planning and Programming for Service Delivery

Systems Engineering

SYSTEMS ENGINEERING FOR NEW SYSTEM DEVELOPMENT

NOTICE: This publication is available at: http://www.nws.noaa.gov/directives/.

OPR: W/OPPSD/ES (Benjie Spencer) **Certified by:** W/OPPSD (Kevin C. Cooley)

Type of Issuance: Routine

SUMMARY OF REVISIONS: This directive supersedes NWS Program Directive 80-303, dated April 21, 2009. Changes were made to reflect the NWS Headquarters reorganization effective April 1, 2015. Other changes were made to (1) update the entire document to align with the NWS System Engineering Framework described in 80-301 (2) update Section 2 to present a refined definition of Systems Engineering relevant to the current NWS environment; (3) add new Section 3 to details for the Needs and Definition stage; (4) add new Section 4 to provide details for Design and Development Stage.

Signed 01/03/2020

Kevin C. Cooley Date

Office of Planning and Programming for Service Delivery

Table of Contents

Table of Figures

1 Introduction

This instruction provides guidance on National Weather Service (NWS) Systems Engineering (SE) for new system development. The intent is to outline an approach and a set of activities that provide a consistent method for performing SE across all new NWS development projects and to provide practical guidance for implementing that approach. The approach and activities described in this instruction can be adapted to meet the specific requirements or constraints of virtually any new development project. This document expands on the material presented in the NWS Instruction 80-301: *Systems Engineering Process and Life Cycle*, which identifies the five stages of the NWS SE approach. This instruction further describes the inputs, activities, outputs, deliverables, and reviews associated with each stage with focus placed on the first two stages: the Need and Definition Stage, and the Design and Development Stage.

2 Overview of Engineering Life Cycle Approach

This instruction is specific for new system development and follows the same stages documented in NWS Instruction 80-301: *Systems Engineering Process and Life Cycle*. Each of the five stages in the life cycle development takes inputs that are usually the outputs of a previous stage, except for the first stage, and perform activities using those inputs, which then generates particular outputs consisting of various artifacts and/or deliverables. This instruction provides detailed information on how the activities are performed and the outputs generated for the Needs and Definition Stage and Design and Development Stage. However, the instruction is not meant to be prescriptive and rigid, instead, project teams can further tailor the information in this instruction to meet the project specific needs, requirements, constraints, environment, schedule, budget and situation. As an example, activities of the first three stages of life cycle ("Need & Definition Stage", "Design and Development Stage", and "Operations Validation Stage") can be simplified, grouped and iterated to support an Iterative and Incremental Development approach for a large and loose coupled system development effort.

The *NWS Instruction 80-304 Systems Engineering for Software Development* provides additional details in the Design and Development Stage for software development. The Operations Validation Stage details are covered in *NWS Instruction 80-305 Test and Evaluation, 80-306 System Acceptance Test (SAT) Process and NWS Instruction 80-307 Operational Test and Evaluation (OT&E) Process*.

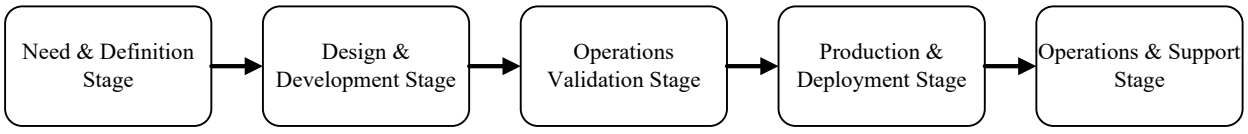

Figure 1: NWS Life Cycle Model

3 Need and Definition Stage

The SE process starts upon recognition of a need for a new or modified System of Interest (SOI). Figure 2 below illustrates the inputs, activities and outputs of this stage.

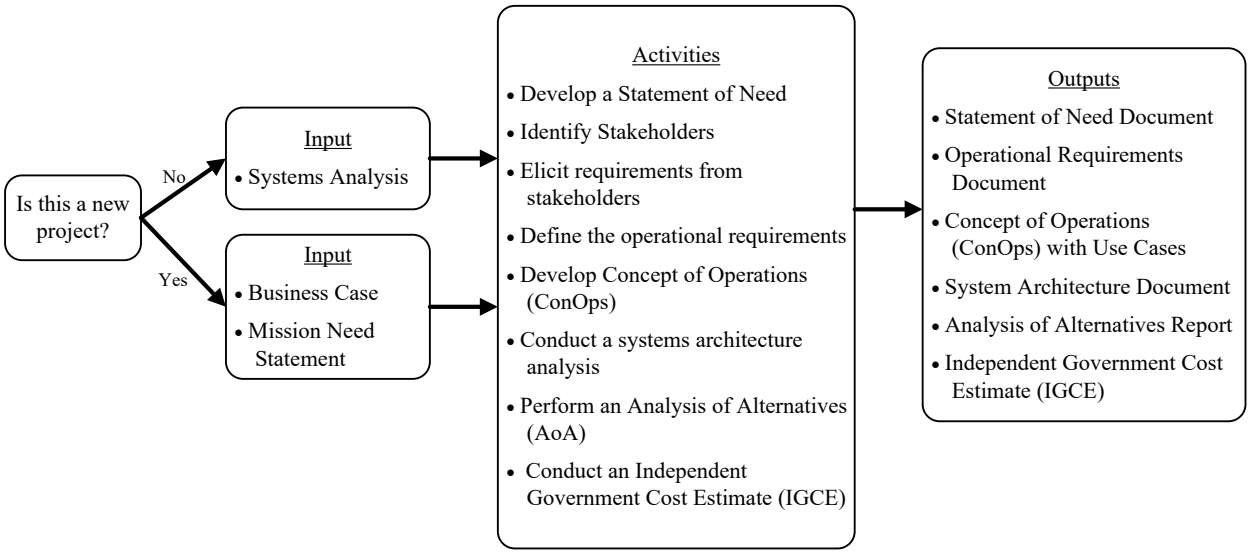

Figure 2: Need and Definition Stage

One of the first questions to address during this stage is: "Is this a new project and require a new system of interest or is the project for the development of upgrades to an existing system?" If the answer is "No", then the required input for this stage is a systems analysis that contrasts the current technical capabilities versus current and future needs. If the answer is "Yes", then the required inputs for this stage are a business case and/or a mission need statement.

Once the proper inputs have been identified, the activities that are performed during this stage are:

- Developing a Statement of Need (SON)
- Identifying stakeholders
- Eliciting requirements from stakeholders
- Defining the operational requirements
- Developing a Concept of Operations (ConOps)
- Conducting a systems architecture analysis
- Performing an Analysis of Alternatives (AoA)
- Performing an Independent Government Cost Estimates (IGCE).

3.1 Develop a Statement of Need

The purpose of developing a Statement of Need (SON) is to document the reasons why a new system needs to be developed. It summarizes the expected operational mission of the system as well as serves as a vehicle to gain sponsor and key stakeholder agreement on the purpose and scope of the system development. The SON also describes the gap in current operational capabilities and summarizes the impact of not addressing that shortfall. The SON must assess the criticality and timeframe of the need, and roughly estimate the resources required.

Developing the SON clearly identifies the need or gap that the new system will resolve. In addition, developing the SON serves to identify deficiencies in current and projected capabilities. A key step in developing the SON is defining the justification for the new system in a clear and concise manner. Finally, developing rough order of magnitude estimates for cost and schedule is needed to inform decision makers of the general scope of the anticipated development.

3.1.1 Elements of SON

Title – Create a concise title which identifies the need or opportunity. The title should be stated in terms understood by persons with limited knowledge in the subject area; in other words, use acronyms sparingly.

Submitting Authority - Select the name for the Submitting Authority (SA) which corresponds to originator's Office, Region, or Financial Management Center (FMC). If the originator is an external customer, their NWS point of contact is the one to fill out the form and the SA corresponding to the point of contact shall apply.

Description - A description of the mission need is to be provided in this section. The need should relate to a deficiency or potential efficiency gain, and should be stated with sufficient details so that it adequately answers the question "What does the NWS need?" or "What opportunity should the NWS consider?"

Justification - This section is used to convey the justification or origin for the need and how the need is linked to NWS plans, mission, or requirements. Discuss the strategic risk to the overall mission of NWS and NOAA of not filling the capability gap. Describe the impact (to safety, health, environment, security, capacity, operations, maintenance, cost, productivity, efficiency, or other factors) on the program's ability to perform its mission if the capability shortfall is not resolved, including the timeframe when this impact would occur.

Linkages – Identify specific connection(s) to NOAA/NWS strategic plans, relevant information in the Planning, Programming, Budgeting, and Execution System (PPBES), and other valid requirements. Describe the priority of fulfilling the mission need relative to other NWS programs, projects, and at the site, installation or laboratory.

Existing capabilities, capacities, and limitations related to the need - Include existing capabilities and capacities that are related to the need. Specify the limitations of these existing capabilities/capacities. Identify constraints or limitations that need to be considered to achieve the mission need.

Benefits and Performance - This section is used to convey impacts of the need. Describe the benefits accrued by the needed capability or technological opportunity. Benefits may accrue from more efficient operations, improved responsiveness to customers, lower operational costs, or other savings.

Supporting Information - List any additional information which supports the justification for the need. Specify any work already done, testimonials, potential alternatives, resources, and analysis.

Resources and Schedule Forecast - Provide a rough estimate of the resources that will likely be committed to this mission need in competition with all others, within the constraint of realistic projections of future budget authority. Develop a rough order of magnitude estimate of the project cost range to acquire/develop various potential alternatives, which address the stated mission need. Also, identify the estimated dates (fiscal year or quarter) for meeting subsequent program milestone dates.

3.1.2 SON Correspondence to other Activities

The SON develops a formal statement of project need and objectives. The SON facilitates approving the project and starting the process of allocating resources. The SON initiates the next activity in this stage to identify stakeholders and elicit their requirements.

3.2 Identify Stakeholders and Elicit Requirements from Stakeholders

Identification of stakeholders and eliciting their requirements are critical to defining operational requirements. The stakeholders are any entities (individuals, systems, organizations) with an interest in, or that will be affected by, the new system. The stakeholder requirements are the particular needs of the various stakeholders related to development, implementation, operations, and/or maintenance of the new system.

Identification of stakeholders for the system is done by identifying all the entities that are affected by, have influence or power over, or have an interest in the system. Depending on the complexity and scope of a project, there may be a few or a large number of stakeholders. Typically, the stakeholders are the system users, operators, maintainers, organizational decision makers, data users and data providers. It is important to understand that not all stakeholders have the same influence or effect on a system, nor are they affected in the same manner. Identification of stakeholders requires a logical method which identifies those who are directly impacted by the system and those who may be indirectly affected.

Requirements elicitation is the process of discovering, reviewing, documenting, and understanding the stakeholders' needs and constraints for the system. The elicitation process with stakeholders is critical, because they frequently identify operational and other requirements that may not be identified without this variety of perspective and experience. The process uses more than one method of elicitation of stakeholder requirements. The following are examples of various elicitation methods: interviews, focus groups, survey/questionnaire, brainstorming session, workshop and conference. When eliciting requirements, it is important to record the requirements in a manner that is design and/or implementation independent. This means

capturing requirements based purely on what operational functionality is needed without attaching it to any specific design criteria.

Identifying stakeholders and eliciting their requirements drive the project by forming the basis of the system capabilities, functionalities, services, quality standards, system technical constraints, cost and schedule constraints, and training and maintenance needs. Stakeholder requirements are used to define the operational requirements in the next activity and to identify assumptions, constraints, and impacts when developing the ConOps.

3.3 Define Operational Requirements

Operational Requirements define how users and operators interact with the system. In addition, the requirements establish how well and under what conditions the system must perform. The stakeholder requirements collected are reviewed, analyzed, and transformed into operational requirements (with certain exceptions, such as budget and schedule requirements). The operational requirements will be transformed to address the following focus areas of operations:

- Operational distribution or deployment: Where will the system be used?
- Mission profile or scenario: How will the system accomplish its mission objective?
- Performance and related parameters: What are the critical system parameters to accomplish the mission?
- Utilization environments: How are the various system components to be used?
- Effectiveness requirements: How effective or efficient must the system be in performing its mission?
- Operational life cycle: How long will the system be in use by the user and are there specific maintenance or support considerations?
- Environment: In what environments is the system expected to operate effectively?

These requirements are used as a basis for the ConOps and the system requirements. It is necessary to transform these operational requirements, which are stated from an operational perspective as required outcomes, into a set of system requirements that are stated in terms of engineering characteristics.

To support their transformation into system requirements, it is imperative that the operational requirements are recorded in a qualitative manner. There are qualitative factors that should be taken into consideration when capturing and documenting requirements during the elicitation process and afterwards. Examples of qualitative factors to consider are as follows:

- Correct
- Unambiguous
- Verifiable
- Complete
- Traceable

- Implementation Independent
- Concise
- Consistent
- Singular, generally one sentence
- Achievable

Some examples of requirements without consideration of quality factors and those with quality factors consideration:

Ex 1: The car's maximum speed should be fast.

The car's maximum speed shall be 100 miles per hour.

The first sentence is ambiguous because the term *fast* cannot be validated nor is it verifiable. Whereas, one can verify is the car's max speed is indeed 100 miles per hour.

