Systems Engineering for ITS

Table of Contents

1 Introduction ... 4
1.1 Purpose ... 4
1.2 Intended Audience ... 4
1.3 Navigating ... 5
1.4 About the Icons ... 5
2 What Is Systems Engineering? ... 7
2.1 Definitions ... 7
2.1.1 What is a System? ... 7
2.1.2 What is Systems Engineering? ... 7
2.2 Key Principles ... 8
2.3 Why use Systems Engineering? ... 8
2.4 Transportation Context for Systems Engineering ... 9
2.5 When to use Systems Engineering? ... 12
2.6 Cost and Schedule Impacts of Systems Engineering ... 13
2.7 Systems Engineering Life Cycle Models ... 14
2.7.1 Life Cycle Model Overview ... 14
2.7.2 Sequential Methods ... 15
2.7.3 System vs Project Life Cycles ... 18
2.7.4 Incremental Methods ... 19
2.7.5 Iterative Methods ... 20
2.8 US DOT Regulations ... 23
3 The Systems Engineering Process ... 27
3.1 Key Process Topics ... 27
3.2 High level SE Process Overview ... 28
3.3 SE Process Steps ... 29
3.3.1 Vee Overview ... 29
3.3.2 Regional ITS Operations Planning ... 31
3.3.3 Project Identification and Scoping ... 36
3.3.4 Project Planning ... 42
3.3.5 Concept of Operations ... 47
3.3.6 Requirements ... 55
3.3.7 Design and Specifications ... 64
3.3.8 Software and Hardware Implementation ... 76
3.3.9 Integration and System Verification ... 84
3.3.10 Deployment and Acceptance ... 91
3.3.11 Validation ... 97
3.3.12 Operations and Maintenance ... 102
3.3.13 Retirement/Replacement ... 108
3.4 Cross-Cutting Processes ... 113

3.4.1 Project Management ... 113
3.4.2 Configuration Management ... 119
3.4.3 Traceability ... 125
3.4.4 Risk Management ... 126
3.4.5 Trade Studies ... 133

4 Managing SE Projects ... 137
4.1 Intro to the Agency Perspective ... 137
4.2 Risk Evaluation ... 138
4.3 Process Tailoring ... 144
4.4 ITS Procurement ... 147
4.4.1 Request for Information ... 147
4.4.2 Request for Qualifications ... 148
4.4.3 Request for Proposal ... 149
4.4.4 Invitation to Bid ... 149
4.4.5 Public-Private Partnership ... 149
4.4.6 Implementation Models ... 149
4.4.7 Staffing Options ... 151
4.5 Agency Implementation of Systems Engineering Processes ... 154
4.5.1 Regulation Requirements ... 154
4.5.2 Implementing the SE Process ... 156

5 Systems Engineering Resources ... 159
5.1 ITS Specific Publications ... 159
5.2 General Systems Engineering References ... 159
5.3 Systems Engineering Training ... 160
5.4 Systems Engineering Tools ... 160

6 Systems Engineering Documentation ... 161
6.1 Project Management Plan ... 161
6.1.1 Purpose of this Document ... 161
6.1.2 Tailoring This Document to Your Project ... 161
6.1.3 Checklist: Critical Information ... 161
6.1.4 Template ... 162
6.2 Systems Engineering Management Plan (SEMP) ... 165
6.2.1 Purpose of this Document ... 165
6.2.2 Tailoring this Document to Your Project ... 165
6.2.3 Checklist: Critical Information ... 165
6.2.4 Template ... 165
6.3 Configuration Management Plan (CMP) ... 172
6.3.1 Purpose of this Document ... 172
6.3.2 Tailoring this Document to Your Project ... 172
6.3.3 Checklist: Critical Information ... 172
6.3.4 Template ... 172
6.4 Concept of Operations Template ... 176
6.4.1 Purpose of this Document ... 176

6.4.2 Tailoring this Document to Your Project 176
6.4.3 Checklist: Critical Information 176
6.4.4 Template 178
6.5 Requirements Template 181
6.5.1 Purpose of This Document 181
6.5.2 Tailoring this Document to Your Project 181
6.5.3 Checklist: Critical Information 181
6.5.4 Template 182
6.6 Design Specification Template 184
6.6.1 Purpose of these Documents 184
6.6.2 Tailoring these Documents to Your Project 184
6.6.3 Checklist: Critical Information 184
6.6.4 Templates 185
6.7 Integration Plan Template 192
6.7.1 Purpose of this Document 192
6.7.2 Tailoring this Document to Your Project 192
6.7.3 Checklist: Critical Information 192
6.7.4 Template 192
6.8 Verification Documents Template 195
6.8.1 Purpose of these Documents 195
6.8.2 Tailoring these Documents to Your Project 195
6.8.3 Checklist: Critical Information 195
6.8.4 Templates 196
6.9 Deployment Plan Template 201
6.9.1 Purpose of this Document 201
6.9.2 Tailoring this Document to Your Project 201
6.9.3 Checklist: Critical Information 201
6.9.4 Template 202
6.10 Validation Documents Template 204
6.10.1 Purpose of these Documents 204
6.10.2 Tailoring these Documents to Your Project 205
6.10.3 Checklist: Critical Information 205
6.10.4 Template 206
6.11 Operations & Maintenance Plan Template 209
6.11.1 Purpose of this Document 209
6.11.2 Tailoring this Document to Your Project 210
6.11.3 Checklist: Critical Information 210
6.11.4 Template 210
7 Glossary and Acronyms 214
7.1 Glossary 214
7.2 Acronyms 221

1 Introduction

1.1 Purpose

This updated document is intended to introduce transportation professionals to a needs-focused, requirements-driven engineering process that minimizes the risk of procuring technology that does not meet their needs. In the broader industry, needs-focused, requirements-driven processes are known as systems engineering, and this document provides an explanation of this process and how transportation professionals can tailor it to manage intelligent transportation system (ITS) projects. This document describes the systems engineering project life cycle and details how to manage a systems engineering process. It also describes how to begin implementing the systems engineering approach on your next ITS project and incorporate it more broadly into your organization's business processes and practices. This is the fourth generation of the Systems Engineering for ITS content, published December 2022. It combines and updates content that was previously included in the FHWA Systems Engineering Guidebook (SEGB) and Systems Engineering Handbook (SEHB) publications.

This edition of Systems Engineering for ITS is consistent with the 4th Edition of the International Council on Systems Engineering's (INCOSE) Systems Engineering Handbook – A Guide for System Life Cycle Processes and Activities. As the primary professional society in the Systems Engineering field, the INCOSE handbook is consistent with international standard ISO/IEC/IEEE 15288:2015 (Systems and software engineering-System life cycle processes).

Reading this document will provide an overview of how agencies can manage an ITS project using the systems engineering process. To support this objective, the document provides an introduction to the systems engineering process, but it is not intended to make you a systems engineering expert. Many excellent and comprehensive resources are available that describe every aspect of systems engineering in detail. These resources are identified throughout the document in case you want more information.

This document is a resource and a learning tool on the topic of systems engineering. It is not formal guidance from US DOT on how to meet the systems engineering requirements in 23 CFR part 940, Intelligent Transportation System Architecture and Standards which applies to both Regional ITS Architectures (940.9) and ITS project Systems Engineering Analysis (SEA) (940.11). The corresponding FTA National ITS Architecture Consistency Policy for Transit Projects applies to FTA ITS projects and has essentially the same wording as the FHWA 23 CFR 940. Compliance with the Regulation/Policy is established by each FHWA Division and FTA Regional Office. It is strongly recommended that you contact your federal representative for the specific requirements in your state.

1.2 Intended Audience

This document is designed primarily for public sector managers and staff who deploy ITS projects. System owners, operators, maintainers, private sector transportation professionals or anyone else in need of a primer on the basics of systems engineering for ITS will also find parts of this document useful. The document assumes the reader has a transportation background and knows something about ITS, but you don't need any previous knowledge of systems engineering to benefit from this document.

You might have noticed that systems engineers are not included in the above list. Systems engineers should already be familiar with the systems engineering content included here, though they may benefit from discussion of the application of systems engineering to ITS as well as the discussion of how to tailor the systems engineering effort to different types of ITS projects. The document is intended for transportation professionals who are involved in ITS project development and will need to know something about systems engineering in order to manage the contractor or agency teams developing an ITS project.

1.3 Navigating

This document will introduce you to systems engineering and then describe how systems engineering can be applied to your ITS projects.

Here is a breakdown of the 6 remaining chapters and what you will find in each:

Chapter 2: What Is Systems Engineering? sets the stage for the following chapters by defining some key terms and explaining the guiding principles behind systems engineering. Various systems engineering project lifecycle models are introduced. approaches. The chapter also gives some background on the FHWA Regulation and FTA Policy requirements for systems engineering.

Chapter 3: The Systems Engineering Process follows an ITS system from initial operations planning all the way through retirement of the implemented system. The systems engineering approach is described in the context of SE project life cycle models and steps through topics like Concept of Operations, Requirements, Design and Testing. The chapter also includes a discussion of key cross-cutting processes that apply to many of the steps in the life cycle.

Chapter 4: Managing SE Projects provides an agency perspective on managing the systems engineering for an ITS project. It discusses evaluating the risk associated with a project and considers the actions an agency might engage in depending on the level of risk identified for the project. The chapter concludes with a discussion of the FHWA regulation/ FTA policy relating to Architecture and Standards, including a discussion of the requirements for Systems Engineering Analysis.

Chapter 5: Systems Engineering Resources lists many excellent books, reports, training courses, and other systems engineering resources that you can use to learn more about any of the systems engineering topics that are introduced in this document.

Chapter 6: Systems Engineering Documentation provides a series of templates for each of the systems engineering documents that could be developed on a specific project.

Chapter 7: Glossary and Acronyms provides definitions of terms used in the document.

1.4 About the Icons

Icons are used to highlight different kinds of information throughout this document.

This “lightbulb” icon identifies suggestions that may improve the systems engineering analysis or the quality of the systems engineering products that are created. Usually based on actual experience, these are ideas that have worked in the past.

This “exclamation point” icon flags warnings. In contrast to tips, these are problems that have been encountered that you should avoid. Also frequently based on actual experience, these are ideas that have NOT worked in the past.

This “laptop” icon highlights resources that offer additional information related to systems engineering, including books, reports, presentations, and other documents. Chapter 5 includes a list of all the resources that are identified in this document.

This “scissors” icon identifies ways that the systems engineering process can be tailored for smaller ITS projects. Many ITS projects are relatively low risk and low complexity, and the systems engineering process should be tailored accordingly. Section 4.3 provides a more comprehensive discussion of how to tailor the systems engineering approach

This “hammer and wrench” icon identifies software tools (programs, databases, spreadsheets, etc.) that support some aspect of the systems engineering or ITS project development processes. The information provided is not intended to endorse or recommend any particular tool.

This “scales” icon identifies references to the FHWA Regulation and FTA Policy on ITS Architecture and Standards. These are normally references to the portion of the regulation/policy related to systems engineering analysis (Sections 940.11 of the Regulation and VI of the Policy). (See https://www.ops.fhwa.dot.gov/its_arch_imp/policy.htm regarding the Regulation/Policy.)

This “book” icon is used where ITS and systems engineering terminology is defined. Terminology is one of the first hurdles to overcome in any new subject area. Readers can skip quickly past the definitions of familiar terms.

2 What Is Systems Engineering?

This chapter sets the stage for the following chapters by defining some key terms and explaining the guiding principles behind systems engineering. We also introduce various systems engineering project lifecycle models based on different approaches

2.1 Definitions

2.1.1 What is a System?

Everyone uses the term and has an intuitive notion of what a system is, but there is a formal definition. INCOSE and ISO/IEC/IEEE defines a system as:

"A combination of interacting elements organized to achieve one or more stated purposes."

This general definition covers almost everything you can think of – household appliances, transportation management systems, the latest weapon system – all of these are systems.

The two elements of this definition drive many of the processes we use. They are:

- Elements that interact. A system is a sum of parts that must work together.
- Stated purpose. Those elements interact to serve a clearly defined purpose.

Without understanding the purpose, defining what a project must build becomes impossible. But with the purpose defined, those parts and how they are integrated can be measured at every step of procurement from preliminary design to operation.

Reference: ISO/IEC/IEEE 15288-2015, ISO/IEC/IEEE International Standard – Systems and software engineering – System life cycle processes, https://standards.ieee.org/ieee/15288/5673/

2.1.2 What is Systems Engineering?

To understand "What is Systems Engineering?" it is useful to discuss what systems engineers do. Systems engineers are concerned about the "big picture" of a project. They oversee all aspects of a project in a variety of fields, such as electrical, civil, transportation and manufacturing. Systems engineers collaborate with project team members to ensure that the parts (e.g., software, hardware, interfaces, security systems, databases, users, etc.) of the project work together to accomplish its stated purpose. Systems engineering is needs focused and requirements driven.

More formally, the systems engineering organization called the International Council on Systems Engineering (INCOSE) defines systems engineering as follows: Systems Engineering *is a transdisciplinary and integrative approach to enable the successful realization, use, and retirement of engineered systems, using systems principles and concepts, and scientific, technological, and management methods.*

- *Transdisciplinary* - "transcends" all of the disciplines (i.e., fields of study) involved, and organizes the effort around common purpose, shared understanding and "learning together" in the context of real-world problems or themes.
- *Integrative* - involves either interdisciplinary (e.g. integrated product teams) or multi-disciplinary (e.g. joint technical reviews) methods
- *Engineered Systems - are a composite of people, products, services, information, and processes (and possibly natural components) that provides a capability that satisfies a stated customer need or objective.*

- *Systems Principles and Concepts* - are the ways that systems thinking and the systems sciences infuse systems engineering.

Link to a more detailed definition of SE: https://www.incose.org/about-systems-engineering/system-and-se-definition/systems-engineering-definition

Learn more about Systems Engineering at https://www.incose.org/about-systems-engineering .

Learn more about INCOSE at https://www.incose.org/about-incose.

2.2 Key Principles

Some of the key systems engineering principles are:

Viewing the system from the stakeholder points of view, this means walking in the shoes of the system's owner and stakeholders. Key processes for this principle include needs assessment, user need elicitation, developing a Concept of Operations, and especially stakeholder involvement.

Start at the finish line defines the expectations for the system and the way the system is going to operate. The details might change but the key concepts and ideas on what the system should do to meet particular user needs should remain consistent. Key processes for this principle include Concept of Operations and Validation Plan.

Address risks as early as possible when the cost impacts to addressing those risks are the lowest. Key processes for this principle include risk management, requirements, and stakeholder involvement.

Push technology choices to the last possible moment. Define *what* is to be done before defining *how* it is to be done.

Focus on interfaces of the system during the definition of the system. Defining clear and standard interfaces and managing them through the development will ease the integration of the individual elements of the system.

Understand the organization of the system's owner, stakeholders, and development team.

2.3 Why use Systems Engineering?

The overall goals of systems engineering are to reduce risks, identify and correct defects as soon as possible, provide a common language between subject matter experts (SMEs) and system designers/technologists, and a system design that responds to **needs** and **requirements** instead of dictating **needs** and **requirements**

Other benefits of systems engineering include:

- better system documentation
- higher level of user engagement
- system functionality that meets user needs
- potential for shorter project cycles
- systems that can evolve with a minimum of redesign and cost
- higher level of system reuse
- more predictable outcomes from projects

2.4 Transportation Context for Systems Engineering

Systems Engineering is primarily a project development process and is performed within the larger context of transportation planning and operations for a region. This section will provide some of the context regarding how systems engineering fits into operations and operations planning.

Operations comprise the activities performed by transportation agencies to achieve the most value from the transportation network, as currently constructed. Operations includes traffic management and a range of other activities, with the goal of maximizing the return on the infrastructure investment. The value of the network is adversely affected principally by congestion, which occurs whenever demand exceeds capacity in some part of the network. Congestion is marked by persistent, residual queueing, which imposes increased delay, interruptions of continuous flow (that may contribute to crashes), unreliable travel times, increased emissions, and reduced quality of life.