When eliciting requirements, it is also important to record the requirements in a manner that is design and/or implementation independent. This means capturing requirements based purely on what operational functionality is needed without attaching it to any specific design criteria. The SE team should be cautious about added constraints that might limit the means of fulfilling the requirement(s).

3.4 Develop a Concept of Operations

Developing a Concept of Operations (ConOps) document is an essential, foundational activity that bridges the gap between the users' requirements and the technical vision of how the SOI will be implemented. The ConOps describes the system characteristics for the proposed SOI from the viewpoint of the intended users and communicate the overall quantitative and qualitative system characteristics to the various stakeholders. As such, the ConOps provides an operational view of the existing and proposed systems and their functional characteristics, and includes supporting operational scenarios and a summary of the anticipated impacts associated with the new system.

Using the operational requirements developed in the previous activity as the baseline, the SE Team performs an analysis to describe how the proposed SOI will function to meet user needs. As this functional analysis progresses, the team works with stakeholders to identify operational scenarios that demonstrate proposed system functionality and to understand the various operational impacts that the new system introduces into affected segments of the organization. These results, along with other related information, are documented in the ConOps. The suggested content of the ConOps is summarized below.

3.4.1 Elements of a ConOps

Background and Scope:

This introduction sets the general context of the project within the organization and establishes the scope and boundaries of the SOI project. This section also summarizes relevant assumptions, constraints, and organizational policies relevant to the development of

the SOI. It may also contain a description of the purpose of the ConOps to communicate to a broad audience the nature of the proposed SOI and its anticipated operations.

Current State of Operations:

This section is used to describe the system or situation (either automated or manual) as it currently exists. If there is no current system on which to base changes, this section can describe the situation that motivates the development of the proposed system. Where a current capability does exist, this section describes the current operational capabilities, highlighting the existing high level architecture from technical, functional, and operational perspectives, describing the role and interactions of the current system within the NWS environment. It may also include areas such as major system components, interfaces to other systems, cost of operations, and performance characteristics. Description of the current state of operations may also list the current user classes and other involved personnel as well as the current support structure and environment.

Justification for and Nature of Changes:

This section summarizes the gap between the current system and/or situation and what is required to meet the newly developed operational requirements. It describes current deficiencies and limitations in meeting new or changed mission requirements and summarizes new or modified capabilities that are needed and that will be provided by the SOI in response.

Proposed State of Operations:

This central section describes the proposed system that results from the desired changes previously discussed in the document. It describes the proposed SOI in a high-level manner, indicating the resulting operational capability without specifying design details. The methods of the description to be used and the level of detail in the description depend on the situation, but it should be sufficient as to fully explain how the proposed system is envisioned to operate to fulfill users' needs. This can include a notional architecture using various perspectives such as Architectural, Functional, Technical, and Operational Views. Areas of focus should include, but not be limited to, the operational environment, major system components, external interfaces, capabilities and functions, and performance characteristics. The descriptions should be simple and clear and should make use of graphical representations when possible, given the varied nature of the intended audience, and should be written from the perspective of the users to the extent possible. Aspects to be described also include modes of operation, user classes, organizational structure, and the anticipated support environment.

Operational Scenarios:

Operational scenarios are practical examples of how users will interact with the SOI in order to perform required mission activities. They are descriptions of sequences of events demonstrating how the SOI will operate and interact with users and external interfaces under given sets of circumstances, which should include all operational modes and user classes identified. Operational scenarios can be depicted in various manners such as sequence diagrams or step-by-step tables that describe user interactions, subsequent data flows, and

outcomes. In general, operational scenarios should also summarize the use case objective, actors, supporting systems, and pre-conditions and may include any alternative or exception conditions and resulting sequences.

Operational Impacts:

This section describes the impacts that introduction of the SOI will have primarily on the users and the support and maintenance organization. Such impacts may include additional activities required for particular user classes and any activities that it may eliminate. It may include temporary impacts on other stakeholders such as the development organization. For users and maintainers, this should highlight any additional skills required to interact with the SOI and skills that may no longer be required, such as those specific to supporting systems that may be getting replaced. Other impacts can include changes to interfaces, procedures, data sources, operational modes, input methods, and operational risks. Organizational changes, modifications of responsibilities, and training required should also be addressed.

Operations Evolution:

The section may be included to describe potential future improvements to the proposed system or service that are not in the current scope of development. This may depict a longer term view of the SOI's capabilities evolution, along with the envisioned evolution of the needs and requirements satisfied by the introduction of the SOI.

3.4.2 ConOps Correspondence to Other Activities

The output of this activity is the ConOps document itself. The ConOps is used in conjunction with the operational requirements to guide subsequent activities related to architecture and alternatives analysis, as well as activities in the follow-on Design and Development Stage. The ConOps is also typically used throughout development to communicate the fundamental nature of the SOI, its anticipated operations, and the various impacts it will introduce.

3.5 Conduct System Architecture Analysis

The system architecture analysis activity is the first step towards defining a comprehensive solution to satisfy the operational requirements and fulfill the vision presented in the ConOps. This initial definition of the solution addresses the architectural principles, concepts, properties, and characteristics of the new system. The architecture forms the organization of a system, embodied in its components, their relationships to each other and the environment, and the principles governing its design and evolution. The following are the steps during this activity:

- Build an understanding of the environment/context
- Establish an architecture structure
- Define and document necessary architecture viewpoints.

Building an understanding of the environment/context of the system can be achieved by analyzing relevant market, industry, stakeholder, enterprise, business, operations, mission, and other information that help to understand the perspectives that could guide the definition of the system architecture views and models. An architectural structure includes methods, modeling techniques, tools, need for any enabling systems, products, or services, process requirements

(e.g., measurement approach and methods), evaluation process (e.g., reviews and criteria). Defining and documenting necessary architecture viewpoints can also be useful in conducting a system architecture analysis. Based on the identified stakeholder requirements and ConOps, it's important to identify relevant architecture viewpoints and architecture frameworks that relate back to both. Determine the design properties applicable to system elements in order to satisfy the architectural characteristics.

The artifacts or documents for this activity may include the following:

- Document Scope - Describes the context and the goals of this document in a narrative
- System Architecture Context Diagram - The System Architecture Context Diagram provides the "big picture" view of the system's architecture
- System Architecture Model - Represents the various architecture components that comprise the system and shows their interrelationships
- Overall Architectural Considerations - Defines how operational requirements and ConOps have been addressed by the architecture
- Architecture Component Definitions - Provides narrative describing and explaining each architecture component in the System Architecture, and identifies specific elements that comprise that component in this system.

These artifacts enable relevant stakeholders to have an in-depth understanding of architecture of the SOI and how it will function. The system analysis also forms the basis for the alternatives in the AoA and ultimately the starting point for the system design.

3.6 Conduct Analysis of Alternatives

A key activity that will give decision makers a basis for choosing the best solution to meet the identified mission need is performing an Analysis of Alternatives (AoA). This activity is driven by the SON, the Operational Requirements, and the ConOps to identify, analyze, and compare solution alternatives that can fulfill the requirements. The process starts by identifying the possible solutions for achieving the required capability, proceeds with evaluating the effectiveness of the alternative solutions and estimating their life cycle costs, and concludes with documenting the analysis and recommending the best alternative for the organization to pursue. The core analysis of this activity evaluates and compares the most viable solution alternatives based on criteria that may include, but are not limited to, effectiveness and performance, cost, risk, operational impacts, and alignment with relevant business plans. The comparative analysis should include a summary of the advantages and disadvantages in these areas for each of the considered alternatives.

The following paragraphs present the suggested content for the AoA report with summaries of the supporting activities in each area.

3.6.1 Elements of the AoA

Introduction:

The introduction gives a summary of the background and the mission needs that drive the analysis. The team performing the AoA and preparing the report should assemble and list in this section the assumptions and constraints applicable to the project and to the analysis. These may be related to various scenarios, risks, and the target operational environment. The team may also include management information that outlines the organization of, approach to, and participants in the analysis.

Description of Alternatives:

In this section, a description of each alternative solution being considered is provided to meet the identified need and summarizes the advantages and disadvantages of each alternative. The breadth of this listing may encompass a greater number of identified alternatives than are included for more detailed analysis, in which case, the team should provide a brief explanation of why certain alternatives have been identified but are not included in the full analysis and specifically identify the down-selected alternatives that will be included. Depending upon the nature of the requirement, the state of the current system or situation, and potential constraints, a baseline alternative is frequently included wherein no new system or capability would be developed, i.e. the "status quo" alternative. In general, all of the alternatives, with the possible exception of status quo, must meet two basic criteria: they must address the capability gaps identified in the statement of need, and they must present a viable or realistic technical solution to satisfy the ConOps and operational requirements.

Analysis:

In this section, the team documents the comparative analyses performed on the alternatives and documents the results along with more detail regarding advantages and disadvantages of each. The analyses to be conducted typically consist of the following:

Effectiveness and Performance: An analysis is performed on each alternative based on its anticipated ability to meet operational requirements. This analysis should include both the functional and performance requirements so far identified, and will be qualitative and quantitative respectively.

Cost: For the cost analysis, an rough order of magnitude estimate of a life cycle cost for each viable alternative is made. Accurate, detailed cost estimation is not required at this point in the process, since this will be done during later activities for the selected solution alternative. However, relevant background information, techniques employed, and basic rationale for the rough order of magnitude estimates should be included.

Risk: For risk analysis, the team identifies and analyzes the major risks associated with each alternative, including assessment of the likelihood and consequence, determination of risk severity, and preliminary mitigation strategies, if applicable. Risks to be analyzed may be categorized as: operational, technical, program management, schedule, or cost related. Based on the risk analysis, it can be useful to assign an overall, relative risk level to each alternative, e.g. low, moderate, or high.

Operational Impacts and Business Plan: Based on the nature of the project, the team may include an analysis of the anticipated operational impacts of each alternative and the level to which each alternative aligns with the organization's overall business plan.

Recommended Solution:

Based on the comparative analyses performed, the SE team selects and recommends the strongest solution among the alternatives. This section should provide a summary of the analysis and the rationale for the recommendation. One or more tables can be useful in this section to list the evaluation criteria and show the relative strength of each alternative in each area. The rationales will likely be a mix of qualitative and quantitative factors.

3.6.2 AoA Correspondence to Other Activities

The AoA report documents the steps taken to identify and analyze alternatives and the selection of a recommended solution. The report clearly states the preferred alternative and provides the rationale for the recommendation based on comparative analysis results.

3.7 Perform Independent Government Cost Estimate (IGCE)

Once a recommended solution has been identified to fulfill the required capability, an Independent Government Cost Estimate (IGCE) is performed to estimate the total costs of development and acquisition of that solution to establish a realistic cost for budget purposes. The IGCE is the Government's estimate of the resources and projected cost of the resources that the development organization will incur to develop and procure the system and may include deployment costs if warranted by the acquisition and deployment strategy. These costs include direct costs such as labor, products, equipment, travel, and transportation; indirect costs such as labor overhead, material overhead, and general and administrative (G&A) expenses; and any profit or fee applicable to an outsourced acquisition. In the case where a system will be outsourced through a commercial contract, the IGCE will also be used by the Contracting Officer for technical and management information and as the baseline for evaluating an offeror's contract price/cost.

For standard materials readily available on the commercial market, catalog or market survey prices may suffice for the estimate and may draw on information available in relevant General Services Administration (GSA) schedules. However, the development of new and/or unique capabilities will require more thorough engineering and management analyses to arrive at a reasonably complete and accurate cost estimate for acquisition of the recommended solution.

IGCEs are typically performed using a spreadsheet to itemize cost elements, calculate subtotals for extended costs based on unit costs, and to sum the total of the entire estimate. Costs can be generally divided into the following primary cost elements: labor, burden on labor costs, other direct costs, indirect costs (overhead), G&A, and profit/fee. A brief summary of each of these areas is provided below.

3.7.1 Elements of the IGCE

Labor costs:

Typically, the most significant category in a system development effort, labor costs must be estimated by listing all anticipated labor categories required with an associated estimate of hours and an hourly rate for each. A good rule of thumb is that a full time employee will have 1,880 productive hours in one year. Percentages or multiples of that can be applied as necessary based on the overall level of effort estimated for each labor category. Labor categories and their associated costs can come from a number of sources such as the GSA, industry surveys and averages, and previous similar development/acquisition projects. The Project Manager should be consulted when deciding upon what source of labor rates will be used.

Burden on labor costs:

This includes such cost items as health insurance, Federal Insurance Contributions Act (FICA) costs, and state and federal unemployment taxes. Burden on labor costs does not need to be estimated separately if already accounted for under labor costs by using burdened rates. Care should be taken in determining if burdened or unburdened rates are used to determine if adding burden costs is needed.

Other direct costs:

This includes, but is not limited to, items such as required materials and equipment, travel, and subcontracts that may be necessary. The types of other direct costs to include are heavily dependent on the type of system being acquired.

Indirect Costs (Overhead): Overhead includes any costs not directly identified with a single, final cost objective, individual project, or contract, and can be stated as general business costs such as rent, utilities, general office supplies, telecommunications, and depreciation.

General and Administrative Expenses:

G&A expense captures the costs of organization-wide support functions (accounting, personnel, purchasing, legal) that are not directly chargeable to any single project or contract. Like overhead, G&A expenses are grouped together and recouped by applying a percentage to other cost categories.

Profit or Fee:

For an outsourced development project performed under a commercial contract, an estimate for the developer's profit or fee needs to be included. Profit or fee is the amount of money that a contractor can expect to earn above and beyond the costs incurred to complete the contract. For the IGCE, profit or fee may be calculated and expressed as a percentage of the total estimated cost and this method can vary based on the anticipated acquisition strategy. The Project Manager should be consulted regarding the acquisition strategy and the desired method for estimated profit or fee.

IGCEs focus on the cost associated with initial development and acquisition of a system or capability. Cost estimating can be expanded beyond that level to further estimate anticipated life cycle costs such as operational costs, maintenance and logistics costs, training costs, and telecommunications costs. The SE team tailors this list as necessary for the specific system or

capability and the resulting estimate can be used by NWS management to support budgeting and other planning activities.

3.8 Outputs of the Needs and Definition Stage

The outputs or artifacts produced from the activities in the Need and Definition Stage of the SE life cycle are:

- **Statement of Need (SON)** - documents the fundamental operational need
- **Operational Requirements Document** - documents the requirements elicited from stakeholders and present them as structured requirements
- **Concept of Operations (ConOps) with Use Cases** - presents a vision of the system's anticipated operations from the users' perspective and is created using the operational requirements
- **System Architecture Document** - presents a high-level, technical framework for the system
- **Analysis of Alternatives Report** - identifies and compares alternative means to achieve the system required to fulfill operational requirements and presents a recommendation
- **Independent Government Cost Estimate (IGCE)** – presents an estimate of development and acquisition costs for the system/project.

4 Design and Development Stage

The Design and Development Stage defines, validates, and realizes that a SOI meets stakeholder requirements and can be produced, deployed, operated, and supported. This stage uses the outputs of the Need and Definition Stage as inputs with the primary output of this stage being the System of Interest approved for entering Operational Test & Evaluation. Figure 3 illustrates the inputs, activities and outputs of this stage.

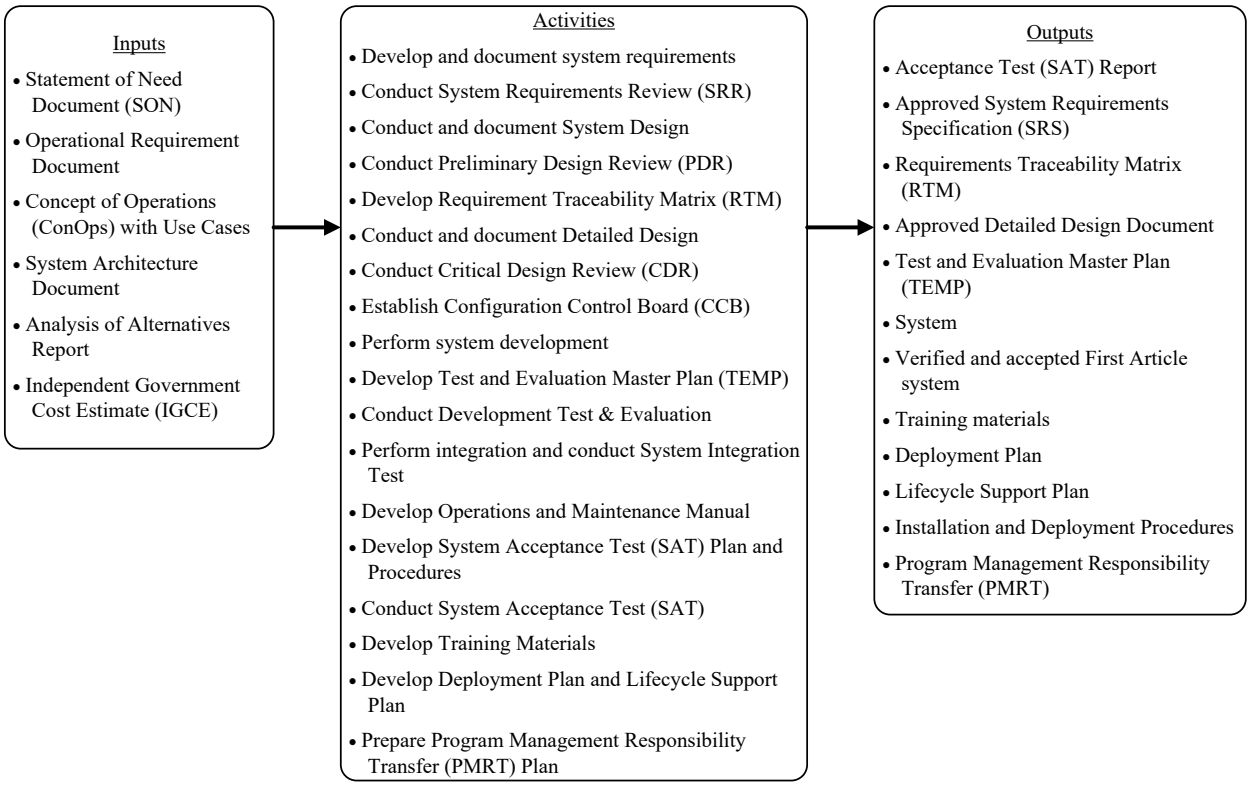

Figure 3: Design and Development Stage

The activities in the Design and Development Stage are:

- Develop and document system requirements
- Conduct System Requirements Review (SRR)
- Conduct and document System Design
- Conduct Preliminary Design Review (PDR)
- Develop Requirements Traceability Matrix (RTM)
- Conduct and document Detailed Design
- Conduct Critical Design Review (CDR)
- Establish Configuration Control Board (CCB)
- Perform system development
- Develop Test and Evaluation Master Plan (TEMP)
- Conduct Development Test and Evaluation (DT&E)
- Perform integration and conduct System Integration Test
- Develop SAT Plan and Procedures

- Develop Operations and Maintenance Manual
- Conduct SATor System Test (ST)
- Develop Training Materials
- Develop Deployment Plan and Life Cycle Support Plan
- Prepare Program Management Responsibility Transfer (PMRT) Plan

It should be noted that in parallel with the SE life cycle processes, separate but related project management activities are being performed under the responsibility of a Government Project Manager. Typically, one of those activities will define an acquisition strategy that will provide definition and guidance as to how the System of Interest will be acquired by the NWS. Pursuant to that acquisition strategy, and as may be defined within any related procurement documentation, all or portions of some of the activities within the Design and Development Stage may be performed by one or more commercial vendors under contract to the Government.

4.1 Develop and Document System Requirements

Developing and documenting system requirements is the activity that transforms stakeholder requirements, ConOps, and operational requirements into verifiable, technical requirements that define what the system will do, but not how it will do it. The system requirements identify what functions are to be performed on what data, to produce what results, at what location, and for whom.

The team uses the outputs from the needs and definition stage and decomposes these artifacts to create more detailed, system level requirements sufficient to guide system design, development, and test. Incorporating design issues and specifications in the system requirements should be avoided. Generally, a requirement specifies an externally visible function or attribute of a system (i.e., “what”). A design describes a particular instance of how that visible function or attribute can be achieved (i.e., “how to”). These system requirements can be assigned categories that can include, but not be limited to, the following.

4.1.1 System Requirement Categories

- **Functional -** Define the system functions or tasks to be performed in operation.
- **Performance** - Define quantitatively the extent, or how well, and under what conditions a function or task is to be performed.
- **Interface** - Define how the system is required to interact or to exchange material, energy, or information with external systems (external interface), or how system elements within the system, including human elements, interact with each other (internal interface). Interface requirements include physical connections (physical interfaces) with external systems or internal system elements supporting interactions or exchanges.
- **Communication** - Define connectivity and access requirements within and between locations and between other groups and applications.

- **Reliability** - Define the dependability rate or baseline of the system.
- **Availability** - Define the user's access and/or ability to utilize the system.
- **Maintainability** - Defines how the system is to be maintained including preventive and predictive maintenance requirements.
- **Security** - Defines how the system is secured, usually requiring specific means because the means are usually regulatory.
- **Regulatory** - Define local, state, federal and international laws that must be followed.

The system requirements are defined in a requirements hierarchy in which the highest-level "parent" requirements are supported by more detailed "child" requirements. A hierarchy allows requirements development to start with high-level requirements and work down to the more detailed requirements. The highest-level requirements should trace to stakeholder requirements, the Concept of Operations and Operational Requirements.

The creation of a standard identification system for all requirements is required in order to facilitate configuration control, requirements traceability, and testing activities. The identification system must provide a unique designator for each requirement. For example, the identification system can classify the requirements by type (e.g., functional, input, or IT security). Within each type classification, the requirements can be assigned a sequential number. Select an identification system that is appropriate for the scope of the project.

All of this information feeds in to a Systems Requirements Specification (SRS). The SRS documents the requirements in a well-organized, approachable fashion so that the stakeholders and system development team can all easily understand and review them. A complete set of SRS elements can be found in section 4.1.2

4.1.2 Elements of the System Requirements Specification (SRS)

Purpose and Scope of this Specification

This section describes the purpose of this specification and its intended audience. It includes a description of what is in scope and what is outside of the scope of these specifications.

System Overview

This section provides a high level description of system including its usage and context.

Assumptions

This section lists any assumptions that affect the requirements.

Constraints

This section describes any items that will constrain the design options.

Dependencies

This section lists dependencies that affect the requirements.

Requirements

This section describes all system requirements in enough detail for designers to design a system satisfying the requirements and testers to verify that the system satisfies requirements. These requirements should be organized in a way that works best for the project. Every input/output into/out of the system should be described, and every function performed by the system in response to an input or in support of an output should be described. For instance, specify what functions are to be performed on what data to produce what results at what location for whom. In addition, each requirement should be numbered (or uniquely identifiable) and prioritized.

4.1.3 System Requirements Correspondence to Other Activities

The first goal of developing and documenting system requirements is to specifically define the required technical capability that will satisfy stakeholder and operational requirements in the manner described in the ConOps. The second goal is to provide a solid technical basis for the system design with specific criteria that can be verified by the test and evaluation process. Consequently, the System Requirements Specification (SRS), will feed directly into both the design and development processes and the test and evaluation activities once it is approved in the next activity of conducting the System Requirements Review.

4.2 Conduct System Requirements Review (SRR)

The SRR is a technical review to determine the acceptability of the defined system requirements. The review will show that all system requirements are defined and are consistent with budgetary, risk, and other system constraints. In addition, this review assesses the system requirements to ensure they are supported by the recommended system solution and available technologies.

4.2.1 Entrance Criteria

Before the SRR can occur, a preliminary SRR agenda, the defined success or exit criteria, and supporting technical artifacts are needed that will be presented at the SRR review. The System Requirements Specification (SRS) is the primary technical artifacts to be presented and reviewed, however additional artifacts and information may also be presented that may include, but not be limited to, the following:

- Operational Requirements Document
- System Architecture Document
- Concept of Operations (ConOps)
- Key and driving requirements and requirements dependencies
- Requirements allocation; preliminary system requirements allocation to the next lower level systems
- Verification and validation overview; preliminary test and evaluation strategy
- Risk analysis.

4.2.2 Exit Criteria

Prior to the SRR, a list of exit criteria is developed for the review to determine its level of success. Such criteria may include, but not be limited to, the following:

- The overall technical concept is reasonable, feasible, complete, and is consistent with system requirements and available resources (cost, schedule, etc.).
- Requirements definition is complete with respect to top-level operational requirements, and interfaces with external entities and between major internal elements have been defined.
- Requirements allocation and flow down of key driving requirements can be defined down to subsystems.
- Operational concepts are consistent with the documented requirements.
- Preliminary approaches have been determined for how requirements will be verified and validated.
- Major risks have been identified and viable mitigation strategies have been defined.

4.2.3 System Requirements Review Correspondence to Other Activities

The primary output of the SRR is the approved SRS, which will drive all subsequent design, development, and testing activities. The design and development activities implement the SRS in the form of the new system, and the test and evaluation process verifies that the new system meets all the technical requirements of the SRS.

4.3 Conduct and Document System Design

The system design is based on the system requirements and provides an overall structure for the system. Specifically, the system design describes the logical system flow, data organization, system inputs and outputs, processing rules, and operational characteristics of the system. The system is decomposed into subsystems and requirements are allocated to the subsystems, and interfaces are specified in detail. The design shows not only the functions that have to be performed, but also the logical sequencing of the functions and performance requirements associated with the functions.

4.3.1 System Design Activities

Decompose System - System decomposition divides the system into a lower level of abstraction consisting of subsystems. At this level of abstraction, each subsystem is structurally and functionally distinct from other subsystems and is separately named and referenced. The number and type of subsystems required to partition a design are dependent on factors such as the complexity of the product, the design method used, and the development environment. The goal of the decomposition is to create a cohesive, loosely coupled, and readily adapted design.

Analyze and Allocate Requirements – Once the requirements are decomposed to sufficient granularity, they are allocated to specific subsystems and components. This allocation is focused on the functional and performance requirements. Through this process, each component is made as independent of the other components as possible based on system functions and performance. This process should account for all system requirements and may lead to definition of additional subsystems to account for requirements that are initially unallocated.

Identify Data Flows - The data flow defines the flow of data through the system and determines a logically consistent interface structure for the system. Each subsystem that defines a function is identified, interfaces between modules are established, and design constraints and limitations are described. In addition, the data flows will describe the sources and destinations, the flow control, and the net transfer of data across the system boundary.