Congestion may be inevitable. Increasing demand for movement of people, goods and services is the natural product of increased economic and social activity, which are identified as goals by most state and local governments that manage traffic. Thus, increasing demand is the outcome of desirable trends. Network capacity, on the other hand, is limited, and the ability to increase it is constrained by many factors, some of which are also closely tied to perceptions of the quality of life. Thus, transportation operators are faced with increasing operational challenges in trying to keep congestion at bay. To minimize congestion, operations strategies either seek to increase or optimize capacity, or they seek to manage demand.

Capacity may be increased by expanding the built infrastructure, but doing so incurs a high cost and requires a long-term process of planning to address environmental and societal challenges. But capacity may also be increased by making use of existing infrastructure more efficiently, even in temporary ways. For example, part-time use of shoulders during congested conditions is a strategy for increasing capacity. Adjusting signal timing to favor the congested direction, particularly to serve a larger network need, is also a strategy for increasing capacity.

Many operational activities undertaken by agencies also seek to manage demand. Demand management includes strategies that seek one or more of the following three objectives:

- Spatial diversion, where traffic is encouraged to use an alternate route that avoids congestion in order to minimize the congestion delay faced by most travelers.
- Temporal diversion, where travelers are encouraged to choose a less congested time of day to make their trip.

- Modal diversion, where travelers are encouraged to choose a less congested mode, particularly a high-density mode such as transit.

Transportation agencies and private-sector providers employ a range of tools to empower travelers to seek these alternatives. Modal and temporal diversion, however, affects travelers *before* they begin their trip, and strategies to attain these objectives are based on providing information to travelers before travel, or at least before relevant portions of their trips. Travelers are empowered to choose different departure times or make decisions based upon additional information—information about current network conditions or information they need to explore alternatives. A substantial and increasing body of technology has and will be implemented to carry out these strategies. Many of these technologies are implemented by private-sector service providers, and focus on information provided directly to travelers on devices they carry with them.

Once the trip has begun, travelers no longer have the option to change departure times or modes, and spatial diversion becomes the principal objective that transportation agencies seek to attain in order to manage congestion. Much traditional operational technology involving ITS field devices carries out these strategies, including such technologies as dynamic message signs, highway advisory radio, lane control signals, and the like. But agencies are expanding their operational strategies to use emerging technologies, including routing services that depend on macroscopic performance data—from agency-owned sensors as well as crowd-sourced—to recommend congestion-avoidance routes.

Clearly, attaining these objectives requires that data-gathering technologies, information technologies, and control technologies integrate fully and meaningfully. It also requires that the agencies who own or are served by these technologies work together at the agency to agency level.

The integration of agency activities and the integration of their technologies are separate challenges. Agency-level integration suggests robust regional planning processes, where all the agencies in a region develop a consensus around their goals, objectives, and strategies. Agencies in a region must:

- Understand the nature of the congestion problem
- Devise and document the operational strategies that might minimize the effects of those problems
- Plan the activities undertaken by each agency to carry out those strategies
- Develop within each agency the capability to reliably undertake those activities (including staffing, facilities, management structures, equipment, administrative and business support, among others)
- Determine the systems and technologies needed to support those strategies
- Develop an implementation plan by which those systems and technologies will be implemented and integrated, formulated as a series of projects
- Carry out that plan with a series of projects, with engineering processes in place that ensure those projects will efficiently attain the objectives driving the plan

Only the last two bullets are *project-related.* A strong program of ITS projects depends fundamentally on a strong regional operations planning process that first defines what agencies will do as the basis for determining what technologies they need.

Systems engineering is the process by which the operational strategies articulated by an agency become fully and efficiently supported by the technology they build. Stated another way, *systems engineering* *helps agencies ensure that their scarce operational capital funds are spent in a way that most directly serves the attainment of their operational goals and objectives*.

Project success is therefore measured by how well the implemented technology supports the needs of the people who use it, as embodied in the activities they undertake to attain their objectives.

Articulating and documenting objectives should flow from a planning process that determines the nature, scope, and basic budget for a project. An understanding of goals is an important input to the deployment of a project to support transportation system operation that contributes to the achievement of those goals. The outputs of the planning process help support program and project development.

Some planning activities are parts of formal planning processes that conform to guidelines required for the expenditure of the Highway Trust Fund. Some are not defined formally in that way, but provide a contextual clarity for operations technology projects that reduces the workload required to develop systems engineering products. Figure 1 shows a representation of this relationship between planning and project focused processes.

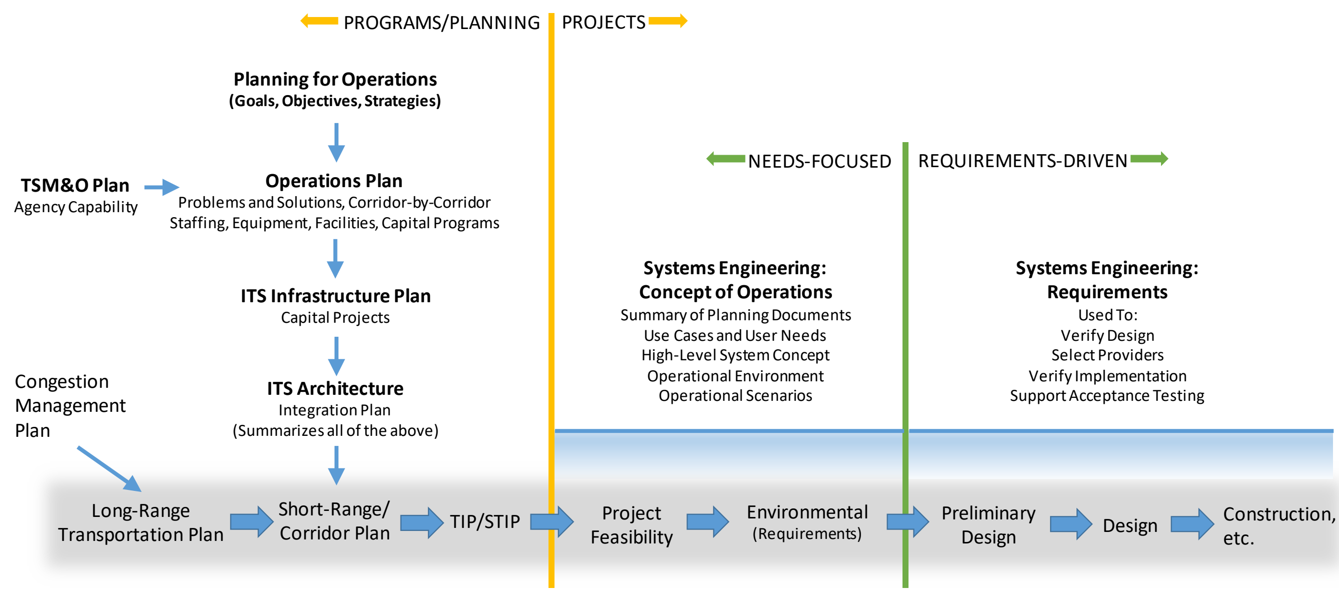


Figure 1. The Relationship of the Planning Process to the Systems Engineering Process

(Source: FHWA)

The importance of the planning foundation for ITS projects cannot be overstated. Technology projects support operational activities, and operational activities attain operational objectives, which are drawn from organizational and regional goals. The exact organization of the planning documents shown at the left in **Figure 1** is not formally defined, and they can be combined or developed concurrently. No matter how they are developed, however, they are mutually informed to provide a consistent regional picture of operational activities and how integrated ITS supports those activities.

The question of *who* develops these documents helps clarify what they are about. The Long-Range Transportation Plan, the Congestion Management Plan, short-range and corridor plans, and the

Transportation Improvement Program are formal planning documents subject to specific requirements and oversight. These documents are prepared by transportation planners, often in collaboration between metropolitan planning organizations and local agencies, and with support from operations professionals in those agencies. The Operations Plan (whatever it may be called in a region) should primarily be a product of operations professionals in regional agencies, perhaps with support and facilitation from MPOs. The Regional Architecture is a collaboration between transportation planners and operations professionals. Agencies often seek assistance from consultants and contractors for developing these various documents, but they will only be meaningful and useful if they accurately reflect the goals, objectives, and strategies of the transportation agencies in a region. To realize the full value of those documents, agencies must remain fully engaged in their development, and be prepared to own what those documents say.

These planning documents explain and document the *why* of projects and their basic strategic operational approach, and are developed as a collaboration between planners and agency operations professionals. A project formulated to install dynamic messages signs, for example, might attract the question, "why dynamic message signs instead of, say, smart phone apps?"   These questions are best addressed during the planning process.

When that planning support has not been developed, systems engineering must attempt to fill that vacuum, and Concepts of Operation end up with extended discussions of planning questions and answers, instead of focusing on the users of the system being implemented and their specific needs. The lack of planning puts a burden on the systems engineering process that it is not designed to bear.

Good supporting planning greatly reduces the effort required for systems engineering, even for complex projects where systems engineering documents will be developed from scratch as part of the project development.

The reader should take from this section simply that systems engineering is a project process that depends on a strong transportation and operational planning foundation.

2.5 When to use Systems Engineering?

Systems engineering mitigates the risk of technology-based projects not meeting the needs of their users. Therefore, agencies should use systems engineering approaches when projects demonstrate risk, and the detail applied to the SE process should be commensurate with that risk. State departments of transportation develop procedures in consultation with their FHWA Division office primarily aimed at assessing risk as the basis for determining how much systems engineering the agency should undertake. Agencies should determine and follow these procedures.

Section 2.8 describes the Federal systems engineering requirements for an ITS project. Section 4.5 provides a general discussion of the processes used by some state DOTs for determining the level of required systems engineering.

Even in low-risk projects, however, agencies should address all the elements of a complete systems engineering process, including user needs and requirements in advance of design, procurement, and implementation, followed by verification and validation steps to demonstration that requirements have been fulfilled and needs have been satisfied. Lower-risk projects, however, may use needs and requirements that have already been documented as part of some prior process, and need less detail in each of these steps. Higher-risk projects may need addition systems engineering activities to fully document needs and requirements in sufficient detail to mitigate identified risks. And the highest-risk projects may need to be conducted in phases to minimize investment until the most serious risks have been addressed.

2.6 Cost and Schedule Impacts of Systems Engineering

Several studies, including studies performed by the International Council of Systems Engineering, Boeing, IBM, and many others, demonstrate that systems engineering results in better cost and schedule performance. **Figure 2** shows the results of an INCOSE study that collected planned and actual project cost data and systems engineering cost data for 43 projects. The survey indicated that investing in systems engineering improved both project cost and schedule performance. The responses indicated a 50% overrun in cost and an 80% overrun in schedule on average without systems engineering and a clear trend towards better cost and schedule performance results with systems engineering.

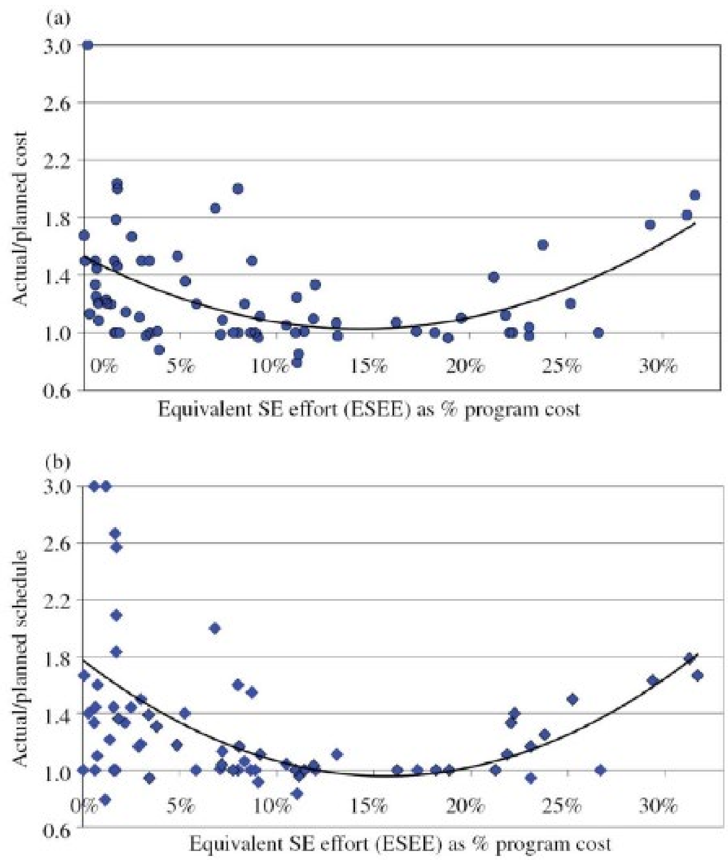


Figure 2: Cost and Schedule impacts of Systems Engineering
(Source: INCOSE)

One might think of systems engineering being analogous to buying insurance. You buy insurance to help manage your risk. Systems engineering is critical to managing project risk. The level of systems engineering you need is directly related to the level of perceived risk understood for the project.

Following a systems engineering project lifecycle process ensures that you consider and perform each stage of the process with linkages or traceability backwards and forwards with the preceding and upcoming processes. There is a cost for systems engineering just like there is a cost for insurance. The value of preventing costly mistakes (e.g., missed user needs, missed requirements, unfulfilled project functionality, etc.) is the core motivation for implementing systems engineering and is critical to the success of your project.

A successful project is characterized by being delivered within its schedule and budget, and meeting the user's needs. Systems engineering provides an overall project process that increases the likelihood of a successful project deployment. A general tenet of systems engineering is to find and fix problems/issues early in the project lifecycle so correcting them will have less impact to the schedule and budget. It is always more expensive to make changes later as the project progresses. Generally, this will reduce the project's risk factors.

2.7 Systems Engineering Life Cycle Models

2.7.1 Life Cycle Model Overview

You may not be familiar with the term "life cycle model". A life cycle model describes the distinct stages of a system's "life". Generally, a system moves through different stages: planning, concept, development, implementation, operations and support, and retirement. The role of the systems engineer encompasses the entire life cycle of the system with principal focus on development and implementation – the stages when the system is created. There are several approaches that can be used to develop a system:

Sequential Methods – Sequential methods are often called the "waterfall" approach since the work flows through sequential steps from initiation to completion. A sequential approach can be very efficient if you know exactly what you want and there is little risk of change over the course of the system development.

Incremental Methods - The most common incremental method is really a variation of a sequential method. It may have incremental aspects relating to design and development of the system, but a key aspect of this approach is that the complete system is initially planned and specified. In this case, you are making one pass through the first part of the development process to determine the needs addressed and the requirements of the system. Several projects then iterate through the latter part of the development process for each phased increment.

Iterative Methods - There are several development methods that employ iterative approaches throughout the development process. In these methods, developers plan, specify, and implement an initial system capability. Following the initial development, which may or may not be determined acceptable for operational use, this process leverages experience gained with the initial system to define the next iteration to fix problems and extend capabilities. Iterative methods have been widely adopted for software projects that use Agile development to respond effectively to change, risk, and uncertainty.

The best method depends on how much you know about the system that you want to implement, whether you have all the funds that you need to implement the system in one fell swoop, your agency and contractor capabilities, and your assessment of the risks.

2.7.2 Sequential Methods

Sequential methods are characterized by a series of defined processes with gates that are passed through between each process. These gates are usually deliverables such as a Concept of Operation or Requirements document. Traceability between the processes is another trait of a sequential method as applied to project development. This approach is frequently called the "waterfall" approach since the work flows in a single direction from initiation to completion, with limited opportunity to revise products from prior portions of a project. Such an approach works well if the vision is clear, the requirements are well understood and stable, and there is sufficient funding. The problem is that there isn't a lot of flexibility or opportunity for recovery if your vision or the requirements change substantially.

Key Sequential Methods

The two most common sequential models in the transportation domain are the traditional transportation project development process and the Vee model.

Traditional Transportation Project Development

Transportation projects are identified and funded through transportation planning and programming/budgeting phases. Funded projects are then implemented using a process similar to the traditional capital project development process shown in Figure 3, but the exact process used for ITS projects will vary with the type of project. For example, ITS projects that install only field equipment (e.g., dynamic message signs) would use a process that is very close to the traditional process shown in Figure 3. ITS projects that involve hardware and software development and integration would require additional systems engineering analyses that would be significant extensions to the traditional process.

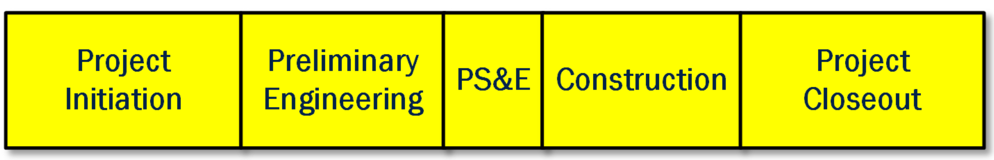


Figure 3: The Traditional Transportation Project Development Process
(Source: FHWA)

While project development processes vary from state to state and from organization to organization in each state, the transportation project development process tends to have the same major steps.

- **Project Initiation** – In this step, the project manager is identified, the project team is assembled, and the project development is planned. A high-level definition of the project is developed, costs are estimated, and the required forms and checklists are completed to garner approval for the project from the sponsoring and funding agency(ies). For FHWA and FTA, this is a critical point in the process where approval to proceed is given and federal funds are obligated.
- **Preliminary Engineering** – In the traditional capital project development process, environmental, right-of-way, and other studies are performed depending on the type of project. These studies result in better understanding of the project requirements and

constraints. ITS projects that include a construction component will require these same studies as well as additional engineering analyses to fully specify the project requirements for the ITS portion of the project. Note that from a federal aid perspective, "preliminary engineering" also includes PS&E. PS&E is split out separately here to differentiate between requirements-oriented and design-oriented steps in the traditional project development process.

- **Plans, Specifications, and Estimates (PS&E)** – The detailed design for the project, complete with detailed project specifications, estimates of material needs, and associated costs are documented. In a traditional construction project, this process step provides companies with all the information they need to develop an accurate bid. Construction elements within an ITS project will also require traditional design documentation (i.e., layout sheets, plan and elevation views, cross-section details, etc.). Design documentation is also required for the hardware and software components in an ITS project, but it takes the form of high-level design, interface specification, and detailed hardware and software specifications.

- **Construction** – The project is built. For a traditional transportation project, this is construction of the actual physical improvement. For an ITS project, this includes the procurement and implementation of the actual hardware, software, and enabling products (e.g., manuals, operating procedures, and training). This step also includes inspection of the physical improvement(s) and integration and testing of the implemented system(s).

- **Project Closeout** – After final inspection/testing, the completed project is accepted, as-built plans are created, and a project history file is completed.

Vee Model

Another important example of a sequential method is the Vee model shown in Figure 4. Since it was first developed in the 1980s, the Vee model has been refined and applied in many different industries.

As shown in the Vee, the systems engineering approach defines project requirements before technology choices are made and the system is implemented. On the left side of the Vee, the system definition progresses from a general user view of the system to a detailed specification of the system design. The system is decomposed into subsystems, and the subsystems are decomposed into components – a large system may be broken into smaller and smaller pieces through many layers of decomposition. As the system is decomposed, the requirements are also decomposed into more specific requirements that are allocated to the system components.

As development progresses, a series of documented baselines are established that support the steps that follow. For example, a consensus Concept of Operations supports system requirements development. A baseline set of system requirements then supports system design. The hardware and software are implemented at the bottom of the Vee, and the components of the system are then integrated and verified in iterative fashion on the right. Ultimately, the completed system is validated to measure how well it meets the user's needs. (Each of the steps in the Vee are defined in detail in Chapter 3.)

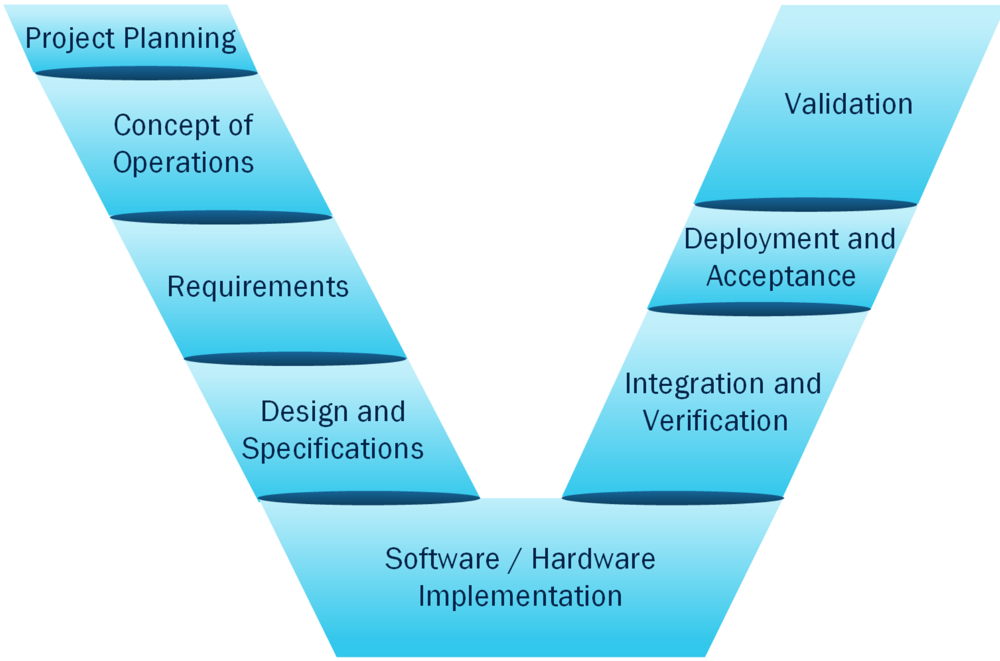


Figure 4: Sequential Method: Vee Model
(Source: FHWA)

While these two models may look quite different, you can easily relate the steps of the traditional transportation project development process to the Vee model. There is not quite a one-to-one relationship between the different stages or phases of the models, but since each are versions of sequential models, they line up fairly well.

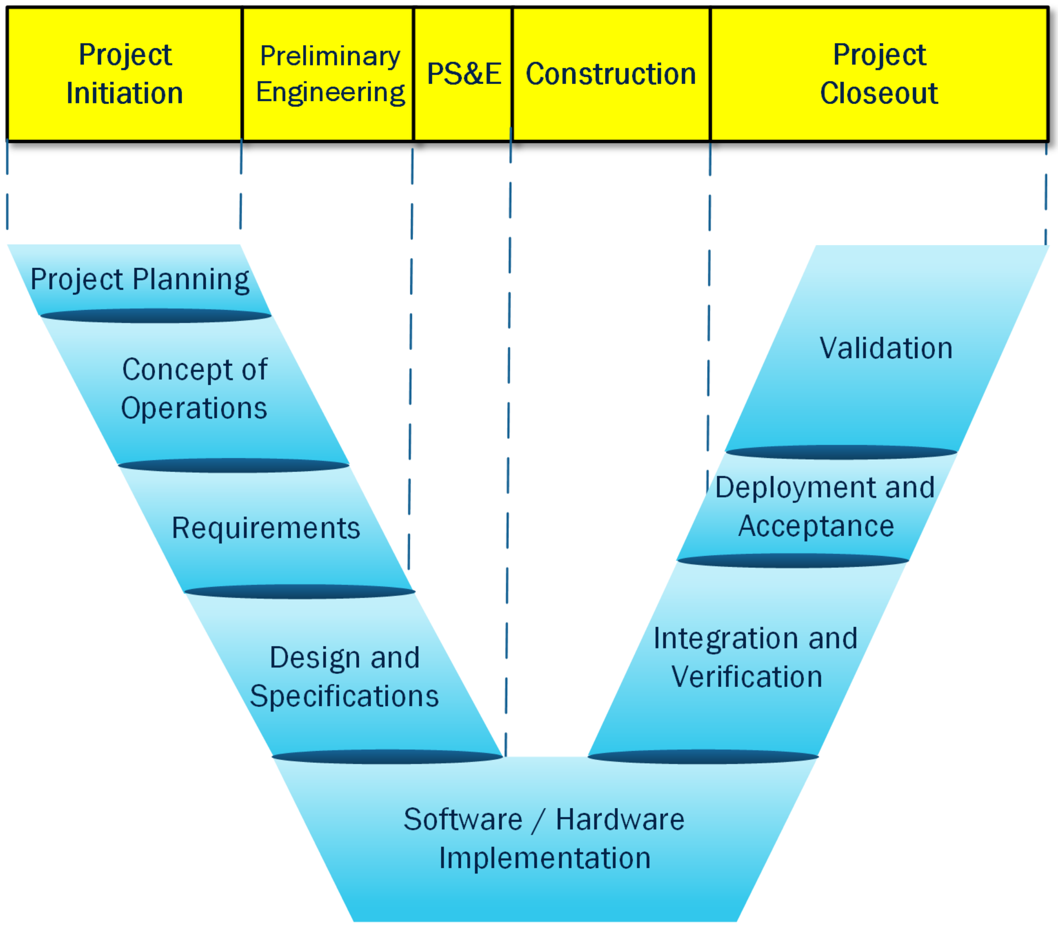


Figure 5: Comparison of Sequential Models
(Source: FHWA)

2.7.3 System vs Project Life Cycles

In the previous section, we were focused on the life of a single project from initiation to close out. As we noted in Section 2.4, there are important planning steps that occur before the project is initiated. The system life cycle also extends beyond project closure to include operations and maintenance and ultimate retirement of the system. This broader system context is reflected in "wings" that are added to the Vee model that we use for ITS, as shown in Figure 6. This figure also distinguishes between a system life cycle and a project life cycle. The project life cycle encompasses all the activities of a project, how a project is planned, designed, developed and tested and covers from inception to completion for a specific project.

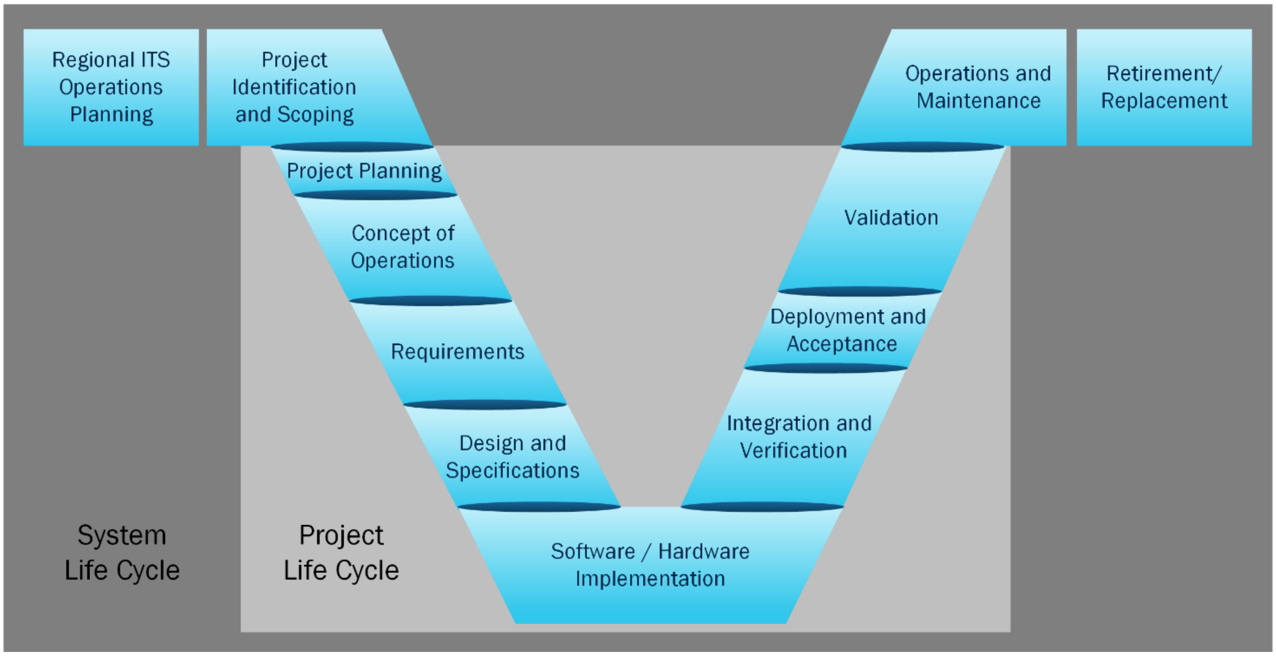


Figure 6: System vs Project Life Cycle
(Source: FHWA)

2.7.4 Incremental Methods

Most transportation systems are implemented incrementally through multiple projects, requiring multiple passes through the sequential project development process. A key aspect of the incremental method is that the complete system is initially planned and specified. In this case, you are making one pass through the first part of the development process to determine the needs addressed and the requirements of the system. One or several projects then iterate through the latter part of the development process for each phased increment. This is a common strategy for a phased deployment of a system with stable system requirements and design, such as for field equipment deployment where a core system can be deployed to provide initial capability and additional components can be incrementally implemented and deployed across a metropolitan area in several phases and several projects. An example of adding an incremental aspect to the Vee model is shown below.

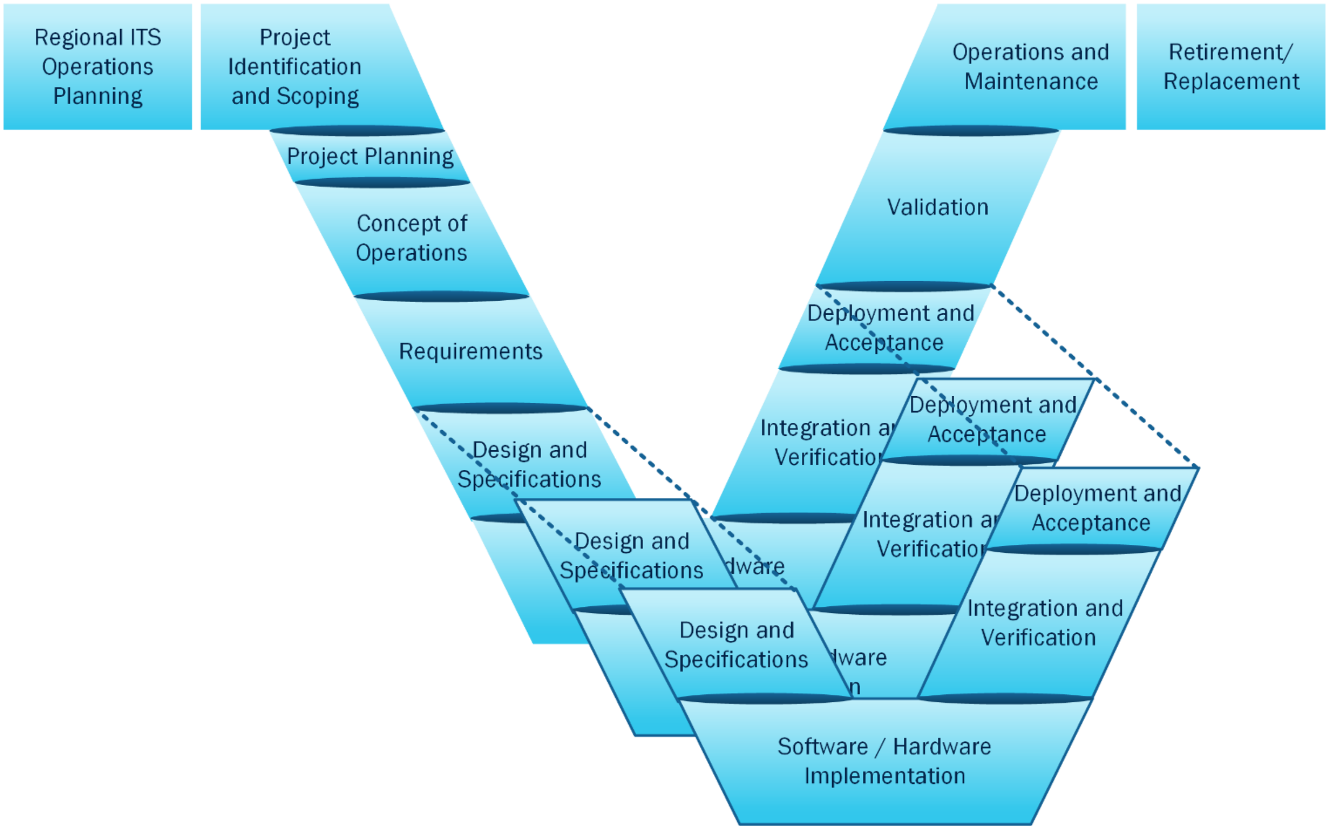


Figure 7: Systems Engineering Vee with Multiple Incremental Projects
(Source: FHWA)

2.7.5 Iterative Methods

There are several development methods that employ iterative approaches throughout the development process. In these methods, developers plan, specify, and implement an initial system capability. Following the initial development, which may or may not be determined acceptable for operational use, this process leverages experience gained with the initial system to define the next iteration to fix problems and extend capabilities. In each iteration of the development process, prior efforts can be revised and refined including the Concept of Operations, system requirements, and design, as necessary. This process is continued with successive iterative refinements until the system is complete. These iterative approaches are often used when the requirements are unclear from the beginning, or the stakeholder wishes to hold the system of interest open to the possibilities of inserting new technology or capabilities. As such, they are not widely applicable to ITS projects.