Identify and Define the System Interfaces – Interfaces are the functional and/or physical interaction points between two systems or subsystems. The interfacing components can be internal or external to the system under development. The interface requirements are used to determine the interfaces between internal subsystems and which subsystems will support external interfaces.

Document the System Design – As the system design is being developed, it is captured in a preliminary design document. This document can make use of different system views with each view focusing on a single aspect of the system, facilitating understanding and analysis of the system design. The specific views presented will vary, but they will typically include a physical view that identifies the system components and their relationships; a functional view that describes the system's behavior; an interface view that identifies the interfaces in detail, and an informational view that describes the information that will be managed by the system. Section 4.3.2 provides is a high-level outline that can be used to present the information in the system design document.

4.3.2 Elements of the Software System Design Document

Introduction

This section provides a brief description of the Systems Design Document's purpose and scope.

System Overview

This section describes the system in narrative form using non-technical terms. It should provide a high-level system architecture diagram showing a subsystem breakout of the system. The high-level system architecture or subsystem diagrams should, if applicable, show interfaces to external systems.

Design Constraints

This section describes constraints in the system design and includes any assumptions made by the project team in developing the system design.

System Design

This section describes the system and/or subsystem(s) architecture for the project. Discuss the general architectural decisions that have been approved. Include graphical representations, where appropriate.

Data Flow

This section describes the different levels of data flows of the system such as: summary of top-level, system level (between system(s)/user/ device), between subsystems.

Interface Design

This section describes the design of both internal and external interfaces.

4.3.3 System Design Correspondence to Other Activities

The system design comprises the materials to be reviewed and approved through the Preliminary Design Review (PDR). The system design will also be used as the starting point to create the detailed design before system development starts.

4.4 Conduct Preliminary Design Review (PDR)

The Preliminary Design Review (PDR) ensures the preliminary system design and basic system architecture are complete, and that there is technical confidence the capability need can be satisfied within cost and schedule goals. The PDR also establishes the system's allocated baseline. The allocated baseline describes the functional and interface characteristics for all subsystems (allocated and derived from the higher-level product structure hierarchy) and the verification required demonstrating that specified characteristics can be achieved. The PDR also demonstrates that the preliminary design meets all requirements with acceptable risk and within the cost and schedule constraints and establishes the basis for proceeding with the detailed design. It shows that the correct design options have been selected, interfaces have been identified, and verification methods have been described.

To ensure that the team is ready to conduct the PDR, an entrance criteria checklist should be prepared in advance and reviewed prior to scheduling the review. A sample list of entrance criteria can be found in section 4.4.1.

4.4.1 Entrance Criteria

- Successful completion of the SRR and closure, or accepted plan for closure, of outstanding issues.
- A preliminary PDR agenda and success criteria list
- Technical artifacts for both hardware and software system elements. A sample listing of such products is included below:
    - Updated System Design Document
    - Preliminary subsystem design specifications
    - Updated risk assessment and mitigation
    - Updated cost and schedule data

- Updated logistics documentation, as required
- Applicable standards
- Safety analyses and plans
- Engineering drawing tree
- Preliminary interface control documents
- Verification and validation plan
- Plans to respond to regulatory requirements, as required
- Technical resources utilization estimates and margins.

In order to evaluate how well PDR satisfies the intent of the process, a success criteria checklist should be prepared in advance, reviewed and agreed to, and used to measure the outcome of the PDR. A sample list of success criteria is described in section 4.4.2.

4.4.2 Success Criteria

- The top-level requirements-including mission success criteria, Technical Performance Measures (TPMs), and any sponsor-imposed constraints-are agreed upon, finalized, stated clearly, and consistent with the preliminary design.
- The flow-down of verifiable requirements is complete and if not, an adequate plan exists for timely resolution of open items. Requirements are traceable to mission goals and objectives.
- The preliminary design is expected to meet the requirements at an acceptable level of risk.
- Definition of the technical interfaces is consistent with the overall technical maturity of the SOI and provides an acceptable level of risk.
- Any required new technology has been developed to an adequate state of readiness, or backup options exist and are supported to make them a viable alternative.
- The project risks are understood and have been credibly assessed, and plans, a process, and resources exist to effectively manage them.
- The operational concept is technically sound, includes (where appropriate) human factors, and includes the flow-down of requirements for its execution.

4.4.3 PDR Correspondence to Other Activities

The overall result of a successful PDR is approval of the primary design so that detailed design and development can commence. The approval may be contingent upon the design team implementing corrective actions identified at the PDR and may require a follow-up review of the corrected aspects.

4.5 Develop Requirements Traceability Matrix (RTM)

A Requirements Traceability Matrix (RTM) is a tool to track the development, implementation, and testing of system requirements. Each requirement in the requirements specification document should be uniquely identified in a RTM. The RTM traces each operational and system requirement from its source to all applicable design and development level requirements and traces each of these to design components, subsystems, and test cases for each planned test, including verification methods. Key elements of the RTM can be found in section 4.5.1

4.5.1 Elements of the RTM

Requirement ID - A unique identification number of the requirement and a number assigned in ascending order

Requirement Description - The requirement using "shall"

Rationale – Rationale for the requirement. This field provides guidance to the developer

Parent Requirement – Parents of the requirement

Child Requirement – Children of the requirement

Verification method - (analysis, inspection, demonstration, and/or test)

Verification Result – Indicates if requirement passed or failed verification.

Modification field – Provides when a requirement is changed, eliminated, or replaced, indicate disposition and authority for modification

4.5.2 RTM Correspondence to Other Activities

It is paramount that the team keeps orderly requirements documentation to support subsequent design, development, and testing activities. The RTM enables the team to trace the life cycle of each individual requirement, along with creating an artifact that can be transferred to new members of the SE or relevant stakeholders. The RTM is used in the detailed design activity to ensure that every technical requirement is accounted for in the design. The RTM is then used as a reference during development to further ensure that all requirements are implemented in the system. Finally, the RTM is used to track the verification of the requirements throughout the test and evaluation processes.

4.6 Conduct and Document Detailed Design

Detailed design of the system is the last design activity before implementation begins. It is the "build-to" design of the hardware, software, and selection of commercial-off-the-shelf (COTS) products (if applicable), and defining how the components are developed to meet the system and subsystem requirements. The software design is described in enough detail that the software team can write the individual software modules. For hardware, this activity describes the hardware elements in enough detail to be fabricated or purchased. If it has been determined that COTS equipment should be utilized, the detailed design activity can be where alternative products are evaluated based on pre-determined selection criteria and a selection can be made

accordingly. At the completion of this activity, a Critical Design Review will be held to review and approve the "build-to" design.

4.6.1 Steps for Conducting and Documenting Detailed Design

Develop detailed hardware and software component design specifications – Detailed design specifications are created for each hardware and software component to be developed. The level of detail in the detailed design specifications is greater than that in the high-level design in two important respects:

- The detailed design often includes another layer of architectural design for complex components. All hardware/software units and their interfaces are defined to provide a framework for development of the component.
- The detailed design specifies exactly how the component is implemented so that it meets the requirements. For hardware, schematic drawings and parts lists are defined. For software, this includes identification of algorithms, detailed data structures, and specification of third-party software packages that will be used.

Select COTS products – A vendor selection process will be developed based on the type of COTS needed. There are two fundamental ways that a product can be selected:

- A trade study can be performed that compares the alternative products and selects the best product based on selection criteria that are in turn based on the specification.
- A competitive procurement can be used that allows vendors to propose products that will best meet the specification.

In either case, the functional and performance requirements that are allocated to the COTS product should be used to define the selection criteria and the selection should be driven by COTS specification and/or prototyping the product.

Define the interfaces – Define and document the details of the physical and functional interface characteristics during the detailed design activity. Specifically, the characteristics of the interface, the media involved in the interaction, and the characteristics of the information crossing the interface will be defined. Once an interface has been categorized and its initial contents defined, that interface definition must be documented and approved by the owners of both sides of the interface. This documentation typically takes the form of Interface Control Documents (ICD).

Prototype user interface – If a user interface is to be developed, a user interface prototype should be employed to help the user and developer(s) visualize the interface before significant resources are invested in software development. This is one area in particular where multiple iterations are expected as the developers incrementally create and refine the user interface design based on user feedback.

The outputs or artifacts for this activity are the following, but not limited to:

- List of COTS products, their specifications, and vendor
- Interface Control Document (ICD)
- User interface prototype
- Detailed Design Document

4.6.2 Elements of the Detailed Design Document

- **Purpose** - The purpose of this document is to detail the activities required to perform the Detailed System Design. During this process, the preliminary system design is translated into a technical detailed system design.
- **Detailed System Design Standards** – List the standards that were followed during the detailed design.
- **System Overview**- Provide a general description of the system overview from the system design document.
- **Design Constraints** - Describe any global limitations or constraints that have a significant impact on the design subsystem
- **Design Details** - All components described in the system design document section will require a more detailed discussion. Other lower-level components and subcomponents may need to be described as well. Each subsection of this section will refer to or contain a detailed description of system hardware of software component.

4.6.3 Detailed Design Correspondence to Other Activities

At the completion of this activity, a Critical Design Review will be held to review and approve the "build-to" design. The documents developed during this activity will be used to perform the system development. Some of the documentation will also be used during the Operations Validation Stage to verify requirements.

4.7 Conduct Critical Design Review (CDR)

A Critical Design Review (CDR) is a technical review to ensure that the system can proceed into fabrication, demonstration, and test and can meet stated performance requirements within cost, schedule, and risk. The CDR determines the technical solution's completeness (i.e. that it meets all the requirements in SRS) and consistency with standards. In addition, it raises and resolves any technical and/or project-related issues. The CDR also identifies and mitigates project, technical, security, and/or resource risks affecting continued detailed design and subsequent development, testing, implementation, and O&M activities. Finally, the CDR establishes requirements and system interfaces for enabling system elements such as support equipment, training system, maintenance, and data systems. At this point the system has reached the necessary level of maturity to start fabricating, integrating, and testing pre-production articles with acceptable risk.

To ensure that the design team is ready to conduct the CDR, an entrance criteria checklist should be prepared in advance and reviewed prior to scheduling the review. A sample list of entrance criteria is given below.

4.7.1 CDR Entrance Criteria Sample

- Successful completion of the PDR and closure, or accepted plan for closure, of outstanding issues.
- A preliminary CDR agenda and success criteria
- CDR technical work products lists below for both hardware and software system elements
    - Software design document(s)
    - Hardware Design Document(s)
    - Build-to specifications for each hardware and software configuration item, along with supporting trade-off analyses and data,
    - Fabrication, assembly, integration, and test plans and procedures
    - Operational limits and constraints
    - Acceptance criteria
    - Verification and Validation plan
    - Updated technology development maturity assessment plan
    - Updated risk assessment and mitigation
    - Updated reliability analyses and assessment
    - Update cost and schedule data
    - Updated logistics documentation
    - Subsystem-level and preliminary operations safety analyses
    - Architecture and design drawings
    - Updated SRS and RTM
    - Database design and data dictionaries
    - Component drawings and parts lists
    - Preliminary test plans
    - Quality assurance plan

In order to evaluate how well CDR satisfies the intent of the process, a success criteria checklist should be prepared in advance, reviewed and agreed to, and used to measure the outcome of the CDR. A sample list of success criteria is given below.

4.7.2 Success Criteria

- The Detailed Design is expected to meet the requirements with adequate margins at an acceptable level of risk.
- The Interface Control Documents are appropriately matured to proceed with fabrication, assembly, integration, and test, and plans are in place to manage any open items
- High confidence exists in the product baseline, and adequate documentation exists or will exist in a timely manner to allow proceeding with fabrication, assembly, integration, and test.
- The product verification and product validation requirements and plans are complete.
- The testing approach is comprehensive, and the planning for system assembly, integration, test, and operations is sufficient to progress into the next phase.
- Adequate technical and programmatic margins and resources exist to complete the development with budget, schedule, and risk constraints.
- Risk to mission success are understood and credibly assessed, and plans and resources exist to effectively manage them.

4.7.3 CDR Correspondence to Other Activities

The overall result of a successful CDR is approval of the detailed design so that development can commence. This approval may be contingent upon the design team implementing corrective actions identified at the CDR and may require a follow-up review. The completion of CDR also initiates the start of formal Configuration Management (CM) of the Technical Baseline. Any changes to that baseline can only be accomplished with the approval of the CCB.

4.8 Establish Configuration Control Board (CCB)

In conjunction with approval of CDR, a technical Configuration Control Board (CCB) is established. The CCB is established to govern changes to and versioning of approved documentation from the CDR. The CCB has several functions, which include, but not limited to, the following:

- Serve as the principal decision body for the system change control
- Review and approve or deny requested changes to technical documentation including, but not limited to, the documents approved as a result of the CDR
- Ensure changes concur with validated requirements
- Identify net effect of proposed changes on controlled baselines, documentation, maintenance and logistics support, field maintenance workload impact, costs, schedules, performance, interface with other systems, and when necessary, priority among other proposed changes
- Request and review test strategies for changes and review test results

- Identify program and/or system resources necessary to accomplish all tasks assigned by the CCB or, if required resources are unavailable, request and justify additional resources through designated channels
- Document and report all major decisions and their rationale
- Recommend changes that have a significant effect on the system, budget, or previously approved development schedule(s) to the Program Manager
- Report or elevate unresolved CCB issues to the Program Oversight Board or other relevant high level stakeholders serving in that capacity

The CCB can comprise the following members, but not limited to: chair, secretary, and general membership (non-officer members). The CCB Chair provides direction for team, resolves conflict within the team, assigns task, and enforces the team's decisions on the project. The secretary prepares and distributes minutes, completes and disseminates Change Request Review Packages, maintains change status and provides the CCB with routine changes status reports and other documentation related to CCB activities. The general members represent their office or organization in CCB matters, review and provide feedback to Change Request Review and complete actions assigned by the Chair.