One key type of iterative method, described under the term Agile development, has been considered for use for ITS projects. The discussion below is taken from the FHWA document *Applying Scrum Methods to ITS Projects*.

Agile enterprise concepts were formulated in several commercial domains (e.g., automotive, semiconductor, telecommunications) and the military. Agile system frameworks and agile enterprise reference models were being developed in the mid to late 1990s, and eventually migrated over to the software development community.

In the early 2000's a group of software development experts developed The Manifesto for Agile Software Development" and the "12 Principles Behind the Agile Manifesto." The Agile Manifesto and the 12 supporting principles were written specifically as a philosophy for software development teams to easily adjust to stakeholder and user needs by focusing on people and interactions, not processes and tools. Using agile development allows project teams to incrementally deliver planned functionality earlier in the development cycle. Since then, Agile methods have been embraced by the DOD and NASA.

While a number of Agile methodologies have been developed over the past 20 years, one of the most common is referred to as the Scrum method. Scrum is an iterative agile methodology for managing product development "within which people can address complex adaptive problems, while productively and creatively delivering products of high value"[1] The original idea of the agile concept was to provide an alternative to the waterfall method for software development, providing an alternative to documentation driven, heavyweight software development processes. The idea was to incrementally deliver planned functionality earlier in the development cycle. Scrum focuses a team's efforts on quick and incremental delivery of the product with regular feedback from stakeholders. This framework allows product development to respond quickly to changing requirements and adapt to evolving technologies.

A key takeaway from the FHWA document is where, in the context of ITS projects, are Agile methods appropriate.

Characteristics for projects that may be better suited to using Agile (or Scrum) include:

- The client does not have a good vision of specific product (or a single unit) functions and needs to see something tangible to help them decide on said functions
- System upgrades to existing systems where the new/needed functionality is well understood by the stakeholders
- New human interfaces that require frequent user trials to perfect the interface
- Web sites that require frequent user trials to perfect functionality
- Functionality that can be delivered incrementally

Characteristics for projects that are not suited to using Agile (or Scrum) include:

- Systems or system components dealing with safety critical or safety of life features/functions
- Systems requiring long-term maintenance and/or thoroughly documented project design decisions
- Systems consisting of high levels of integrated disparate systems

There are also several challenges that should be considered before adopting Agile methods (or Scrum) to your project:

- Don't use Agile methods (or Scrum) when safety of life, long-term maintenance, and integrating large disparate systems are at risk

[1] Applying Scrum Methods to ITS Projects, Final Report — August 2017, Publication Number: FHWA-JPO-17-508

- Consider the skill set, staff knowledge, and resources required when using Agile methods (or Scrum)
- Remember Agile is new to ITS community and implementation is still evolving

Below is a short description of the Scrum methodology. For additional information about Scrum, and its use for ITS projects, refer to the FHWA document Applying Scrum Methods to ITS Projects (https://rosap.ntl.bts.gov/view/dot/32681).

Scrum prescribes a process of events and artifacts (deliverables/ by-products of Scrum development) that center around a time-boxed (time-constrained) iteration called a Sprint. Once a Sprint begins, its duration is fixed and cannot be shortened or lengthened. Typically, a Sprint will last from two to four weeks with two weeks being the most common. The basic Scrum Method is described below and illustrated in Figure 8.

- Sprint Planning- As illustrated in Figure 8, each Sprint starts with a Sprint Planning Meeting where the Scrum Team plans the work to be completed during the Sprint. In collaboration with the Product Owner, the Development Team selects a subset of the prioritized items in the overall list of features to be developed (called the Product Backlog) that they can reasonably complete in the fixed Sprint duration (usually 2 weeks). This subset of items and/or features that the Development Team commits to completing is called the Sprint Backlog.

- Daily Scrum Meeting- During the Sprint, the Development Team meets daily to discuss daily progress, planned work, and any roadblocks they have encountered.

- Sprint Review and Retrospective- Each Sprint ends with a Sprint Review Meeting where the work completed is reviewed and a demo of the working product is performed.

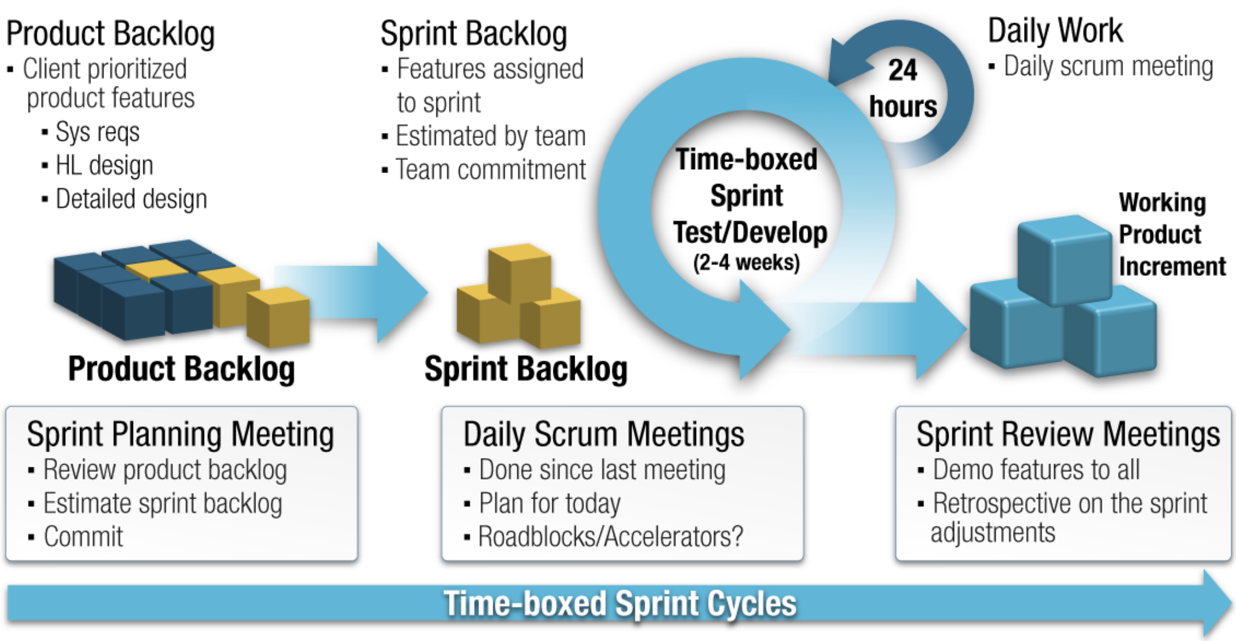


Figure 8: Illustration of Scrum Method
(Source: FHWA)

2.8 US DOT Regulations

To promote an understanding of how ITS infrastructure serves the activities agencies undertake to attain operational goals and objectives, and to promote the effectiveness of ITS projects in supporting those activities, Congress included Section 5206(e) in the Transportation Equity Act for the 21st Century. As a result of this section, in 2001 USDOT established in Title 23 of the Code of Federal Regulation a new Part 940 – Intelligent Transportation System Architecture and Standards. This section identifies the National ITS Architecture, the requirement for regions to develop a regional ITS architecture, and the requirement for agencies to undertake good, documented engineering processes that help ensure that ITS projects attain the integration and operational objectives embodied in their regional ITS architectures. In addition to CFR 940, FTA has implemented a similar policy to address requirements on transit projects.

The CFR 940 requirements on ITS projects deployed by state, county, or municipal transportation agencies that utilize money from the Highway Trust Fund are managed by FHWA Division Offices. These requirements cover the two basic topics mentioned above: 1) the development of regional ITS architectures that can be used to plan the integration of ITS deployments within a region and 2) the documentation of systems engineering processes relating to ITS projects. Many of the Division Offices have defined procedures or processes that local agencies should follow to address the requirements. They may also develop forms, such as a Systems Engineering Review Form (SERF), to support the reporting of how ITS projects address the requirements. Appendix B provides a general set of approaches for agencies to address FHWA or FTA requirements.

As you will see in the following chapters, the systems engineering approach applied in the broader industry encompasses more than the requirements identified in the Regulation/Policy. A complete systems engineering process will meet or exceed the specific systems engineering analysis requirements identified in the Regulation/Policy. Recently the USDOT has put out a memorandum to clarify the use of systems engineering for ITS projects (Information Memo - Systems Engineering for ITS Projects (dot.gov)). Relevant information from this memorandum is provided below.

Systems engineering provides a needs-focused, requirements-driven engineering process for ensuring that projects involving technologies used for intelligent transportation management meet the needs and the expectations of the agencies undertaking them. It is also used to demonstrate that projects are consistent with any applicable regional ITS architecture to ensure that they maintain interoperability in accordance with the stated needs and objectives of their stakeholder agencies.

Under 23 CFR 940.13, Division Offices (or States that have signed a Stewardship and Oversight Agreement) provide oversight of ITS projects. Oversight of ITS projects includes ensuring that compliance with the requirements in 23 CFR 940.11 is demonstrated. Projects that do not fund the acquisition of technologies that provide or contribute to the provision of ITS user services do not fall within the definition of an ITS project in 23 CFR 940.3 and, therefore, are not subject to these requirements. Examples of such projects include construction of traffic signals that are not expected to be part of a coordinated system of traffic signals, upgrades to existing signals, signal timing and other operational studies, operations and ITS feasibility and planning studies, and routine operations.

While systems engineering comprises a broad spectrum of proven engineering practices that take a variety of forms, 23 CFR 940.11 requires only the provision of a Systems Engineering Analysis that includes the seven attributes outlined in 23 CFR 940.11(c):

(1) Identification of portions of the regional ITS architecture being implemented (or if a regional ITS architecture does not exist, the applicable portions of the National ITS Architecture);

(2) Identification of participating agencies roles and responsibilities;

(3) Requirements definitions;

(4) Analysis of alternative system configurations and technology options to meet requirements;

(5) Procurement options;

(6) Identification of applicable ITS standards and testing procedures; and

(7) Procedures and resources necessary for operations and management of the system.

The analysis should be proportional to the scope and complexity of the project.

A regional ITS architecture, which is discussed in 23 CFR 940.9, can be described as a database of ITS technologies and systems, the interfaces and data that flow between them, the roles and responsibilities of their operating agencies, the requirements those systems will fulfill, and the goals and objectives that those systems help regional stakeholders to attain. Importantly, the regional ITS architecture is an outgrowth of an operations planning process, as identified in 23 CFR 940.9(a):

> *regional ITS architecture shall be developed to guide the development of ITS projects and programs and be consistent with ITS strategies and projects contained in applicable transportation plans.*

The Systems Engineering Analysis, therefore, connects an implementation project to a program of operational improvements that deploy ITS over the horizon of applicable transportation plans. Systems engineering stands on a foundation of operations planning.

Most ITS projects that deploy conventional ITS technologies in support of existing operational activities pose lower risk than projects deploying innovative technologies in support of operational activities new to the agency undertaking them. A low-risk project has the following characteristics:

- Experienced users who understand the application of the technology being implemented.
- Clear and sufficiently detailed documentation of the agency’s activities, needs, and requirements that are applicable to the proposed work.
- No new agencies, jurisdictions, or modes that may require additional documentation of their activities and needs beyond what is already identified in existing documentation.
- No new software being developed.
- No new interfaces between systems not previously in use.
- Use of technologies already shown to fulfill documented requirements.
- Use of technology products that are not at the end of their service life such that their use would shorten the project lifecycle.

For low-risk projects, the required attributes of a Systems Engineering Analysis provided in 23 CFR 940.11 can be addressed using existing, reused, or pre-drafted documentation. For these low-risk projects, State and local transportation agencies have demonstrated several different approaches for conducting systems engineering at an appropriate level of detail to minimize risk without unnecessarily developing systems engineering products from scratch, and without the need to hire a systems engineering consultant. These approaches can include:

1) Categorical Systems Engineering. For example, Minnesota Department of Transportation (DOT) devised a risk-based approach that provides pre-drafted systems engineering documents for common lower-risk categories of projects. These pre-drafted documents can be used as supporting documentation that provides complete and correct descriptions of needs and requirements for the projects within the category. (See https://www.dot.state.mn.us/its/systemsengineering.html). Ohio DOT similarly identified specific types of projects that would be given defined systems engineering treatment. (See the Ohio DOT Traffic Engineering Manual, Part 13 Intelligent Transportation Systems (section 1300 (ITS), subsection 1301-3.2)): https://www.dot.state.oh.us/Divisions/Engineering/Roadway/DesignStandards/traffic/TEM/Documents/Part_13_Complete_011714Revision_011614_bookmarked.pdf).

2) Extensions of existing systems, for which systems engineering documents already exist. For example, California Department of Transportation (Caltrans) assesses risk for projects in this category based on the existence of the elements of a Systems Engineering Analysis already in place that can be used for the project in question. (See the Caltrans Local Assistance Program Guidelines, Chapter 13 Intelligent Transportation Systems (ITS) Program (pages 1 – 19): https://dot.ca.gov/-/media/dot-media/programs/local-assistance/documents/lapg/g13.pdf).

3) Extracting Systems Engineering Analysis elements from current and appropriately updated regional ITS architectures. A well-maintained regional ITS architecture developed in accordance with 23 CFR 940.9 includes roles, objectives, and requirements at the planning level. A project architecture includes these elements extracted from the regional architecture. These elements can be used to support a Systems Engineering Analysis.

4) The use of Model Systems Engineering Documents. A number of State and local agencies successfully employed the Model Systems Engineering Documents for Adaptive Control Signal Technologies developed by FHWA as part of Every Day Counts (Round 1). The success of these applications led to the expansion of the library of Model Systems Engineering Documents to include the following:

a) Traffic Signal Systems (incorporates and supersedes Adaptive Control Signal Technologies): https://ops.fhwa.dot.gov/publications/fhwahop19019/index.htm
b) Dynamic Message Signs: https://ops.fhwa.dot.gov/publications/fhwahop18080/index.htm
c) Closed-Circuit Television: https://ops.fhwa.dot.gov/publications/fhwahop18060/index.htm
d) Transportation Sensor and Detection Systems: (forthcoming)

Projects that use more than one of these technologies may combine the needs and requirements extracted from these model documents.

Some ITS projects, such as projects that lack a clear and documented understanding of the requirements they must fulfill, or the needs they must satisfy, demonstrate higher risk. In some cases, these projects may involve technologies approaching obsolescence or technologies not previously tested. In addition, projects that will develop new software or interfaces not previously in use pose greater risk. The degree of agency familiarity with a technology or operational activity also influences risk. For higher-risk projects, implementing agencies should consider the development of new systems engineering documentation as needed to provide the attributes required to support a Systems Engineering Analysis per 23 CFR 940.11.

Many States use a Systems Engineering Review Form (or similar document) to assess the risk of the project as the basis for identifying the systems engineering steps appropriate to mitigate those risks (for example, see the New Jersey DOT TSM Procedures Manual at page 12: https://www.state.nj.us/transportation/eng/elec/ITS/pdf/TSMProceduresManual.pdf).

If that review reveals gaps in the required systems engineering documentation, a plan for filling those gaps may be embodied in a Systems Engineering Management Plan (SEMP) or similar document (see Caltrans Local Assistance Program Guide at Chapter 13 page 2: https://dot.ca.gov/-/media/dot-media/programs/local-assistance/documents/lapg/g13.pdf).

3 The Systems Engineering Process

This chapter follows an ITS system from initial operations planning all the way through retirement of the implemented system. The systems engineering approach is described in chronological order as it steps through topics like Concept of Operations, Requirements, Design and Verification. The chapter also includes a discussion of key cross-cutting processes like Project Management and Risk Management that are not confined to a single step in the system life cycle.

3.1 Key Process Topics

Systems engineering is a process used to improve the outcomes of project development. There are a variety of ways that the systems engineering process can be implemented depending on the scope of the project. Whichever process is used for project development, this process occurs within a larger context in which transportation projects are planned within a region, wherein the operational strategies articulated by an agency become fully and efficiently supported by the technology they build. Stated another way, systems engineering helps agencies ensure that their operational funds are spent in a way that most directly serves the attainment of their operational goals and objectives. A more detailed discussion of the context in which project development using systems engineering occurs is given in Section 2.4.

Process and Life Cycle Models

Agencies implement projects using a development process. This development process can be considered part of a larger life cycle model, which begins with the planning of the system and ends with its retirement or replacement. There are several life cycle approaches used by agencies to develop a technology project. The best development strategy depends on how much you know about the system that you want to implement, whether you have all the funds that you need to implement the system in one fell swoop, your agency and contractor capabilities, and your assessment of the project risks. A discussion of the different development processes and life cycle models relevant to ITS projects is found in Section 2.7.

Process Steps

The steps in the systems engineering process are explained using the Vee diagram approach described in the Life Cycle Models discussion. The process is described as a set of steps along the Vee diagram showing the progression of SE processes from regional transportation planning to a deployed system ready to be retired or replaced. A short overview discussion of the step can be found in Section 3.3.1. Discussion of each individual process step can be found in the subsections of Section 3.3. In addition, a set of cross cutting activities are described in Section 3.4.

Process Tailoring

The process used for the development of a project should be tailored to fit the nature of the project. Issues such as the risk or complexity of the project affect how the systems engineering process would be applied to a specific project. Process tailoring is particularly important for ITS projects because so many of these projects are smaller, less complex, less risky projects like signal system upgrades. Even for small projects, you still should have documented requirements, design, and verification procedures. Tailoring allows you to adjust the amount of formal documentation and reviews and to focus the process on those steps that are most critical to your project's success. Ultimately, you want to define a process that will address the project's risks, no more and no less, so a preliminary risk analysis is a good way to determine how much process is appropriate. An introduction to the agency perspective relating to systems engineering is found in Section 4.1. The first step in tailoring is to evaluate the risk inherent to the project. A discussion of risk evaluation for a project is found in Section 4.2. Once the determination of risk is made then the tailoring of the SE Process can be performed. A discussion of the tailoring of the SE Process is found in Section 4.3. A discussion of the how agencies can address FHWA regulation (and FTA Policy) regarding the application of systems engineering is found in Section 4.5. Finally, the USDOT has recently provided additional information about applying the SE Process to lower risk projects which can be found in Section 2.8.

ITS Procurement

ITS projects are developed as the result of procurement of the systems and services. For technology projects there are certain procurement strategies and models that are relevant based upon the scope and complexity of the project. A discussion of issues relating to ITS procurement is found in Section 4.4.

3.2 High level SE Process Overview

As shown here, the systems engineering process can be shown using the letter "V". Let's start at the upper left of this simplified Vee diagram. The first major process step is capturing the Needs of the stakeholders. These needs provide the basis for forming the requirements in the next step. The requirements inform the bottom of the Vee Diagram depicting Design and Implementation. This step is where the system gets built. Once the system is built, we travel up the right side of the Vee diagram where we test the system. The last step on this diagram is Validation where we make sure that the system is useful to the stakeholders.

Time increases from left to right as depicted by the solid arrows on the outside of the Vee diagram. The dashed arrows in the middle of the Vee indicate that the Testing is against the requirements and the Validation is ensuring the needs are being met by the final system.

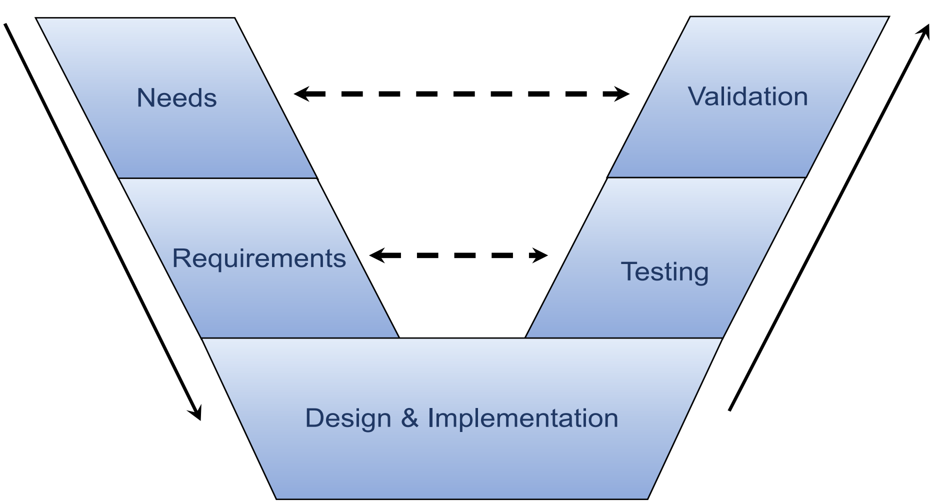


Figure 9: Overview of Systems Engineering Process
(Source: FHWA)

3.3 SE Process Steps

This section introduces the detailed steps in the Vee diagram and then walks through each step in sequence, describing the input, process, and output for each step in turn.

3.3.1 Vee Overview

The SE process steps are based on the detailed steps of the Vee diagram.

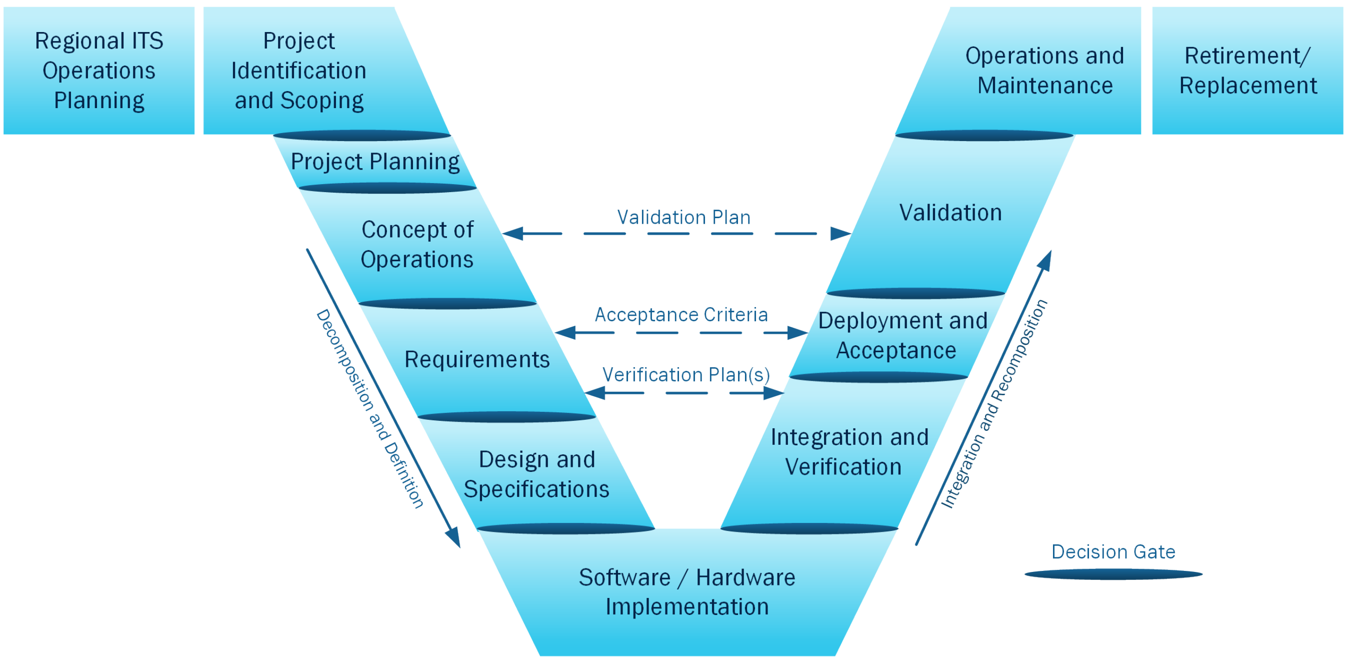


Figure 10:Vee Process Steps
(Source: FHWA)

The following discussion provides a quick stroll through the steps in the diagram. Later in the chapter each step will be expanded upon.

Before ITS project deployment can begin, **regional ITS operations planning** occurs within the context of regional (or statewide) transportation planning. Two of the key outputs of the step are a Transportation System Management and Operations (TSMO) Plan and a Regional ITS Architecture.

Project identification and scoping covers studies for major deployments in a region, sometimes called feasibility study or concept exploration as well as the programming activity that identifies the projects that will receive funding in a region (or state).

Moving down the left side of the Vee, an ITS project is initiated and its progress is monitored with **Decision Gates** that represent milestones where project stakeholders and management determine that the results of the preceding process step are adequate and the project is ready to move on to the next step. The first Decision Gate represents the decision to proceed with the project. Like the process steps, the Decision Gates are also tailored based on the complexity of the project. Small projects may have informal documentation and reviews while larger and more complex projects will have formal documentation submittals and formal reviews and approvals to proceed.

For each ITS project under development, the **project planning** step begins the project development process by identifying the tasks, resources, schedule and risks of the project, and documenting them in a project management plan.

Concept of operations, which documents the user needs and corresponding objectives to be satisfied by the project, as well as the operational and support environments of the project is the first key SE output, which is then used as input to a complete system requirements analysis.

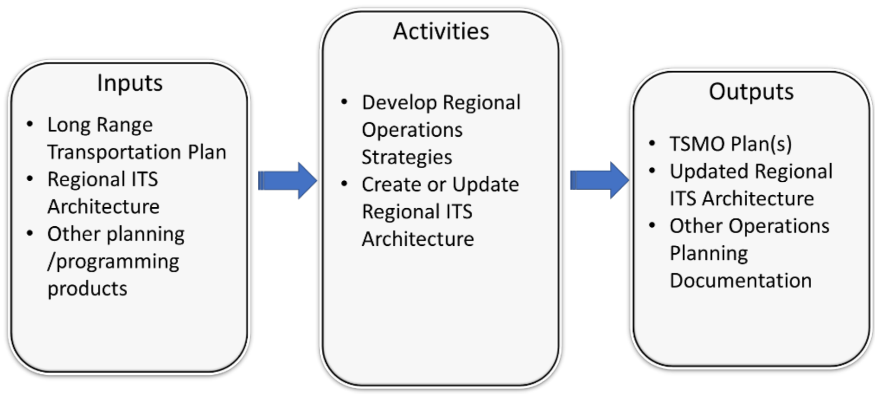
The **system requirements** state the functional, performance, and environmental requirements on the ITS project and are for the most part technology neutral.

Once the project ITS system requirements (functional, performance, and environmental) are defined, they can be used to select the technologies to be used in the ITS. The selection of specific implementation including the technologies for the project is the output of the **design and specifications** step.

The left side of the “Vee” is sometimes called the “decomposition” side, because it starts with a very high-level view of the ITS in the regional setting, then gets more and more detailed as each high level need is decomposed into system requirements, and then specific technology specifications resulting in a complete detailed design for the ITS. The base of the Vee, **Software and Hardware Implementation**, is the actual build or procurement of hardware and coding or procurement of software for the ITS, and then the right side of the Vee begins, sometimes called the “recomposition” side, where the individual hardware elements and software modules of the ITS are **integrated** and combined to result in the complete ITS system, and **verified** that all requirements have been satisfied (in the **integration and system verification** step).

After **deployment and acceptance**, the system can be **validated** that it meets all the needs/objectives originally intended. During **operations and maintenance**, the system may be updated as needs, requirements, and technology evolve. Eventually, this evolution either removes the need for the system or cannot be accommodated by the system and the system is **retired or replaced.**

3.3.2 Regional ITS Operations Planning

OBJECTIVES	Place ITS operations into overall regional transportation planning.
DESCRIPTION	Planning processes are used to identify projects whose implementation will respond to regional needs. This step describes some of the key planning/ programming activities relevant to ITS deployments in a region. These activities include development of operations plans such as a Transportation Systems Management and Operations (TSMO) plan, development of a regional ITS architecture, which provides a plan for the integration of ITS systems in the region.
CONTEXT	
INPUT *Sources of Information*	• Long Range Transportation Plan • Regional ITS Architecture • Other planning /programming products
PROCESS *Key Activities*	• Develop Regional Operations Strategies • Create or Update Regional ITS Architecture
OUTPUT *Process Results*	• TSMO Plan(s) • Updated Regional ITS Architecture • Other Operations Planning Documentation

Overview

ITS Projects are developed within the larger context of regional transportation planning. The goal of the regional transportation planning process is to make quality, informed decisions pertaining to the investment of public funds for regional transportation systems and services. Two of the key outputs of the transportation planning process are the long-range metropolitan transportation plan (MTP) and the Transportation Improvement Program (TIP). Each metropolitan planning organization (MPO) must prepare an MTP, in accordance with 49 USC 5303(i), to accomplish the objectives outlined by the MPO, the state, and the public transportation providers with respect to the development of the metropolitan area's transportation network. The MTP has a horizon of 20+ years and is fiscally constrained. The TIP is the mechanism that assigns federal funding to a prioritized list of specific projects to be constructed over a several-year period (usually 5 years) after the program's approval. The TIP is considered the near-term "project implementation" mechanism of the MTP.

There are additional planning efforts that occur between the development of the MTP and the programming of transportation projects in the TIP. One key effort is the Transportation System Management and Operation (TSMO), which is defined by CFR 23 USC 101 (a) (30) as "an integrated set of strategies to optimize the performance of existing infrastructure through the implementation of multimodal and intermodal, cross-jurisdictional systems, services, and projects designed to preserve capacity and improve security, safety, and reliability of the transportation system." The TSMO plan identifies the things that an agency will do operationally and the staff, facilities, equipment and infrastructure needed to do those things. These elements constitute an operations plan.

A second output is a Regional ITS Architecture, which is defined by 23 CFR 940 as "a regional framework for ensuring institutional agreement and technical integration for the implementation of ITS projects or groups of projects". According to 23 CFR 940.9 "a regional ITS architecture shall be developed to guide the development of ITS projects and programs and be consistent with ITS strategies and projects contained in applicable transportation plans". This effort defines the planned ITS deployments for a region with a 10+ year timeframe, and is not fiscally constrained, meaning that it represents an aspirational view of ITS services and projects in a region.

Risks to be managed

Some of the risks addressed at this step are:

- Institutional integration issues. Anticipate future operational needs and technology deployments (ITS) that require institutional integration and plan for deployments to address operational needs. Planning for these deployments can reduce the risk associated with coordinating effort between institutions by addressing possible integration issues.   This risk is technical and institutional in nature.
- Technical integration issues. Agree on how the information to be shared between stakeholder ITS elements and other stakeholder ITS elements (or systems outside of the ITS) and begin consideration of open standards or documented protocols that will reduce the risks in future projects. This is a technical risk.
- Incorrect customization. Tailoring services to regional needs can lead to significant benefits in scoping, estimation, costing etc. A mistake in tailoring however, can lead to significant challenges later in the process for the individual ITS projects involved, so it is crucial that tailoring be correct if it is relied upon for individual ITS project cost and schedule estimates. This is a technical risk.