4.8.1 CCB Correspondence to Other Activities

Before system development starts the CCB has to be established. The system development team will find information in the design documentation that will need to be modified. To ensure that these inevitable changes are communicated appropriately and accounted for across all relevant project activities, the CCB makes the decisions about which changes to accept and which to reject, and how these changes will be implemented and documented.

4.9 Perform System Development

This activity in the approach develops [builds or constructs] the hardware and software for the system that matches the requirements and component level detailed design documentation. An implementation team of specialists fabricates the hardware and writes the software. Once developed the hardware and software will be implemented, tested, and incorporated into the baseline. Progress is monitored through a planned series of walkthroughs, inspections, and reviews. These steps below are primarily the responsibility of the development team, who fabricates the hardware and writes the software programs.

4.9.1 Steps for Performing System Development

Plan software/hardware development – The implementation team documents its development process, best practices, and conventions that will be used. The team also documents the development methods, documentation requirements, delivery stages, configuration control procedures, technical tracking and control processes, and the review process.

Establish the development environment involves assembling and installing the hardware, software, communications equipment, databases, and other items required to support the coding/construction effort. Specifically, the tools that are used to develop and test the hardware

and software are selected, procured, and installed, including development tools, source control tools, third-party application libraries, and test simulators. Every tool that is used to support development should be documented specifically enough so that the development environment can be replicated if necessary.

Procure COTS products – COTS products are procured based on the product specifications developed in the detailed design. Use just in time procurement and delivery of the COTS to reduce the possibility of the hardware or software becoming outdated before it can be integrated into the project.

Conduct Development – The software is written and the hardware is built based on the detailed design. Hardware and software releases will be developed, tested, and made available to selected users for feedback. Code inspections and code walkthroughs should also be used to check the software quality.

Develop supporting products – Enabling products, such as training materials, user and maintenance manuals, online help, and installation and conversion software, are developed. It is natural to focus on the hardware and software in the "end product", but attention must be given to developing and accounting for all ancillary products that are needed to support the operational system**.**

Perform unit/device testing – The software and hardware components are thoroughly tested to identify as many defects as possible. The first line of defense is the developer, who should step through and test every line of code, including all exception and error cases. Additionally, a series of test cases are designed that will exercise the hardware/software components; these test cases are documented in a unit verification plan. Identified defects are analyzed and corrected, and testing is repeated until all known defects are either fixed or otherwise resolved.

Establish the Development Baseline - A development baseline is an approved "build" of the product. A build can be a single component or a combination of components. The first development baseline is established after the first build is completed, tested, and approved. Once the first development baseline is established, any changes to the baseline must be managed under the change control procedures for the system.

4.9.2 Perform System Development Correspondence to Other Activities

The hardware and software generated from this activity will be integrated during the Perform Integration and Conduct System Integration Test activity of this life cycle stage. The integrated system will also be formally verified and validated against the requirement and design artifacts during the Conduct SAT activity. The tested system will be used to write the O&M manuals and various training materials required by operations and maintenance personnel.

4.10 Develop Test and Evaluation Master Plan (TEMP)

This activity defines and documents the Test and Evaluation (T&E) activities to be carried out for the SOI from its design and development through its commissioning into operations. This is captured by developing the Test and Evaluation Master Plan (TEMP). In developing the TEMP, the team will need to:

- Establish the overall testing philosophy and strategy
- Develop an overall test management process
- Develop an integrated test program schedule
- Identify resources required for the T&E program
- Provide guidance regarding documentation and reporting of T&E activities

The resulting TEMP serves as the parent document for all T&E segments and their specific test plans. The T&E segments will typically consist of development, integration, system, and operational tests, each with its own specific activities and documentation described separately in this instruction.

Additionally, the TEMP should include provision for all operational and system requirements (independent and dependent) to be covered by one or more of the dependent tests to ensure that the system is fully tested and evaluated at the appropriate levels. The allocation of specific requirements to specific tests will be captured in parallel as part of the RTM. There should also be a provision to address discrepancies uncovered between the system requirements and the T&E criteria for each stage of testing.

4.10.1 Elements of the TEMP

Suggested areas of coverage for the TEMP include, but are not limited to:

- System introduction
- Description of overall testing philosophy, strategy, and methodology
- T&E management processes
- T&E integrated master schedule
- Overview and objectives of each of the T&E activities:
    - Developmental Test and Evaluation (DT&E)
    - Integration Test (IT)
    - System Acceptance Test (SAT)
    - Operational Test and Evaluation (OT&E)
- T&E resource summary
- General test and issue reporting processes
- General test approval processes
- Appendices (any additional information used in supporting test program planning)

For complex projects and programs, the test plans for DT&E, IT, and SAT will likely be separate documents referenced in the TEMP. The OT&E Plan, however, will typically be a separate

document from the TEMP regardless of the size or complexity of the project due to the more independent nature of that testing activity.

For further guidance, definitions, and summaries of the milestones please refer to NWSI 80-305 Test and Evaluation.

4.10.2 TEMP Correspondence to Other Activities

The TEMP is the overall, guiding document for all test and evaluation activities. It provides the framework and ground rules for planning, governing, scheduling, conducting, and documenting each of the subsequent test related activities and will be referenced throughout the T&E program.

4.11 Conduct Development Test & Evaluation (DT&E)

Development Test and Evaluation (DT&E) verifies system development against approved system requirements by focusing on unit and component verification during the development phase to assess development progress and maturity in preparation for integration testing. DT&E are tests conducted within the controlled development environment without introducing integration or operational complexities, are typically more informal than subsequent testing activities, and are performed by the system development team with project management oversight. The level of formality and documentation required should be established in the TEMP, but will typically consist of the DT&E Test Plan, Test Procedures, and Test Report.

The first part of the DT&E activity is to develop the DT&E Test Plan that defines the scope, structure, methodology, and test cases of DT&E.

4.11.1 Elements of the DT&E Plan

Suggested areas of coverage for the DT&E Plan include, but are not limited to:

- Background
- Purpose and objectives
- System under test description
- Test cases
- Use cases
- Assumptions and limitations of the test and system under test
- Applicable policies
- Test management
- Entrance criteria
- Success criteria
- Test schedule

A description of the development test environment should be included in the test plan and may be a part of the system under test description or as introductory material for the test case section. In developing the test cases, the development test team will expand each test case to include specific test setup, test steps or procedures, and the expected test results. The team should also reference system requirements that are covered by each test case to provide traceability between DT&E and the SRS (this should also be reflected in the RTM). These complete test cases will form the basis for conducting DT&E.

4.11.2 Steps for Conducting DT&E

The specific activities comprising DT&E will likely vary from project to project based on various factors such as the size and complexity of the system under development, the composition of the test team, and the level of Government involvement in and/or oversight of development testing. However, as a general guideline, the following activities are usually performed as part of DT&E once components, modules, and subsystems are ready for development testing:

- Validate that entrance criteria have been met
- Validate the test environment and test setup
- Execute the test cases via the test procedures
- Document testing results including discrepancies discovered
- Perform any ad hoc testing thought of while performing documented test cases
- Document ad hoc testing and results
- Analyze discrepancy results and formulate corrective actions
- Perform and document regression testing required as the result of corrective actions

4.11.3 DT&E Correspondence to Other Activities

The principal outputs of DT&E are hardware components and software modules ready for the system integration and test activity. Other outputs will include any documentation for DT&E required by the TEMP and/or DT&E Plan, which would typically be in the form of a DT&E Test Report. An example of a DT&E Test Report outline is provided in Appendix B of reference document NWSI 80-305 *Test and Evaluation*.

4.12 Perform Integration and Conduct System Integration Testing

Systems Integration consists of integrating system elements and assembling these implemented elements together, in the course of the assembly. The ultimate goal of systems integration is to ensure that structural and/or logical components of the system function properly as a whole integrated system. The systems integration process is an iterative process in which software and hardware components that comprise the system are progressively integrated one subsystem at a time until the entire SOI is fully integrated.

Systems Integration Test (SIT) requires the development of an integration plan. The integration plan defines the order in which the project components are integrated with each other and with other subsystems. Each integration step includes tests that verify the functionality of the integrated assembly, with particular focus on the interfaces. For less complex projects, the integration plan can be informal. For complex projects, there will be more planning to ensure the system is integrated in efficient, useful increments consistent with the master schedule.

Based on the integration plan the SOI is progressively integrated based on the high-level design and the integration plan. The subsystem components are integrated with each other and with other interfacing systems. Integration tests are used to verify that the components and higher-level assemblies work together properly and do not interfere with one another. Integration tests are used to exercise the interfaces and verify that all interfaces are implemented according to the documentation. Proposed changes to the baseline high-level design, including any required changes to the interface documentation, are identified.

The SIT is included in the Design and Development stage of the NWS systems engineering life cycle model. Testing during the SIT is to simulate as close to an operational environment without involving actual operational sites. The SIT is an informal test managed by project management (PM) and conducted by designated system integrators.

After the SIT has been completed there will be a record created that will document:

- Analysis and disposition of any identified anomalies and outliners that occurred during testing
- A list of test that were performed
- A list or statement stating whether any external systems were affected by the testing.

After SIT, there should be a fully integrated SOI ready for SAT.

4.13 Develop Operations and Maintenance Manual

The purpose of an operations and maintenance manual is to formalize and document the policies, plans, procedures, and other support activities that are performed. This manual is a critical tool that supports the management of day-to-day operations by defining the roles, responsibilities, functional capabilities, services provided, major tasks, and other day-to-day activities that are performed. The manual should address routine maintenance, scheduled preventative maintenance and operations procedures.

The starting point for an operations manual is often the concept of operations document. In general, the ConOps defines what a system will accomplish and how it accomplishes those steps. The operations and maintenance manual provides greater detail about each step, providing specific instructions, contact names, job functions, agency contacts, and interface information. Specific instructions are developed for the frequency of technology refreshes (e.g., how often the software or hardware is upgraded to a new release), performance monitoring and reporting, processes for handling identified issues, and level of support provided to the end user. In addition, detailed processes to be used for identifying, tracking, resolving, and recording all

system issues will be established. The elements of an O&M Manual can be found in section 4.13.1.

4.13.1 Elements of the O&M Manual

- **Introduction** - Describes the intended use of the system.
- **Theory of Operations** - Presents a description of the system design, operational functionality, interfaces, and environment sufficient to provide maintenance personnel a level of background understanding to facilitate troubleshooting and repair of the system, particularly for issues that may not be anticipated elsewhere in the O&M Manual.
- **Inventory** – Describes the inventory of system files and devices or provides a reference to where it is stored.
- **Environment** - Describes the hardware, software, operational activities, and other resources needed to perform actions described in this manual or provides a reference to where it is stored.
- **System Operations**- Describes normal system operational usage and activities or provides a reference to where it is stored.
- **System Architecture** - Describe the system, network, database, etc. architecture or provides a reference to where it is stored.
- **First-Time Users** - Instructions that are unique to first-time users of the system or provides a reference to where it is stored.
- **Access Controls** - Describes required access controls and privileges necessary for proper system installation or provides a reference to where it is stored.
- **Installation** – Provides detailed information on site preparation and system installation including specific step-by-step instructions, space requirements, power and cooling requirements, communications infrastructure needs, unpacking instructions, parts lists, and identification of any special tools or other accommodations required.
- **Configuration** – Describes the default and custom configuration, configuration options, and their associated definitions or provides a reference to where it is stored.
- **Starting the System** - Describes how to properly start the system or provides a reference to where it is stored.
- **Stopping the System** - Describes how to properly stop the system or provides a reference to where it is stored.
- **Instructions** - Describes proper usage of the system and system functionality or provides a reference to where it is stored.
- **Conventions and Error Messages** – Describes the specific system rules, error messages, and their associated definition or provides a reference to where it is stored.

- **Configuration Management**- Describes the system's configuration management process, any tools used to support it and mechanisms for related communications or provide a reference to where it is stored.
- **Change Management** - Describes the system's change management process, any tools used to support it and mechanisms for related communications or provides a reference to where it is stored.
- **Release Management** - Describes the system's release management process, any tools used to support it and mechanisms for related communications or provides a reference to where it is stored.
- **System Administration** - Describes how to properly administer and secure the system. Describe administrative options, messages, and their associated definitions or provides a reference to where it is stored.
- **System Maintenance** - Describes how system maintenance will be managed or provides a reference to where it is stored.
- **Backup and Recovery** – Describes how system backup and recovery will be managed or provide a reference to where it is stored. Include in this section are the procedures and sequences describing backup routines, media type, storage locations, and schedules.
- **Key Contacts** - Identifies key contacts and associated contact information or provides a reference to where it is stored.