Activities

Develop Regional Operations Strategies

A TSMO Plan can be a valuable resource to connect the strategies of the MTP to transportation operations in a region. Creation of a TSMO plan is not a requirement for a region like the MTP, but regions throughout the nation have embraced the need to plan operation from a regional perspective, rather than as silos for each individual organization. A TSMO Plan defines a set of regional strategies across three key elements:

Figure 11: TSMO Planning Elements
(Source: FHWA)

1. **Strategic elements**: Strategic thinking is a foundation for developing a TSMO program. It involves clearly defining the relationship of TSMO to the agency mission or regional vision. The strategic aspect of TSMO program planning provides answers to questions of “why” TSMO is important, and a high-level vision of “what” the agency seeks to achieve, along with strategic goals and objectives.
2. **Programmatic elements**: The programmatic elements of TSMO program planning addresses issues surrounding organizational structure and business processes for implementing TSMO activities. This level of planning addresses “how” the program operates, resource and workforce needs, and internal and external coordination and collaboration. It identifies responsibilities of organizational units for specific TSMO services, projects, and activities, as well as use of analysis tools to guide investment decision-making.

3. **Tactical elements**: The tactical elements are derived from the broad institutional and organizational issues to address specific services, programs, and priorities.

Detailed description of TSMO planning, including rationale for developing a plan, the particulars of what goes into a plan, and references to existing plans can be found at https://ops.fhwa.dot.gov/tsmo/index.htm.

Create or Update the Regional ITS Architecture

A regional ITS architecture identifies the regional ITS services (existing and planned) along with the necessary institutional integration that the regional stakeholders believe will meet their operational needs for ITS investments. The regional ITS architecture identifies the tailored ITS services for a region, and the necessary stakeholder ITS elements and their institutional information sharing dependencies for each tailored ITS service – which is one way of representing the operations plan for a region. The goal of the regional ITS architecture is to document the technological breadth, scope and integration of ITS in a region over the life of those systems. Each ITS service is illustrated by identifying the stakeholder elements and their input and output information flows necessary to implement the ITS service (called an ITS service package, tailored for a specific region – and there may be more than one instance of each tailored service package in a region). Finally, the regional ITS architecture includes the identification of communications solutions that include published or "open" standards (or local protocols) that the stakeholders agree will be used to facilitate the implementation of those information flows, especially for those information flows that cross institutional boundaries.

A regional ITS architecture identifies the existing ITS capabilities in a region, and the ITS projects that will be deployed over time in the region. Each ITS project is made up of one or more regional tailored ITS services identified in the regional ITS architecture. In this way, the regional ITS architecture shows what is already deployed in a region, and the sequence (or timeframes) of the ITS projects that could be deployed (if/when resources for the projects are available) to build out the regional ITS architecture.

If the regional ITS architecture is to be used effectively, it must be a consensus product. That means that all the stakeholders who will be developing, operating, using, and maintaining ITS projects agree with the regional ITS architecture, and agree to use it as the guide for projects to be developed. Consensus in this context means that the developers, users, operators and maintainers (i.e. all the stakeholders) also agree to follow the regional ITS architecture in their ITS project development and deployment. If there is not this consensus in a published regional ITS architecture, then the utility of the regional ITS architecture is severely diminished, and while the regional ITS architecture might be a starting point for a project's institutional integration, the project developers will need to now invest schedule and budget to build the consensus for the project's integration with other projects in the region – which misses the point of having invested in developing (and maintaining) a regional ITS architecture in the first place.

TIP

If you're developing a regional ITS architecture for your region, and there are one or more ITS services and/or interfaces for which you can't get consensus from the stakeholders that should be using those services and/or interfaces, then do not include those services and/or interfaces in your regional ITS architecture.

Additional information about creating and updating Regional ITS Architectures can be found in the Architecture Use pages on the ARC-IT website (www.arc-it.net).

Tailoring this Step

The process might require a step to show compliance with the FHWA regulation or FTA policy. Some regions have established specific guidance for architecture use. For example, in California, the Caltrans Local Assistance Procedures Manual includes a Systems Engineering Requirements Form (SERF) that includes a requirement to map the project to the regional ITS architecture. This form must be completed by local agencies at project initiation. Other regions (e.g., Florida) and agencies (e.g., Virginia DOT) have established similar guidance.

The level of activity involved in using the architecture depends on the scope of the project (i.e., how many systems and interfaces it affects) and the quality of the regional ITS architecture. Use of the architecture will lead to greater savings in later work throughout the project by utilizing the high-level definitions included in the regional ITS architectures. The Use in Project Development section of the ARC-IT Website (https://www.arc-it.net/html/raguide/raguide-c5.3.html#_Use_in_Project) provides additional guidance for architecture use in project implementation.

Policy or standard for Process Step

Regulation 23 CFR 940 and FTA Policy on Architecture and Standards require a regional ITS architecture for any region currently implementing or planning ITS projects. All ITS projects must adhere to this regional ITS architecture. The Regulation/Policy also requires that the development of a regional ITS architecture be consistent with the statewide and metropolitan planning process.

Traceable Content

Some of the artifacts developed by the Regional ITS Operations Planning process are identified in Table 1. This table illustrates how the outputs of this step form the bridge between the MTP, TIP and the initial steps of project development.

Table 1: Regional ITS Operations Planning Traceability

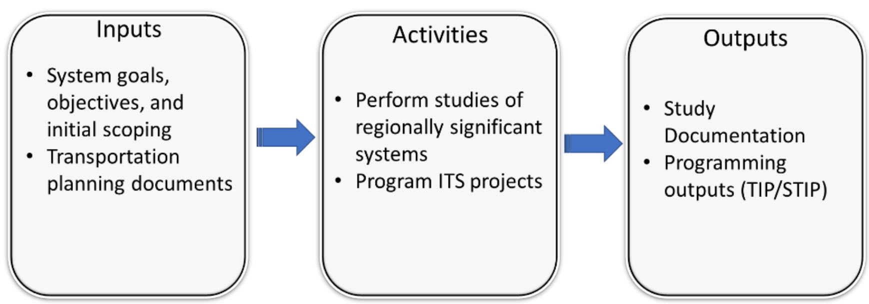
Traceable Artifacts	Backward Traceability To:	Forward Traceability To:
Regional ITS Architecture	Long Range Transportation Plan or Metropolitan Transportation Plan (MTP)	• Operations Planning System Study to select alternative concepts (if needed) • Transportation Improvement Program (TIP)
TSMO Plan	Long Range Transportation Plan or Metropolitan Transportation Plan (MTP)	• Regional ITS Architecture • TIP

Checklist

The following checklist can help answer the question "Are all the bases covered?" once the activities above have been planned or performed. **The checklist is not sufficient by itself.**

- ☑ Do the relevant regional ITS architectures need to be updated (based on their maintenance plans?
- ☑ Have any needed architecture changes been reported to the architecture maintainer?
- ☑ Have any Operations Planning documents been created?

3.3.3 Project Identification and Scoping

OBJECTIVES	• Explore alternatives for major regional efforts prior to beginning actual project development. • Program projects for deployment
DESCRIPTION	Project Identification and Scoping covers two key activities. The first, related to scoping of projects, is performing studies, sometimes referred to as feasibility studies, that consider alternatives for the development of major regional systems. A business case is made for the system and technical feasibility is assessed, benefits and costs are estimated, and key system risks are identified. Alternative concepts for meeting the system's purpose and need are explored and the superior concept is selected and justified using trade study techniques. The second key activity is the programming of transportation projects which identifies those transportation projects that will be funded in the short term in a region or a state.
CONTEXT	
INPUT *Sources of Information*	• Goals, objectives, and initial scoping of regionally significant Systems. • Transportation planning documents such as a metropolitan transportation plan, TSMO plan, or regional ITS architecture
PROCESS *Key Activities*	• Perform studies of regionally significant systems. • Program ITS projects as a part of overall regional (or statewide) transportation programming efforts.
OUTPUT *Step Results*	• Study documentation that identifies alternative concepts and makes the business case for the project and the selected concept • Programming documentation (TIP or STIP)

Overview

This step addresses two key activities that occur between operations planning described in the past step and the steps for the development of a single project. For major deployments in a region, project scoping studies, either as a **feasibility study** or **concept exploration,** may be needed in order to decide the type of projects to be developed. These efforts analyze alternatives for a system so that the alternative can be selected that best meets the needs and/or requirements over the full system life cycle at the lowest "cost". Considerations in this analysis can involve technology impacts, economic impacts and/or policy impacts.

The second key activity of this step is the programming of ITS projects. Regional programming and agency capital planning (a.k.a. budgeting) involve identifying and prioritizing transportation projects, including ITS projects, resulting in funded projects. The Transportation Improvement Program (TIP) is a phased, multiyear, intermodal program of transportation projects that is consistent with the Metropolitan Transportation Plan (MTP). Like the MTP, the TIP is the financially constrained mechanism that assigns federal funding to a prioritized list of specific projects to be constructed over a several-year period (federal regulations require a 4 year minimum for the TIP) after the program's approval. The TIP is considered the near-term "project implementation" mechanism of the MTP. A TIP that covers a state, rather than a region, is called a Statewide Transportation Improvement Program (STIP).

Risks to be Managed

Risk of picking the wrong system approach. Systems may be deployed in different ways to satisfy the same objectives. But which way is best? A wrong approach to the deployment of a system could result in higher deployment costs, life cycle costs, schedule impacts or other unintended economic or policy consequences. Depending on the type of system, this risk could be technical, institutional, scheduling, or funding related.

Risk of limited funds not addressing regional priorities. Programming defines what transportation projects will be funded in the near term. Because funding is limited, regions need to ensure that the highest priority projects are funded. Programming uses a prioritization process to identify the set of projects to be funded, and without this process a region would run the risk of regional priorities not being properly addressed.

Activities

Perform studies of regionally significant systems

Project scoping studies should identify alternative architectures, and identify the "best" architecture based on analyzing the full system life cycle and perhaps the impact of the alternative approaches on other existing or future systems in the region.

The subset of regional ITS architecture used as part of the ITS planning process can be used as input to any necessary project scoping activities, such as a feasibility study, since it identifies the current and expected future deployment of ITS in a region. A feasibility study is a tradeoff study between competing architectures that analyzes the predicted costs and benefits on one or more stakeholder needs by a proposed ITS investment. In this way alternatives can be objectively considered and ranked, and then the "best" system architecture chosen for the deployment of the system

The selected system concept or approach can then be fully documented in a project Concept of Operations report (see Section 3.3.5).

The scoping study process is an opportunity to analyze in more detail the options and associated implications for an ITS system. Because technologies (e.g. data processing and data communications options) are evolving so quickly, it makes sense to defer these analyses until as late as possible in the system development process. Hence this process is distinct from the Regional ITS Operations Planning process, which might be conducted years before a specific ITS system is programmed and initiated through a specific project. Further, because of the large time lag that might separate the Regional ITS Operations Planning process and the individual ITS project development, the Regional process is generally technology neutral, and the project processes are technology specific. The scoping study process is where the suitable architecture is selected when there is a choice of competing technological approaches for a specific project.

For example, consider a system that involves using machine vision to read license plates in electronic toll collection. The machine vision algorithm receives the video image of the vehicle from a camera, and provides the most likely sequence of characters representing the vehicle license along with the state/province of the license tag or other image attributes useful in verifying the tag ID. A study might consider where to locate the machine vision algorithm. Should it be located in a conventional computer located at the customer service center supporting multiple lane cameras, or should it be located in the individual cameras themselves (i.e. “edge computing” technology). Both approaches have unique advantages and disadvantages (how much computing power you can affordably fit in a camera vs how much data you need to send from the camera to the Customer Service Center). Conducting an objective trade study might be used to select the technical approach that meets all the objectives at the lowest cost including consideration of risks and costs associated with each approach.

Other trade studies might consider different approaches in the context of relevant policy considerations (e.g. availability of right-of-way, availability of communications bandwidth, and performance capabilities to support future project functional requirements). For example, the service Transit Signal Priority (TSP) can be implemented in one of two ways:

1. Vehicle-to-Signal (V2S) Communication Method: Buses communicate directly with signal controllers over a short-range communications channel (could be an optical, infra-red, or a line-of-sight radio channel) when they approach an intersection to request a signal priority cycle for the turning movement the bus is planning. This method has been used for many decades, but requires special hardware in each bus and special hardware in each controller for each intersection to communicate directly with the busses.

2. Center-To-Center (C2C) Method: The transit management center is tracking the bus using an AVL (Automated Vehicle Location) service, and the transit management center notifies the traffic management center (using C2C communications) when a bus is approaching that is behind schedule. The traffic management center then can communicate with its signal to select the appropriate priority cycle for the bus’s expected turning movement.

The C2C method may have the advantage of requiring very little additional investment in field and vehicle hardware, IF a transit AVL system is already deployed AND if all the signals in the transit agency’s jurisdiction are already under central (e.g. “closed-loop”) signal control by the traffic management center. Also, the AVL and centrally managed signal control systems should be very reliable.

The V2S method might have the advantage if this TSP project is an extension of an existing V2S TSP system (for example, if the transit agency is adding a new Bus Rapid Transit line (BRT) to a system that already has three BRT lines using V2S based TSP. In this case there is a large legacy investment already in place using DSRC based TSP, and only a small number of intersections need to be additionally provisioned with the V2S TSP technology. Other benefits to consider are that the agencies involved may already know how to operate and maintain the legacy V2S based TSP technology.

In this example, the Operations Planning System Study would consider all the issues associated with the TSP options, and select the option that gives the objective benefits to the agency at the lowest cost.

Project scoping studies are examples of Trade Studies. Section 3.4.5 provides additional information about the steps involved in these studies.

When a feasibility study is conducted, the feasibility study report should document:

- The alternatives considered.
- The criteria used to score the alternatives
- Any assumptions made about costs, other project investments, the environment, policies, etc.
- The ranking of the alternatives based on the results of the criteria analysis
- Identification of the selected alternative.

Programming ITS Projects

Programming is the identification of transportation projects that are funded in the near term and is the key planning activity that assigns funding to projects and authorizes these projects to commence, based on the year by year allocation of funding. The TIP (or STIP) provides a listing of all the transportation projects that are approved in a region (or a state) that use funds either from the federal government or from the Metropolitan Planning Organization (MPO). Some areas of transportation are funded through capital plans (e.g. tolling systems or public safety). While their programming happens outside of the TIP/STIP process, a capital planning process at these agencies accomplishes a similar outcome of defining projects that are funded for development.

Each MPO/ state agency has its own process for the development of the TIP/STIP, which is guided by the FHWA regulations for Metropolitan Transportation Planning and Programming, 23 CFR 450 subpart C. Section 450.326 defines *the Development and content of the transportation improvement program (TIP:) The MPO, in cooperation with the State(s) and any affected public transportation operator(s), shall develop a TIP for the metropolitan planning area. The TIP shall reflect the investment priorities established in the current metropolitan transportation plan and shall cover a period of no less than 4 years, be updated at least every 4 years, and be approved by the MPO and the Governor. However, if the TIP covers more than 4 years, the FHWA and the FTA will consider the projects in the additional years as informational.*

ITS projects (those that are uniquely ITS and those that have some aspects which are ITS) are programmed along with non-ITS projects as a part of each region’s programming process. A key aspect of the programming process is the prioritization of projects for funding. Each region defines the specifics of this prioritization process. In some regions, there are aspects of ITS projects (e.g. how they relate to the regional ITS architecture, how they address regional objectives) that impact the prioritization of the projects relative to other capital projects. A suggestion is to review the regional (or statewide) prioritization process to understand how to define (or improve) the prioritization of ITS related projects.

Tailoring this Step

For project scoping studies, the level of the activity should be appropriately scaled to the complexity of the system being studied. On one hand, for small projects that have widely known capabilities [e.g., signal systems, CMS, and CCTV], a qualitative comparison with a limited number of alternatives might be appropriate.

If the operational system will be significantly different from the one it replaces or it depends the following:

- Significant operational changes
- increased inter-agency coordination
- a new set of unique needs

In these types of projects, alternatives analysis may need to be explored in more detail.

This activity may also be dictated by state or regional reporting requirements. For example, a Feasibility Study Report (FSR) must be approved by the State of California for ITS projects with IT components.

Regarding the programming activities, as mentioned above, a key aspect of tailoring is the requirements of the region (or state) for defining projects and for prioritizing projects relevant to their ITS components.

Policy or standard for process step

The FHWA Regulation requires identifying the portion of the regional ITS architecture being implemented, identifying participating agencies, defining requirements, and analyzing alternatives.

Some states have documented requirements specifically for IT projects. In California, SAM 4819.35 [6/03] requires an FSR for all state IT projects except those with low costs or for acquiring microcomputer commodities.

Traceable Content

Some of the artifacts determined by the Operations Planning System Study process are identified in **Table 2** with their backward and forward traceability. Forward traceability to project needs and requirements will be discussed in the next two sections.

Table 2: Project Identification and Scoping Traceability

Traceable Artifacts	Backward Traceability To:	Forward Traceability To:
System alternative selected. TIP/STIP	Operational planning outputs	Project Planning

Checklist

The following checklist can help answer the question “Are all the bases covered?” once the activities above have been planned or performed. **The checklist is not sufficient by itself.**

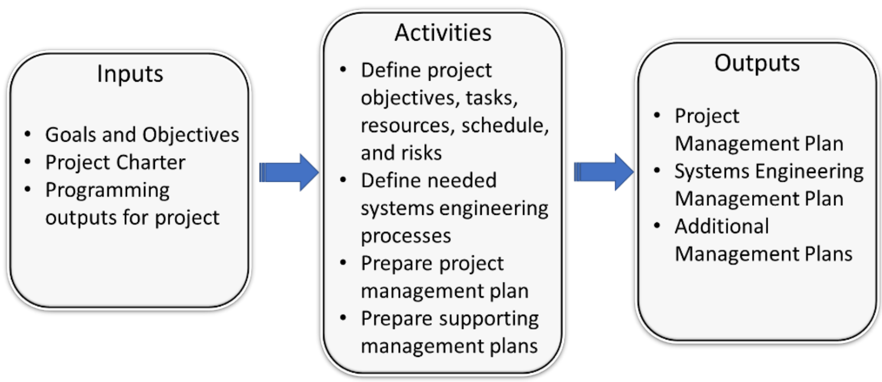
For project scoping studies:

- ☑ Is there a validated statement of vision, goals, and objectives?
- ☑ Have constraints been collected from all key stakeholders?
- ☑ Have the evaluation criteria in comparing alternatives been selected, validated, and documented?
- ☑ Is there a comprehensive list of candidate solutions, both technical and procedural?
- ☑ Is there a comprehensive and varied list of alternative concepts?
- ☑ Is the "Do Nothing" case one of the alternatives?
- ☑ Has the comparison approach been documented and validated?
- ☑ Has the selected concept, and the rationale for its selection, been documented; and has it been reviewed by the stakeholders?
- ☑ Does the documentation satisfy relevant reporting standards, if any, for example, for a Study Report if required by the state?
- ☑ Do the conclusions and recommendations flow in a clear and defensible manner from the needs, alternatives selection, and analysis?

For Programming (of ITS projects):

- ☑ Are ITS projects identified in the TIP (or STIP) project list?
- ☑ Have regional TIP requirements been addressed in creating the information for each project (including identification of ITS aspects of project if that is a part of the requirements)
- ☑ Has relevant information from the regional ITS architecture been included in the project information in the TIP (if this is one of the regional requirements)?
- ☑ Are capital projects that include ITS elements identified in the project list?

3.3.4 Project Planning

OBJECTIVES	Create the detailed plans for the project that define the activities, resources, budget, schedule and systems engineering processes to be used for the project.
DESCRIPTION	Project Planning is an effort that occurs at the beginning of a project to identify the tasks, resources, schedule and risks of the project. The result of this planning is the Project Management Plan (PMP) which identifies the detailed work plans for all tasks. It identifies key events and the technical and program milestones, and establishes a schedule for the project. For projects with systems engineering, a Systems Engineering Management Plan will also be a key output, either as a section of the PMP, or if the project's complexity warrants, a stand-alone document. Based on project complexity, additional [e.g., Configuration Management, Risk Management and Procurement] plans may be needed.
CONTEXT	
INPUT *Sources of Information*	• Goals and Objectives • Project Charter • Programming outputs for project
PROCESS *Key Activities*	• Define project objectives, tasks, resources, schedule, and risks • Define needed systems engineering processes • Prepare project management plan • Prepare supporting management plans
OUTPUT *Step Results*	• Project Management Plan • Systems Engineering Management Plan • Additional Management Plans

Overview

Project planning is one of the five process groups of the Project Management cross-cutting activity (3.4.1). This is the initial step in the project development process, occurring prior to the commencement of development activities. In order to achieve the benefits of the use of systems engineering, the tasks, processes, resources, and schedule of the project need to be defined and documented in a project management plan (PMP). The scope of this step is highly dependent on the complexity of the project and on the processes that have been developed by the agencies involved.

Risks to be Managed

The identification of risks and the definition of a plan to manage those risks is one of the key outputs of the project planning process step. While risks largely fall into three general categories, technical, cost, and schedule, each project has a unique set of risks that must be identified and a management approach determined. For further discussion of risk see the cross-cutting topic Risk Management (section 3.4.4)

Activities

Define project objectives, tasks, resources, schedule, and risks

The first task in planning the project is to identify and define all of the work efforts [tasks] which are needed to accomplish the project's goals. These tasks include all the technical work, but may also include project management itself and other administrative tasks. A clear definition of the project objectives is also a key part of the planning effort.

The resources needed for each task must also be identified. In addition, the staffing responsibilities should be defined, often through an Organization chart, in order to identify clear paths of both responsibility and communications. Other resources, such as a testing laboratory, may not be needed immediately. However, the need for them should be identified as soon as possible.

An understanding of the project's tasks, plus the resources and budget needed for each task, are combined into a project schedule. This schedule is generally constrained by external requirements, such as, a need for the system to be operational by a certain date or a dependence on the installation of another interfacing system. Key in developing the project schedule is to identify the dependencies among tasks. The most common dependency is that the completion of one task is required before the start of a subsequent task.

Another key output of this activity is identification of the risks that are anticipated for the project. Each project will have unique technical, schedule, and budgetary risks that should be identified and a plan for managing the risks should be developed. See the Risk Management cross-cutting topic (Section 3.4.4) for a further discussion of risk identification and risk management.

Define needed systems engineering processes

The project and engineering management will identify the systems engineering processes and resources necessary to support each identified technical task. If significant portions of the systems engineering tasks are contracted to commercial firms, those firms may have to be involved in detailing these processes.

Prepare project management plan

The various parts of the project plan need to be gathered together into a written Project Management Plan. The degree to which the Project Management Plan needs to be documented will vary by project size and complexity.

Prepare supporting management plans

The other plan that needs to be developed is the Systems Engineering Management Plan, which describes the systems engineering processes that will be a part of the project. The scope of this is highly dependent on the complexity of the project and on the processes that have been developed by the agencies involved. For many projects the SEMP information will be incorporated into the PMP, but for more complex projects the SEMP may be a standalone document.

If necessary, additional separate supporting plans are developed to supplement the PMP, such as a software development plan, risk management plan, configuration management plan and other technical plans.

Tailoring this Step

The degree to which various management plans are documented is the prime variable in this process step. They must be documented enough so that the responsible staff knows what to do [the larger the staff, the more important this is]. For small and low risk projects, a 5-10 page document [the Project Management Plan] is all that may be needed to contain all the necessary project planning information. Existing organizational procedures should be referenced in the plan. Depending on the nature of the project, the systems engineering processes needed should be described as part of the PMP. If the project includes custom software development, a more complete SEMP, possibly even a stand-alone document, is probably necessary. In addition, the system's owner must have available a Configuration Management [CM] Plan designed for software products. The system's owner must ensure the organization's standard CM Plan is sufficient. If it isn't, tailor it to the project or have one prepared.

As a minimum, a PMP should consider tasks, resources, schedule, systems engineering processes and identification and management of risks. For projects with higher complexity or risk a larger, more complete PMP will be needed.

The definition of systems engineering process for the project is definitely not one-size fits all. Since systems engineering is intended to address the technical challenges in building a system, it must be tailored to the technical challenges of the specific system.

The biggest variable affecting the scale of the systems engineering analysis is the need to develop custom software applications. If custom software development is needed, requirements definition and design become much more complex and a separate SEMP is usually the best approach.

Projects that only involve the purchase and installation of hardware or hardware with embedded software applications do not require the same depth of requirements analysis and design. Of course, these projects may require serious trade studies on such issues as product selection, site selection, or communications alternatives. The definition of the systems engineering effort (in the SEMP) for such projects may be quite short and can be combined into the PMP for efficiency.

Another factor is the degree to which the system owner is comfortable with the technologies involved. If the system owner is unsure or there is a perceived risk, then added attention to the preparation of a SEMP is advised.

The final factor is the degree to which the System Engineer and Development Teams have their own well-developed processes, such as requirements management, configuration management, or software development.

Where the agency does not have any of these processes in place, it is recommended that they identify and select experienced development firms with established processes. In such cases, the SEMP should reference these processes [tailored appropriately] and only deal in detail with the unique processes needed for the project.

Policy or standard for Process Step

Of all the processes described in this Document, project planning is the one which is most likely to be defined and controlled by established agency procedures. Almost all agencies have internal rules, regulations, and guidelines for project management activities. Furthermore, in the area of procurement, project management intersects with contract law, making it subject to legal requirements. It is the task of project management to be aware of, use, and be compliant with this guidance.

With regards to the systems engineering aspects of the planning, the FHWA Regulation does not specifically mention Systems Engineering Plan development practices to be followed.

The IEEE Standard for Application and Management of the Systems Engineering Process [IEEE-1220] focuses on the engineering activities necessary to guide project development. Annex B of IEEE-1220 provides a template and structure for preparing a systems engineering management plan along with an informative discussion of each section and subsection.

Traceable Content

The key artifacts determined by the Project Planning process step are the definition the Project Management Plan and supporting plans, identified in **Table 3**, with their backward and forward traceability to other process artifacts or external documents.

Table 3: Project Planning Traceability

Traceable Artifacts	Backward Traceability To:	Forward Traceability To:
Project Management Plan Systems Engineering Management Plan	Project programming	All the other Project-related Process Steps

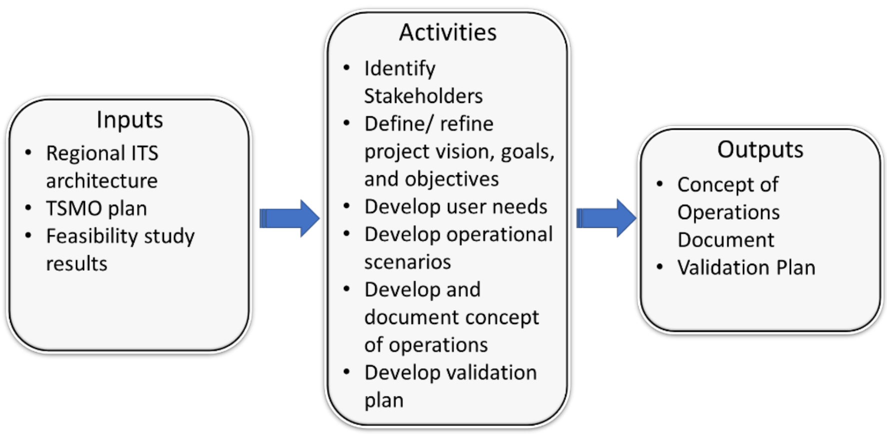
Checklist

The following checklist can help answer the question “Are all the bases covered?” once the activities above have been planned or performed. **The checklist is not sufficient by itself.**

- ☑ Has an effective project manager been selected?
- ☑ Have all project tasks been identified?
- ☑ Have all project tasks been defined enough so they are understood by the performing organization?
- ☑ Are all needed systems engineering process steps, along with their process, inputs, and outputs identified?
- ☑ Does the performing organization agree the task budget is sufficient?
- ☑ Has a project schedule been developed, reviewed, and agreed to by all parties?
- ☑ Does the performing organization agree the project schedule is sufficient?
- ☑ Are all necessary technical reviews identified and planned?
- ☑ Is the required content of each deliverable document clear to the performing organization?

- ☑ Are the project risk areas adequately identified and a management of the risks addressed?
- ☑ Have the necessary documents to support procurement of a contracted effort been prepared [the Request for Qualifications and/or Proposal]?
- ☑ Are the Project Management Plan, Systems Engineering Management Plan and any supporting plans documented?

3.3.5 Concept of Operations

OBJECTIVES	The Concept of Operations includes • High-level identification of user needs and system capabilities in terms that all project stakeholders can understand. • Shared understanding by system owners, operators, maintainers, and developers on the who, what, why, where, and how of the system • Agreement on key performance measures and a basic plan for how the system will be validated at the end of project development.
DESCRIPTION	The Concept of Operations (ConOps) is a stakeholder-oriented description of the system being developed. This ConOps will present each of the multiple views of the system corresponding to the various stakeholders. These stakeholders include operators, owners, developers, maintenance, management, and in some cases end users. The documentation of the ConOps can be easily reviewed by the stakeholders to get their agreement on the system description, operations, and maintenance. It also provides the basis for validating the system being built.
CONTEXT	
INPUT *Sources of Information*	• Regional ITS Architecture • TSMO Plan or other planning documents • Recommended concept and feasibility study from previous step
PROCESS *Key Activities*	• Identify stakeholders • Define/ refine project vision, goals, and objectives • Develop user needs • Develop operational scenarios • Develop and document concept of operations • Develop validation plan

OUTPUT *Step Results*	- Concept of Operations describing who, what, why, where, and how of the project/system, including stakeholder needs. - Validation Plan defining the approach that will be used to validate the project delivery

Overview

The Concept of Operations is a stakeholder-oriented description of the system that is being developed by the project. The purpose of the Concept of Operations is to clearly convey a high-level view of the system to be developed that each stakeholder who has a stake in the system (e.g. owner, operator, maintainer, developer, management, and in some cases the end user) can understand, as shown in Figure 12.

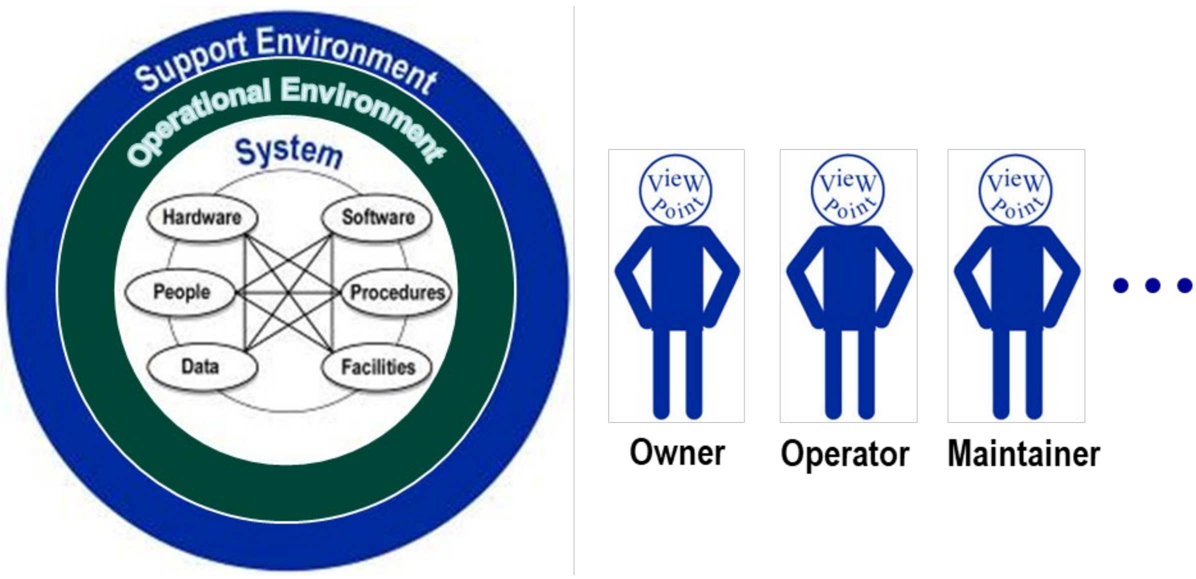


Figure 12: Concept of Operations
(Source: Adapted from ANSI/AIAA-G-043-1992)

A good Concept of Operations answers who, what, where, when, why, and how questions about the project from the viewpoint of each stakeholder.