4.13.2 O&M Manual Correspondence to Other Activities

The O&M Manual provides the uniform standard for installing, operating, and maintaining the system and therefore serves as input to a number of subsequent activities in the life cycle. Predominantly, it will serve as the primary reference material during deployment and installation of systems and throughout the Operations and Support Stage of the life cycle. It may also form the basis for development of training materials and will be required input for several OT&E related activities in the Operations Validation Stage.

4.14 Develop System Acceptance Test Plan and Procedures

The SAT is a formal test managed by the Project Manager (PM) and conducted by the PM designated developers and/or integrators for new systems not already fielded. The SAT is performed in a simulated and controlled environment for and focuses on the validation of technical requirements after the DT&E and SIT have been successfully completed. The purpose of SAT is to verify that the design solution meets all of the system technical requirements and is prepared for successful Operational Test and Evaluation (OT&E).

It should be noted that a change to an existing, already deployed system or capability that is significant enough to make use of the SE Life Cycle approach, will be subject to a System Test (ST) instead of the SAT. The ST is similar, but of somewhat different scope than the SAT and further information can be found in reference documents NWS Instruction 80–305 *Test and Evaluation* and NWS Instruction 80–306 *System Acceptance Test Process*.

4.14.1 System Acceptance Test Plan

The SAT Plan provides background information and documents test management, prerequisites and entrance criteria, test strategy and general methodology, test schedule and required personnel, success criteria, and documentation requirements for the SAT. The SAT Plan typically includes an introduction, purpose, objectives, evaluation criteria for each objective, test system configurations, test materials, test methodology, methods for deficiency adjudication, test focal points, and contact information. In its coverage of test management, the SAT Plan should discuss various roles and responsibilities such as the Program Management Committee (PMC), Configuration Control Board (CCB), Test Reviewed Group (TRG), Test Director, Program Manager, Test Team, and required support from the Information System Security Officer (ISSO) and NWS headquarters personnel. Elaboration on the various roles and responsibilities within the test and evaluation program can be found in reference document NWS Instruction 80–306 *System Acceptance Test Process*.

The SAT Plan is developed by the Systems Engineering Integration and Testing (SEIT) Branch in coordination with NWS Headquarters, NWS Regional Headquarters, Weather Forecast Offices, and other Federal agencies as required. The SAT Plan should be reviewed by the members of the TRG and other subject matter experts as appropriate, and must be approved by the Chief, SEIT Branch, prior to commencing the SAT.

4.14.2 Elements of the SAT Plan

The SAT Plan can be organized in the following manner:

- Introduction
- Purpose and test objectives
- Background/system description
- Prerequisites, assumptions, and limitations of system under test
- Test Strategy
- Test Management/Test Review Group
- Test schedule, facilities, resources
- Test conduct and planned activities
- Test reporting and analysis
- Conclusion

The SAT Plan also includes all Test Case Procedures and a Test Trouble Report (TTR) form as appendices. Typical outlines for a SAT/ST Plan are provided in Appendices of *NWS Instruction* 80–306 *System Acceptance Test Process*.

4.14.3 System Acceptance Test Procedures

Test Case Procedures are developed by the Test Team to evaluate the full functionality and operability of the system and may be designed to examine system hardware, system software, communications, and/or operations. In identifying test cases, the team references the System

Requirements Specification and traces test cases to the system requirements. As stated above, the SAT Test Case Procedures are included as an appendix to the SAT Plan.

4.14.4 Elements of the SAT Procedures

The SAT procedures should be organized as individual forms for each test case and should document the following:

- Cover sheet with description of the test scenario, purpose and objectives of the test, requirements covered by the test case, estimated time to complete, and criteria for success
- Specific test environment, test setup, and other prerequisite requirements
- Specific step-by-step instructions required to perform the test case
- Details of expected results or outcomes
- Pass/fail check-off
- Space for comments and annotations to procedures

An example of a typical Test Case Procedure is provided in Appendix D of NWS Instruction 80-306 *System Acceptance Test Process*.

4.14.5 Correspondence of SAT Plan and Procedures to Other Activities

The outputs of this activity consist of the SAT Plan and SAT Test Case Procedures and feed directly into the next activity, which is the actual SAT test. The SAT Plan and procedures provide the structure, management, schedule, set up, and specific instructions and reporting requirements for conducting SAT.

4.15 Conduct System Acceptance Test (SAT)

The SAT will typically be conducted by the development organization's test team within their facility using a separate environment built specifically for SAT. SAT is conducted in accordance with the approved SAT Plan and Procedures with witnessing and potential participation by designated Government representatives. Testing will be conducted according to the schedule in the SAT Plan and will consist of performing the test case procedures and recording results in the forms provided with the procedures. All test case procedures must be completed and the completed test case procedures are then included in the SAT Report and are part of the official test record. Any discrepancies discovered during SAT are documented in SAT Test Trouble Reports (TTR) and will be addressed by the development organization with corrective action plans and implementation of approved correction actions. Once discrepancies have been corrected and tested appropriately in the development and SAT environments, SAT regression testing is performed to verify the subject corrections, as well as to ensure that they had no adverse impacts on other system functionalities or performance.

Conducting the SAT is dependent on having the test environment thoroughly prepared, specific test setups in place for the test cases, and fulfillment of any other relevant entrance criteria

including obtaining an Authorization to Test (ATT). If the system and its environment are in not connected to any NWS operational systems or environments, the ATT for SAT can be granted by the Government Project Manager based on knowledge of system maturity and review of supporting artifacts and deliverables. If the system or its environment is connected in any way to any NWS operational system or environment, then issuance of the SAT ATT is also conditional upon approval by the appropriate NWS IT security officer.

Upon approval of the SAT, including any regression testing and associated documentation updates if required, the outputs of the conduct SAT activity consist of the SAT Report and the verified and accepted first article system. A successful SAT and the resultant SAT Report and verified system feed into later activities in the Operations Validation Stage, in particular preparation for and performance of OT&E of the system.

4.16 Develop Training Materials

Training materials are required to support training of operational personnel on how to use the system and installation and maintenance personnel on how to install, support, and maintain it. The team (which includes personnel from the Office of the Chief Learning Officer) will develop and/or oversee the development of the necessary materials for training on the newly developed system in a timeframe that will support initial training of Government operational and maintenance personnel involved in the preparations for and execution of OT&E.

4.16.1 Elements of the Training Materials

Training materials should meet the requirements of NOAA Standard S24.804 and include, but not be limited to:

- Lesson plans
- Trainee guides
- Visual aids
- Computer-based and / or web-based training materials
- Classroom materials
- Final examinations

Training materials should be provided in both paper and electronic form, when practical, with sufficient copies of classroom materials being provided for each student in each course with course materials being able to be retained by the students upon completion of each class.

Training materials are coordinated with training personnel in the appropriate divisions under the Office of the Chief Learning Officer (CLO). For most systems, this will include the Forecast Decision Training Division for operational personnel training and the Electronics and Information Technology Division for training for maintenance personnel. Coordination with other training divisions may be needed based on the type and scope of the newly developed system and the resulting training required.

Additional training materials that may be required based on the type and scope of the system must be coordinated with representatives under the CLO to support their overall recommended training strategy for the system. This may include cadre training materials designed to train NWS

trainers on how to perform training of system operations and maintenance personnel. This reduces the need for the system developer to perform all required training by giving the NWS the capability to perform that training on an ongoing basis throughout the system's life cycle.

4.16.2 Training Materials Correspondence to Other Activities

The training materials developed will be used to train all operations, installation, and maintenance personnel on the relevant aspects of the system to perform their duties regarding the system. Approved training materials also constitute an entrance criterion for the activities of Train OT&E Personnel, Conduct OT&E Readiness Review, Install Systems in OT&E Environment, and Conduct OT&E.

4.17 Develop Deployment Plan and Life Cycle Support Plan

The Deployment Plan describes the strategy and logistics of how, when, and by whom the deployment of the system will be executed. The Life Cycle Support Plan provides the strategy and logistics planning for how the system will be supported and maintained throughout its operational life cycle, typically starting with the OT&E phase. The activity of developing these documents may be concurrent with other development and testing activities that occur prior to OT&E.

4.17.1 Deployment Plan

The purpose of the Deployment Plan is to provide a single, common document that describes the strategy and logistics of how, when, where, and to whom the system will be deployed. The plan also identifies resources, communications plan, data conversion, rollout and transition schedule, and training. Preparation of the Deployment Plan should start as early in the life cycle as possible and be expanded at each stage to incorporate newly discovered information that influences deployment.

The Deployment Plan does not cover the user preparation and cutover activities; these are covered in training documentation and the PMRT, respectively. Specific deployment procedures for installation, back-out, rollback, transition to sustainment, and user training must also be supplied separately in stand-alone documents, in online help, or in change management systems, depending on the specific requirements of operations.

The following subsections describe suggested Deployment Plan organization and content. As all deployment processes are system-specific, some of the suggested content may not be relevant for the specific Deployment Plan; conversely, the plan may require more specific information than suggested below.

4.17.2 Elements of the Deployment Plan

Scope

This section identifies the recipients for the items identified in the Deployment Plan, the deployment type (e.g. phased, concurrent, parallel, etc.), the type of target of the deployment (e.g. sites, locations, facilities, users, etc.), and briefly introduces the organizations and teams that will perform the deployment of the system (e.g. Operations, independent contractors, etc.).

Overview

This section explains how the plan is organized

Constraints

This section provides the information on the deployment constraints and dependencies such as target physical environment for deployment, regulatory and statutory compliance (e.g. DoC/NOAA/NWS directives, international and industry standards, etc.), security controls, etc.

Dependencies

This section provides a description of all dependencies for the deployment, including applications, systems, financial, upstream processing, etc.

Definitions, Acronyms, and Abbreviations

This section provides definitions of all terms, acronyms, and abbreviations required to properly interpret the Deployment Plan.

References

This section provides a complete list of all documents referenced elsewhere in the Deployment Plan

Deployment Planning and Strategy

This section describes the deployment strategy and all activities performed throughout the deployment phase, including planning, system components shipping, installation, training, and support. The following subsections provide guidance on material that can be included in the section of the plan. Deployment Roles and Responsibilities - All roles and responsibilities of the members of the deployment team should be identified and listed. Identify technical and support personnel who will be involved in the deployment, including the managers, installers, testers, implementation team, and transition to Operations team, end users, and others.

Deployment Schedule

This section describes the schedule and milestones to conduct the deployment activities. Deployment milestones need to conform to the project milestones.

Site Readiness Assessment

This section discusses the locations where the elements of the system will be deployed, and the preparation required for the sites at which the system will operate.

Deployment Resources

This section describes hardware, software, facilities, and documentation, and any other resources, other than personnel, required for the deployment and installation.

Facility Specifics

This section is optional, depending upon the relevance of the topic to the project, and provides additional details about facilities required to install the system like a raised flooring, power

requirements, and special features to support privacy and security requirements that are unique to particular facilities.

Hardware

This section identifies the platform required to run and support the deployment, as required (hardware, Web-based, and so on), specifies the hardware model, versions, and configurations, and provides information about manufacturer support and licensing.

Software

This section identifies the software required to run and support the deployment, as required; specifies the software version and configuration, and provides information about manufacturer support and licensing.

Documentation

This section identifies and describes each information product that will be produced to aid in the installation, support, testing, transition to sustainment plan, or use of the deployed system (e.g. deployment guide, user guide, etc.).

Training

This section describes the plan and inputs for training the end users such that they can use and adapt the deployed system as required.

4.17.3 Life Cycle Support Plan

The Life Cycle Support Plan provides the performance-based logistics strategies for the system support and maintenance throughout its operational life cycle. This plan describes the approach and resources necessary to develop and integrate sustainment requirements into the system's design, development, testing and evaluation, fielding and operations.

4.17.4 Elements of the Life Cycle Support Plan

The Life Cycle Support Plan identifies and documents the program needs in the following areas:

- The maintenance and support concepts
- The life cycle supportability analysis
- The support performance metrics and how they will be achieved and sustained throughout the life cycle (including the regulatory requirements that influence sustainment performance)
- The assigned responsibilities and management approach for achieving effective and timely acquisition, product support, and availability throughout the system life cycle, including the integrated schedule for life cycle support
- The funding required and budgeted by year and appropriation for the main sustainment cost categories including operating & support costs
- The plan for identifying and selecting sources of repair or support
- The sustainment risk areas and mitigation plans

The Life Cycle Support Plan should either include necessary activities to implement the plan. These activities should be step by step instructions on how to prepare and execute the strategies defined in the Support Plan including names and contact numbers for support personnel when available, hours of operation, how reported faults will be categorized, service levels that will be provided, and other information considered relevant.

4.17.5 Deployment Plan and Life Cycle Support Plan Correspondence to Other Activities

This activity develops the plan for the Production and Deployment Stage and provides the guidance for the Operations and Support stage. These documents are critical to establish an understanding of the requirements for all teams during deployment and operations. It is also key to have these documents establishes before going to the next activity of developing the transfer plan.

4.18 Prepare Program Management Responsibility Transfer (PMRT) Plan

A Program Management Responsibility Transfer (PMRT) Plan may be developed and used to facilitate the transfer of a system's program management responsibilities from the acquisition organization to the operations organization. For systems for which responsibility will transfer during or after deployment to the operations and maintenance organization, the Project Manager and SE team prepares a PMRT Plan. The PMRT defines and guides the activities required for this transfer from the development, validation, and deployment stages into the subsequent Operations and Support Stage. In general, a PMRT is required for projects that result in a significant modification to the NWS enterprise, or when responsibility for the long-term O&M may not be clear.

To develop the PMRT, the acquisition Project Manager, with support from the SE Team, assembles key information regarding the project and coordinates with the management counterpart within the operations and maintenance organization. This coordination informs the team of the O&M organization's needs regarding the transfer of responsibility and will help drive the information and activities to be included in the PMRT. Once it is drafted, the acquisition PM coordinates the document internally and with the receiving O&M organization, to update the plan accordingly and obtain its approval. Development of the PMRT can occur at any time during the acquisition process, but must be approved in a timely fashion to support transition of the project into the Operations and Support Stage and must establish a date for the official transfer.

4.18.1 Elements of the PMRT Plan

Elements of the PMRT Plan may include, but not be limited to, the following areas:

- Definition of necessary management, technical, and logistics reviews to support the transition
- Listing of required transfer activities with a description of the activities and identification of responsible parties
- Schedule of transfer activities with firm date for final transfer

- Entry and exit criteria for the transfer, such as the system having achieved full production, certified as interoperable within the intended operational environment, and supportable as planned
- Listing of materiel to be transferred, such as documentation, drawings, hardware and software systems and subsystems, spares, training and training materials, specialized test and maintenance tools, and any development and/or maintenance environment(s) that may be required for ongoing support
- Identification of impacts to the O&M organization
- Various roles and responsibilities associated with the transfer as well as the ongoing operations, maintenance, and support of the system

4.18.2 PMRT Plan Correspondence to Other Activities

The output of this activity is the approved PMRT Plan. This plan will be executed during the Production and Deployment Stage to affect the transfer of responsibility of the system and move it into the Operations and Support Stage.

4.19 Outputs of the Design and Development Stage

The outputs or artifacts produced from the activities in the Design and Development Stage of the SE life cycle that are carried forward are:

- Approved System Requirements Specification (SRS)
- Requirements Traceability Matrix (RTM)
- Test and Evaluation Master Plan (TEMP)
- System Acceptance Test (SAT) Report
- Verified and accepted First Article system
- Training materials
- Deployment Plan
- Life Cycle Support Plan
- Installation and Deployment Procedures
- Program Management Responsibility Transfer (PMRT).

5 Operations Validation Stage

The Operations Validation Stage is the third stage in the life cycle this is the stage where the Government formally validates system installation and operations in the target operational environment. This includes validating that all requirements are fulfilled in order for the Government to operate and maintain the system at all deployed sites. This is done by conducting

an Operational Test and Evaluation (OT&E) of the System and is the primary focus of the Operations Validation Stage in the SE life cycle depicted in Figure 4 below.

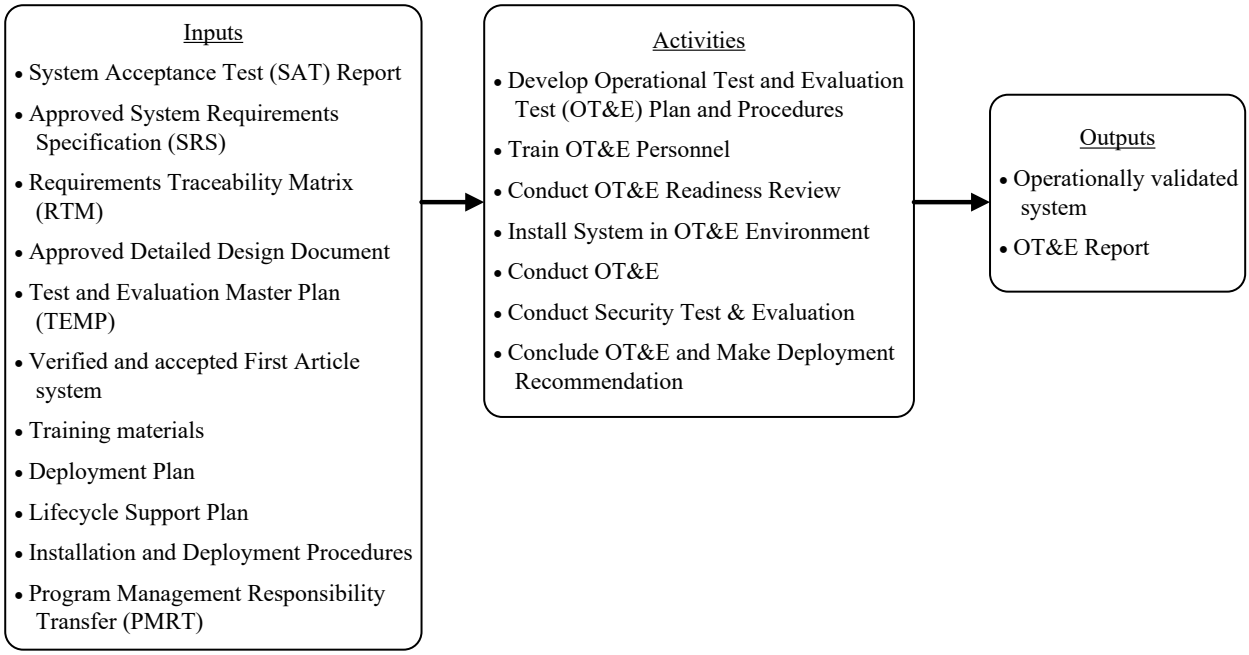

Figure 4: Operational Validation Stage

OT&E is a formal evaluation of a System that is conducted in an operational environment and is performed after the successful completion of the SAT or ST and subsequent deployment of the system at the designated OT&E operational site(s). This is not limited to operational compliance of the system itself, but includes all aspects of life cycle management of the system including: installation, documentation, performance, training, reliability, communications, information technology security, logistics, and maintenance support. OT&E must be successfully completed prior to full deployment of the System-Under-Test. The subject matter presented in this section is covered in more detail in *NWS Instruction 80-307 Operational Test and Evaluation Process.*

6 Production and Deployment Stage

The Production and Deployment Stage is where the system is manufactured, deployed, and installed at all required operational field sites (beyond those sites involved in OT&E). The Production and Deployment Stage is depicted below in Figure 5.

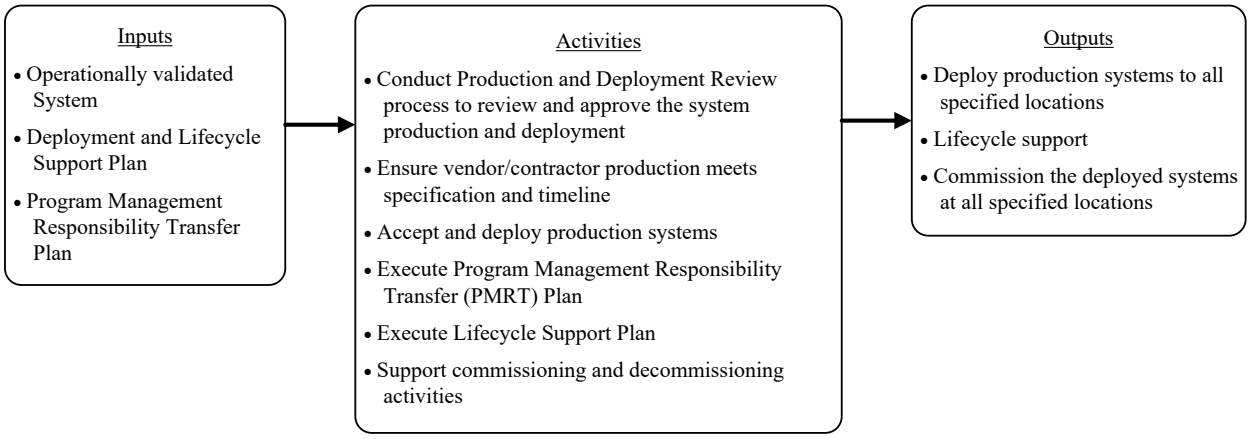

Figure 5: Production and Deployment Stage

The activities for this stage are provided in greater details can be found in NWSI 80-301 *Systems Engineering Process and Life Cycle.*

7 Operations and Support Stage

The Operations and Support Stage is where the software system is operated in its intended environment to deliver its intended services, representing the "steady state" period that lasts until the system is retired or replaced. Figure 7 below illustrates the inputs, activities and outputs of this stage. The activities for this stage are provided in greater details can be found in NWSI 80-301 *Systems Engineering Process and Life Cycle*:

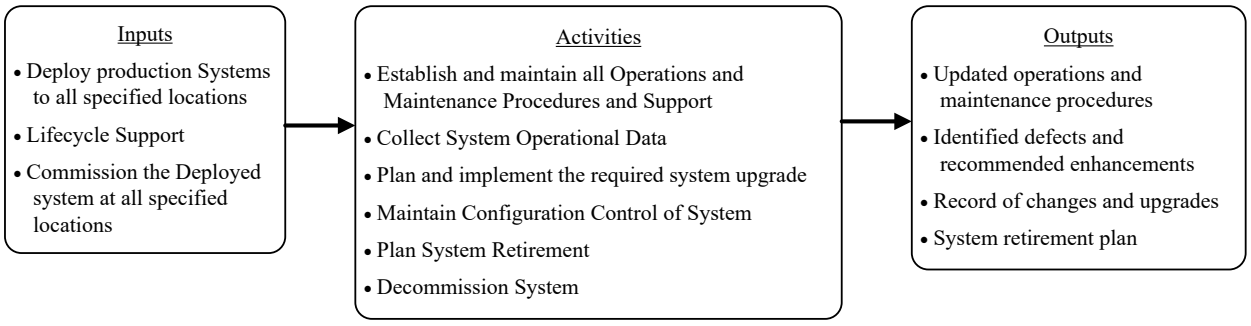

Figure 6: Operations and Support Stage

8 References and Glossary

This procedural directive is supported by the references and glossary of terms listed in Attachment 1.

Attachment 1

REFERENCES AND GLOSSARY OF TERMS

References

NWS Policy Directive 10-1, *NWS Requirements, Operations and Services Improvements*

NWS Instruction 10-103, *Operations and Services Improvement Process Implementation*

NWS Policy Directive 80-1, *Acquisition Program Management*

NWS Instruction 80-3, *Systems Engineering*

NWS Instruction 80-301, *Systems Engineering Process and Life Cycle*

NWS Instruction 80-304, *Systems Engineering for Software Development*

NWS Instruction 80-305, *Test and Evaluation*

NWS Instruction 80-306, *System Acceptance Test (SAT) Process*

NWS Instruction 80-307, *Operational Test and Evaluation (OT&E) Process*

NWS Policy Directive 80-4, *Science and Technology Planning and Programming*

NWS Policy Directive 80-5, *Science Review and Approval*

NWS Policy Directive 60-7, *Information Technology Security Policy*

NOAA Administrative Order 212-13, *Information Technology Security Management*

NIST Special Publication 800-37, *Guide for the Security Certification and Accreditation of Federal Information Systems*

NIST 800-53, *Recommended Security Controls for Federal Information Systems*

NIST 800-64, Revision 2, *Security Considerations in the System Development Life Cycle*

NIST 800-30, *Risk Management Guide for Information Technology Systems*

Glossary

The following is a list of common terms and acronyms used within the Systems Engineering industry. While many of these terms are not mentioned within the body of this document, they are nonetheless important to understanding this directive. These terms are taken from several sources, referenced sources contained from the directive.

Acceptance criteria - The criteria that a software component, product, or system must satisfy in order to be accepted by the system owner or other authorized acceptance authority.

Acceptance process - The process used to verify that a new or modified system is fully operational and meets the system owner's requirements. Successful completion of the acceptance process results in the formal transfer of the system responsibilities from development to maintenance personnel.

Acquisition - The acquiring by contract with appropriated funds of supplies or services (including construction) by and for the use of the Government through purchase or lease, whether the supplies or services are already in existence or must be created, developed, demonstrated, and evaluated.

Activity - A major unit of work to be completed in achieving the objectives of a project. An activity incorporates a set of tasks to be completed, consumes resources, and results in work products. An activity may contain other activities in a hierarchical manner. All project activities should be described in the Project Plan.

Allocated requirement - The subset of the system requirements that are to be implemented within the scope of a given project, and forming the components of the system.

Analysis - Use of mathematical modeling and analytical techniques to predict the compliance of a design to its requirements based on calculated data or data derived from lower system structure end product validations.

Analysis of Alternatives (AOA) - A formal analysis method that compares alternative approaches by estimating their ability to satisfy mission requirements through an effectiveness analysis and by estimating their life cycle costs through a cost analysis. The results of these two analyses are used together to produce a cost effectiveness comparison that allows decision makers to assess the relative value or potential programmatic returns of the alternatives.

Assumption - A condition that is taken to be true without proof or demonstration.

Audit - An independent examination of a work product to assess compliance with specifications, standards, quality or security requirements, contractual agreements, or other predetermined criteria.

Baseline - A set of configuration items (hardware, software, documents) that has been formally reviewed and agreed upon, that serves as the basis for further development, and that can be changed only through formal change control procedures.

Baselined requirements - The set of project requirements that have been approved and signed off by the system owner.

Code - Computer instructions and data definitions expressed in a development language or in a form that is output by an assembler, compiler, or other translator.

Code review - A meeting at which software code is presented to project personnel, managers, users, or other functional areas for review, comment, or approval.

Component - One of the parts that make up a system. A component may be hardware, software, or firmware and may be subdivided into other components.

Concept of Operations (ConOps) - The ConOps describes how the system will be operated during the life cycle phases to meet stake holder expectations. It describes the system characteristics from an operational perspective and helps facilitate an understanding of the system goals. It stimulates the development of the requirements and architecture related to the user elements of the system. It serves as the basis for subsequent definition documents and provides the foundation for the long-range operational planning activities

Configuration control - An element of configuration management consisting of the evaluation, coordination, approval/disapproval, and implementation of changes to configuration items after formal establishment of their configuration identification.

Configuration Control Board (CCB) - A group of people responsible for evaluating and approving/disapproving proposed changes to configuration items, and for ensuring implementation of approved changes.

Configuration item - An aggregate of hardware, software, or documentation components that are designated for configuration management and treated as a single entity in the configuration management process.

Constraint - A restriction, limit, or regulation that limits a given course of action or inaction.

Cost estimate - A formal estimate of the cost to develop and support a project.

Critical Design Review (CDR) - A review that demonstrates that the maturity of the design is appropriate to support proceeding with full-scale fabrication, assembly, integration, and test, and that the technical effort is on track to complete system development and operations in order to meet performance requirements within the identified cost and schedule constraints.

Dependency - A relationship of one task to another where the start or end date of the second task is related to the start or end date of the first task.

Design - The process of defining the architecture, components, interfaces, and other characteristics of a system, product, or component.

Design specification - A document that describes the design of a software component, product, or system. Typical contents include architecture, control logic, data structures, input/output formats, interface descriptions, and algorithms.

Enterprise - The aggregate of all functional elements, equipment, and processes which together accomplish a common mission.

Entry Criteria - Minimum accomplishments each project needs to fulfill to enter into the next life cycle phase or level of technical maturity.

Feasibility - The degree to which the requirements, design, or plans for a software product or system can be implemented under existing constraints.

Function Test - Confirms that the logically-grouped modules function according to specifications. Developers write this test from a user's perspective. Testing is based on output only, without any knowledge of internal code or logic.

Functional area - Any formally organized group involved in the development and maintenance of systems or the support of development and maintenance efforts, or other group whose input is required to successfully implement a systems project.

Functional Decomposition - A subfunction under logical decomposition and design solution definition, it is the examination of a function to identify subfunctions necessary for the accomplishment of that function and functional relationships and interfaces.

Functional requirement - A requirement that specifies a function that a software component, product, or system must be able to perform.

Goal - Quantitative and qualitative guidance on such things as performance criteria, technology gaps, system context, effectiveness, cost, schedule, and risk.

Hardware - Physical computer and other equipment used to process, store, or transmit computer programs or data.

Hierarchy - A structure in which components are ranked into levels of subordination.

Implementation requirements - A requirement that supports the development and maintenance concepts and approaches in the areas of operating environment, conversion, installation, training, and documentation.

Incremental development - A development technique in which requirements definition, design, implementation, and testing occur in an overlapping, iterative (rather than sequential) manner, resulting in incremental completion of the overall system or product.

Information Security - The protection of information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction in order to provide confidentiality, integrity, and availability. [44 U.S.C., Sec. 3542(b)(1)(A)-(C)]

Information System - A discrete set of information resources organized for the collection, processing, maintenance, use, sharing, dissemination, or disposition of information. [44 U.S.C., Sec. 3502(8)]

Information Technology - Any equipment or interconnected system or subsystem of equipment that is used in the automatic acquisition, storage, manipulation, management, movement, control, display, switching, interchange, transmission, or reception of data or information. [40 U.S.C., Sec. 11101(6)(A)]

Inspection - A static analysis technique that relies on visual examination of development products to detect errors, violations of development standards, and other problems. Code inspection and design inspection are two types.

Integration Test - Verifies the system components are integrated and working as an application. The technical development team performs this test to uncover errors that occur in the interactions and interfaces between components.

Integrity - The degree to which a software component, product, or system prevents unauthorized access to, or modification of, computer programs or data.

Interface - A shared boundary between two functional units, defined by functional characteristics, common physical interconnection characteristics, signal characteristics, or other characteristics, as appropriate.

Interface requirement - A requirement that specifies an external item with which a software product or system must interact, or that sets forth constraints on formats, timing, or other factors caused by such an interaction.

Interview technique - A technique employed for the identification, analysis, and documentation of the project requirements.

Life Cycle - See Project Life Cycle.

Life Cycle Cost - The total cost of ownership over the project's or system's life cycle from Formulation through Implementation. The total of the direct, indirect, recurring, nonrecurring, and other related expenses incurred, or estimated to be incurred, in the design, development, verification, production, deployment, operation, maintenance, support, and disposal of a project.

Logistics - The management, engineering activities, and analysis associated with design requirements definition, material procurement and distribution, maintenance, supply replacement, transportation, and disposal that are identified by space flight and ground systems supportability objectives.

Maintainability - The measure of the ability of an item to be retained in or restored to specified conditions when

Major Program/Project - Program/Project with total costs in excess of a predetermined threshold is deemed to be major program/project.

Margin - The allowances carried in budget, projected schedules, and technical performance parameters (e.g., weight, power, or memory) to account for uncertainties and risks.

Maintenance - The process of supporting a software product or system after delivery to maintain operational status, correct faults, improve performance or other attributes, or adapt to a changed environment.

Methodology - A collection of methods, procedures, and standards that defines an integrated synthesis of engineering approaches to the development of a work product.

Metric - The result of a measurement taken over a period of time that communicates vital information about the status or performance of a system, process, or activity. A metric should drive appropriate action.

Milestone - A scheduled event for which an individual or team is accountable and that is used to measure progress.

Mission - A major activity required to accomplish an Agency goal or to effectively pursue a scientific, technological, or engineering opportunity directly related to an Agency goal. Mission needs are independent of any particular system or technological solution.

Mission Critical System (MCS) - A system that is essential in the performance of a mission objective that if lost, would cause failure to meet or support the mission objective.

Module - A program unit that is discrete and identifiable with respect to compiling, combining with other units, and loading. A logically separable part of a program.

Performance requirement - A requirement that imposes conditions on a functional requirement (e.g., a requirement that specifies the speed, accuracy, or memory usage with which a given function must be performed).

Platform - A specific computer and operating system on which a software product or system is developed or operated.

Preliminary Design Review (PDR) - A review that demonstrates that the preliminary design meets all system requirements with acceptable risk and within the cost and schedule constraints and establishes the basis for proceeding with detailed design. It will show that the correct design option has been selected, interfaces have been identified, and verification methods have been described.

Procedure - A written description of a course of action to be taken to perform a given task.

Process - An ordered set of steps performed for a given purpose. Processes define or control the development of the project work products. The use of processes will ensure a consistent methodology across all platforms in producing the life cycle deliverables.

Project - An undertaking requiring concerted effort that is focused on developing or maintaining a specific software product or system.

Project Life Cycle - Covers all activities conducted within the scope of an entire project, from project startup to project closeout.

Project Management Plan - See Project Plan. Synonymous with software development plan and project plan.

Project Manager - The individual with total responsibility for all activities of a project. The project manager plans, directs, controls, administers, and regulates a project.

Project Plan - A document that describes the technical and management approach to be followed for a project. The plan typically describes the work to be done, the resources required, the methods to be used, the procedures to be followed, the schedules to be met, and the way the project will be organized.

Project team - The project manager, analysts, developers, and other staff assigned as the core group for a project.

Prototyping - A technique for developing and testing a preliminary version of the software product (either as a whole or in modular units) in order to emulate functionality.

Quality assurance - A process designed to provide management with appropriate visibility into the work products being built and the systems engineering processes being used by the project team.

Reference - A document(s) or other material that is useful in understanding more about an activity.

Regression Test - Re-execution of specific test cases to ensure defects are fixed, find new defects that may have been introduced, and confirm that module(s) are functioning properly.

Reliability - The ability of a software or system component to perform its required functions under stated conditions for a specified period of time.

Requirement - A condition or capability needed by a system owner/user to solve a problem or achieve an objective. A condition or capability that must be met or possessed by the software product or system to satisfy a contract, standard, specification, or other formally imposed documents.

Requirements analysis - The process of analyzing and understanding the scope and feasibility of identified requirements; of developing a preliminary plan to arrive at a detailed definition of system, hardware, or software requirements; and of crystallizing a preliminary system solution.

Requirements management - A process designed to establish a common understanding between the system owner/users and the project team regarding the system owner/users' software and system requirements. This understanding forms the basis for estimating, planning, performing, and tracking the project's activities throughout the lifecycle.

Requirements Specification - A work product deliverable that specifies the requirements for a software product or system. Typically included are functional requirements, performance requirements, and interface requirements. Describes in detail what will be delivered in the product or system release.

Retirement - Permanent removal of a system or software product from its operational environment.

Risk - The combination of the probability that a program or project will experience an undesired event (some examples include a cost overrun, schedule slippage, safety mishap, health problem, malicious activities, environmental impact, or failure to achieve a needed scientific or technological breakthrough or mission success criteria) and the consequences, impact, or severity of the undesired event, were it to occur. Both the probability and consequences may have associated uncertainties.

Software - Computer programs, procedures, and associated documentation and data pertaining to the operation of a software product or system.

Specification - A document that specifies in a complete, precise, verifiable manner the requirements, design, behavior, and other characteristics of a software component, product, or system.

Stage - A partition of the project life cycle that reduces a project to manageable pieces and represents a meaningful and measurable set of related tasks that are performed to obtain specific work products.

Stakeholder - An individual, group, or organization, who may affect, be affected by, or perceive itself to be affected by a decision, activity, or outcome of a project.

Stakeholder Requirements - Requirements from various stakeholders that will govern the project, including required system capabilities, function, and/or services; quality standards; system constraints; and cost and schedule constraints. Stakeholder requirements may be captured in the Stakeholder Requirements Specification.

Standard - Mandatory requirements employed and enforced to prescribe a disciplined, uniform approach to software and systems development and maintenance.

State Diagram - A diagram that shows the flow in the system in response to varying inputs.

Structured analysis - An analysis technique that uses a graphical language to build models of software products or systems. The four basic features in structured analysis are data flow diagrams, data dictionaries, procedure logic representations, and data store structuring techniques.

System- 1)An integrated set of elements, subsystems, or assemblies that accomplish a define objective. These elements include products (hardware, software, firmware), processes, people, information, techniques, facilities, services, and other support elements. 2) A combination of interacting elements organized to achieve one or more stated purpose.

Systems Analysis - The analytical process by which a need is transformed into a realized, definitive product, able to support compatibility with all physical and functional requirements and support the operational scenarios in terms of reliability, maintainability, supportability, serviceability, and disposability, while maintaining performance and affordability.

System Architecture - The arrangement of elements and subsystems and the allocation of functions to them to meet system requirements.

System Design Document - A work product deliverable that describes the solution to the automation task as described by the requirements.

Systems Engineer - An engineer trained and experienced in the field of systems engineering.

Systems Engineering Processes - A logical, systematic set of processes selectively used to accomplish systems engineering tasks.

System of Interest-The system whose life cycle is under consideration.

System & Standards Test – Verifies functional business requirements, business processes, data flows and other system criteria are met.

Task - The smallest unit of work subject to management accountability. A task is a well-defined work assignment for one or more project team members. Related tasks are usually grouped to form activities. A task is the lowest level of work division typically included in the Project Plan and Work Breakdown Structure.

Telecommunications - The science and technology of communications by electronic transmission of impulses, as by telephone or email.

Test - The use of system, subsystem, or component operation to obtain detailed data to verify performance or to provide sufficient information to verify performance through further analysis. Testing is the detailed quantifying method of verification and is ultimately required in order to verify the system design.

Test case - A set of conditions or variables under which a tester determines whether or not an application or software system is working correctly. It is the mechanism for determining whether a software program or system has passed or failed

Test criteria - The criteria that a software product or system component must meet in order to pass a given test.

Test documentation - Documentation describing plans for, or results of, the testing of a system component or product. Documents typically include test case specifications, test incident reports, test logs, test plans, test procedures, and test reports.

Traceability - The degree to which a relationship can be established between two or more products of the development process, especially products having a predecessor-successor relationship to one another.

Transition - The act of delivery or moving of a product from the location where the product has been implemented or integrated, as well as verified and validated, to a customer. This act can include packaging, handling, storing, moving, transporting, installing, and sustainment activities.

User - The general population of individuals who use a software product or system. User activities can include data entry; read only; add, change and delete capabilities; querying; and report generation.

User interface - An interface that enables information to be passed between a user and hardware or software components of a computer system.

User manual - A document that presents the information necessary to use a software product or system to obtain desired results. Typically described are product or component capabilities, limitations, options, permitted inputs, expected outputs, possible error messages, and special instructions.

Validation - The process of evaluating software or systems at the end of the development process to assure compliance with established software and system requirements.

Verification - The process of evaluating a software product or system to determine whether or not the work products of a stage of the project life cycle fulfill the requirements established during the previous stage.

Walkthrough - An analysis technique in which a team of subject matter experts review a segment of code, documentation, or other work product, ask questions, and make comments about possible errors, violation of development standards, and other problems.