- Who – Who are the stakeholders involved with the system?
- What – What are the elements and the high-level capabilities of the system?
- Where – What is the geographic and physical extent of the system?
- When – What is the sequence of high-level activities that will be performed?
- Why – What does your organization lack that the system will provide?
- How – What resources are needed to develop, operate, and maintain the system?

Risks to be Managed

Risk of not building the right system. Avoid the outcome that at the end of the project one or more stakeholders say: “that’s not what we needed!” This is primarily a technical risk.

Activities

Some of the key activities relating to the development of the Concept of Operations are:

Identify the stakeholders associated with the system/project – SE in general, and this process step in particular, requires participation from the stakeholders who have a stake in the project. One of the first steps in developing a ConOps is to make sure all the relevant stakeholders – owners, operators, maintainers, users, etc. – are identified and involved. You can start with the stakeholder list from the regional ITS architecture (the "operational planning process" discussed earlier) and then expand it to identify the more specific organizations – divisions and departments that should be involved. One of the most effective ways to involve the stakeholders is to create an integrated product team (IPT) of stakeholder representatives (or alternatively called a "project working group") that brings together the necessary expertise and provides a forum for all project stakeholders.

If you hire a consultant, don't assume that this is the end of your responsibilities for ConOps development. The stakeholders remain the foremost experts on their needs and must be materially involved in the development. The consultant can provide technical expertise on how to create an effective ConOps output (see discussion below), facilitate the meetings and outreach activities, prepare the documentation, and coordinate the review, but it's the stakeholders' concept that should be documented in the end. The stakeholders should consider the documentation of User Needs to be THEIR document, not the consultant's document.

The best person to write the documentation may not be the foremost technical expert on the proposed system. Stakeholder outreach, consensus building, and the ability to understand and clearly document the larger picture are key. It's often said that "you can outsource the work (to a consultant), but you can't outsource the responsibility."

Define/ Refine project vision, goals, and objectives

If not already developed, the vision, goals, and objectives of the project should be discussed and documented. A vision statement, usually only associated with large projects, is a simple paragraph length statement describing in non-technical terms the end result the project is expected to achieve. Goals and objectives describe what the potential project should accomplish from the point of view of the operating agencies and their operators, and other stakeholders (e.g system users). If already developed the vision, goals, and objectives should be revisited and expanded or elaborated as necessary to capture multiple viewpoints.

Develop user needs

Getting agreement on a project's needs is essential to the successful completion of every project. The effort to define the needs will be performed by those who have a stake in the system. The needs are most commonly referred to as user needs, but could be more precisely defined as the needs of those groups that will operate, develop, maintain, and manage the system. For certain systems this list may also include the end user, but for many of the common traffic management related projects the end user will not be a key stakeholder. For example, for a project deploying additional dynamic message signs, drivers are the ultimate end users, but they will not be a part of the group that develops the user needs.

For some projects, the needs may be well known. For example, a municipality has five Closed Circuit Television (CCTV) cameras deployed and would like to add three additional (and functionally identical) cameras at key congestion points within their network. In this case the needs have been previously considered and agreed to (whether explicitly or implicitly), and the needs of the additional cameras are the same as for the original cameras. In other cases, the needs might be known only at a very high level (e.g. "the region would like to improve data sharing between agencies"), but a careful documentation of the needs will improve the likelihood of a successful outcome.

Sometimes a needs statement such as in the prior example ("the region would like to improve data sharing between agencies") might be interpreted differently by different stakeholders, and that ambiguity may lead to the risk of not building the right system (at least in the minds of some stakeholders). Needs should be validated right after the project goes operational. Validation of needs means testing whether the needs are satisfied by the now operational project. Documenting (and getting stakeholder agreement) at the beginning of the project how the needs will be validated can clarify in the minds of the stakeholders exactly what the needs mean. For example, the need might be tested by "1). The CCTV camera images shall be visible to the County Fire Department dispatcher in real-time. And 2). The camera PTZ (pan-tilt-zoom) controls shall be operable by the County Fire Department dispatcher during an incident." If there are stakeholders who think that this test is not the data sharing, they had in mind, it's best to find out early in the project when the needs are being discussed and agreed to.

Incrementally create the user need(s), review relevant portions with stakeholders, and adjust the needs as necessary to get buy-in. All stakeholders do not have to agree on every aspect of the project, but all stakeholders must feel like they are achieving their major needs for the project.

The user needs provide an expression of the end users' operational needs that can be met by the system that will be developed or upgraded. User needs have a well-defined set of criteria that have been used for systems engineering development for decades. Well written user needs all share the following criteria:

1. Uniquely Identifiable. Each need must be uniquely identified (i.e., each need assigned a unique number and title). This criterion is necessary for traceability that occurs in other processes.
2. Major Desired Capability (MDC). Each need should express a major desired capability in the system, regardless of whether the capability exists in the current system or situation or is a gap.
3. Solution Free. Each need should be solution free, thus giving designers flexibility and latitude to produce the best feasible solution. This means the needs should be neutral to technology (even though some stakeholders might have an idea of the one or more solutions/technologies they believe are needed).
4. Capture Rationale. Each need should capture the rationale or intent as to why the capability is needed in the system. This can be as brief as a single sentence (or a reference to the need in a prior project), or a few paragraphs, or a trade-off study between competing alternative needs.

User needs can address a variety of need areas including:

- Operations (e.g. reduction of staff to collect tolls, reduction or elimination of dwell time to collect tolls),
- Maintenance (e.g. reduce the frequency of hardware faults), or

- Strategic (e.g. extend transportation services to new geographic areas).

Because there may be competing needs for limited resources, needs can be ranked (e.g. in a trade study) by the value of the satisfaction of the need vs competing needs. A trade study can use objective criteria for comparing needs, including:

- Safety improvement e.g. travel crash/death/injury rate reduction.
- Travel cost reduction
- Air quality improvement
- Travel time reduction
- Travel time reliability improvement
- Security incident reduction
- Incident response time reduction
- Traveler satisfaction improvement

Including how each user need will be validated after the new ITS project becomes operational will clarify the user need in the minds of the stakeholders.

Develop operational scenarios

Operational scenarios are an excellent way to work with the stakeholders to define a key aspect of a ConOps. Scenarios associated with a major incident, a work zone, or another project specific situation provide a vivid context for a discussion of the system's operation. It is common practice to define several scenarios that cover normal system operation (the "sunny day" scenario) as well as various fault-and-failure scenarios.

Develop and document concept of operations

The ConOps should be an approachable document that is relevant and understandable to all project's stakeholders. It should be relevant to system operators, system maintainers, system developers, system owners/decision makers, other transportation professionals, and in some cases the end user. The art of creating a good Concept of Operations lies in using natural language and supporting graphics so that it is accessible to all while being technically precise enough to provide a traceable foundation for the system requirements document and system validation.

Graphics should be used to highlight key points in the Concept of Operations. At a minimum, a system diagram that identifies the key elements and interfaces and clearly defines the scope of the project should be included. Tables and graphics can also be a very effective way to show key goals and objectives, help illustrate operational scenarios, etc.

Portions of the concept of operations can often be created from existing documents. For example, the regional ITS architecture should include stakeholder roles and responsibilities and high-level information flows that stakeholders have agreed can be used. An operations planning system study report, if prepared, or other preliminary study documentation, may provide even more relevant and refined information. Even a project application form used to support project programming will normally include high-level goals and objectives and other information, when relevant, that should be included (or referenced) in the Concept of Operations for continuity with the existing planning investments. An MPO

Long Range Planning document or the original project business case documentation for the project can often be used to trace backwards from the project needs.

Last but not least, the ConOps development phase is when concerns related to security, including cybersecurity and physical security are first considered. Security definition should occur in parallel with systems engineering steps, incrementally achieving greater levels of detail based on the SE content created at each step, until the security design is completed in the design step. Note that cybersecurity is a particularly specialized field, so for systems with significant security concerns specialized personnel with expertise in cybersecurity may be required as part of the project team.

Develop validation plan

In addition to setting expectations for the stakeholders about what the system will achieve, the user needs can serve a more formal role once the system has been developed and tested; they can help to **validate** the system. Defining the method of validating a need in advance of developing the system will clarify for all the stakeholders the precise definition and intent of the need. Creating a validation plan reduces the risk of stakeholders misunderstanding the definition and intent of the need, and can be used to manage the stakeholders' expectations from the beginning of the project. This plan is usually (but not always) created separately from the Concept of Operations documents described below.

Tailoring this Step

The level of each activity should be scaled to the size of the project. For example, a small project may have a Concept of Operations that is only a couple of pages long. The emphasis on concept exploration depends more on the newness of the project than on its size. For example, if the system will be automating activities that were formerly manual, or integrating formerly independent activities, it is a good idea to look at alternative ways for structuring the system. This will be useful for allowing the stakeholders to envision using the new system. Whenever formerly independent activities are merged, it is essential to carefully spell out the new operational responsibilities of each agency.

Policy or standard for Process Step

The FHWA Regulation/ FTA Policy requires participating agency roles and responsibilities to be identified in the systems engineering analysis for ITS projects funded from the Highway Trust Fund, including the Mass Transit Account. It also requires procedures and resources necessary for operations and management of the system to be defined. The roles and responsibilities, operations procedures and resources are initially defined and documented for the project as part of the Concept of Operations.

Two different industry standards provide suggested outlines for Concepts of Operations: ANSI/AIAA-G-043-1992 and ISO/IEC/IEEE 29148 as shown in **Figure 13**. Both outlines include similar content, although the structure of the 29148 outline lends itself more to incremental projects that are upgrading an existing system or capability. The ANSI/AIAA outline is focused on the system to be developed, so it may lend itself more to new system developments where there is no predecessor system. Successful Concepts of Operation have been developed using both outlines.

ANSI/AIAA-G-043 Outline

1. Scope
2. Referenced Documents
3. User-Oriented Operational Description
4. Operational Needs
5. System Overview
6. Operational Environment
7. Support Environment
8. Operational Scenarios

ISO/IEC/IEEE 29148 Outline

1. Scope
2. Referenced Documents
3. The Current System or Situation
4. Justification for and Nature of Changes
5. Concepts for the Proposed System
6. Operational Scenarios
7. Summary of Impacts
8. Analysis of the Proposed System

Figure 13: Alternative Concept of Operations Document Outlines

(Source: ANSI/AIAA-G-043 and ISO/IEC/IEEE 29148)

Note that this guide uses the term Concept of Operations (ConOps) where other sites like the International Council of Systems Engineering (INCOSE) use the term Operational Concept when discussing a project level document that captures the needs the stakeholders have and how those needs will be met in the system. In the ITS industry, an Operational Concept is defined in 940.9 (c) (3) as "An operational concept that identifies the roles and responsibilities of participating agencies and stakeholders in the operation and implementation of the systems included in the regional ITS architecture;". Thus, an Operational Concept tends to be associated with a broader regional view of ITS within the ITS industry, while the Concept of Operations is a more specific project or system-level document. When you consult references outside the ITS industry, these terms may be used in precisely the opposite way.

Traceable Content

The key artifacts determined by the Concept of Operations development process are the definition of Stakeholder Needs, identified in **Table 4**, with their backward and forward traceability to other process artifacts or external documents. Traceability from needs to requirements will be discussed in the next section.

Table 4: Concept of Operations Traceability

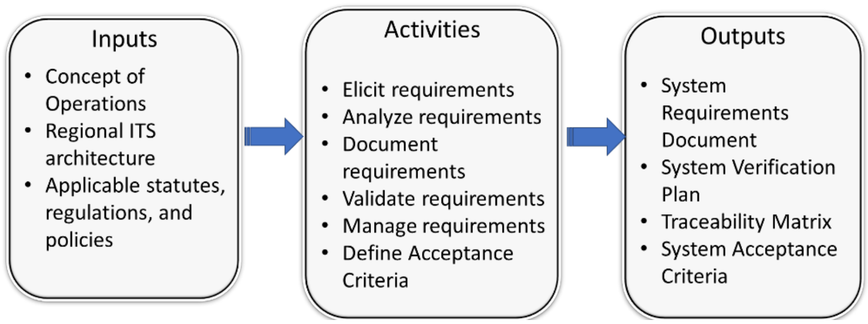
Traceable Artifacts	Backward Traceability To:	Forward Traceability To:
Concept of Operations	Needs in the Long Range Plan or Original Project Business Case documentation	Requirements
Validation Plan	Concept of Operations	Validation Tests

Checklist

The following checklist can help answer the question “Are all the bases covered?” once the activities above have been planned or performed. **The checklist is not sufficient by itself.**

- ☑ Are goals, objectives, and vision (if included) evident?
- ☑ Has an identification of stakeholders and their responsibilities been made?
- ☑ Are the operations described from the viewpoints of all key stakeholders?
- ☑ Have the operational user needs been defined?
- ☑ Does the system description include external interfaces?
- ☑ Are both operational and support environment included?
- ☑ Does evidence exist for alternative concepts and rationale for the selection process?
- ☑ Have operational scenarios been documented?
- ☑ Are both normal and failure operational scenarios included?
- ☑ Is the Concept of Operations documented in an easily understood manner?
- ☑ Has the Concept of Operations been reviewed and accepted by the key stakeholders?

3.3.6 Requirements

OBJECTIVES	• Develop a validated set of system requirements that meet the stakeholder's needs.
DESCRIPTION	The stakeholder needs identified in the Concept of Operations are reviewed, analyzed, and transformed into verifiable requirements for the system. Working closely with stakeholders, the requirements are elicited, analyzed, validated, documented, and baselined.
CONTEXT	
INPUT *Sources of Information*	• Concept of Operations (Stakeholder Needs) • Functional requirements, interfaces, and applicable ITS standards from the regional ITS architecture • Applicable statutes, regulations, and policies
PROCESS *Key Activities*	• Elicit requirements • Analyze requirements • Document requirements • Validate requirements • Manage requirements • Define Acceptance Criteria
OUTPUT *Process Results*	• System Requirements Document • System Verification Plan • Traceability Matrix • System Acceptance Criteria

Overview

The Electronics Industry Association (EIA) Standard 632, Processes for Engineering a System defines requirement as "something that governs what, how well, and under what conditions a product will achieve a given purpose." This is a good definition because it touches on the different types of requirements that must be defined for a project. Functional requirements define "what" the system must do, performance requirements define "how well" the system must perform its functions, and a variety of non-functional environment requirements define "under what conditions" the system must operate.

One of the most important attributes of a successful project is a clear statement of requirements that meet the stakeholder's needs. Unfortunately, creating a "clear statement of requirements" is often much easier said than done. The initial list of stakeholder needs will normally be a jumble of requirements, wish lists, technology preferences, and other disconnected thoughts and ideas. A lot of analysis must be performed to develop a good set of requirements from this initial list.

Risks to be managed

Needs are Satisfied: The key risk being managed is that the system defined will satisfy all the needs identified in the previous process. This ensures that the ITS project system requirements, i.e., how the project "works", supports the selected project needs (traceability to needs), and conversely, that all project needs are supported by one or more system requirements. As described above the requirements should address the functions to be performed by the system, how well the system performs these functions, and the conditions under which the functions must be performed.

Activities

The basic activities of requirements definition are shown in Figure 14 and include the basic steps of elicitation, analysis, documentation, validation, and management. The actual approach taken to performing these steps will vary by organization and project. There isn't one "right" approach for requirements development. Different organizations develop requirements in different ways. Even in the same organization, the requirements development process for a small ITS project can be much less formal than the process for the largest ITS projects that specify complex hardware/software systems. The differences are primarily in the details and in the level of formality.

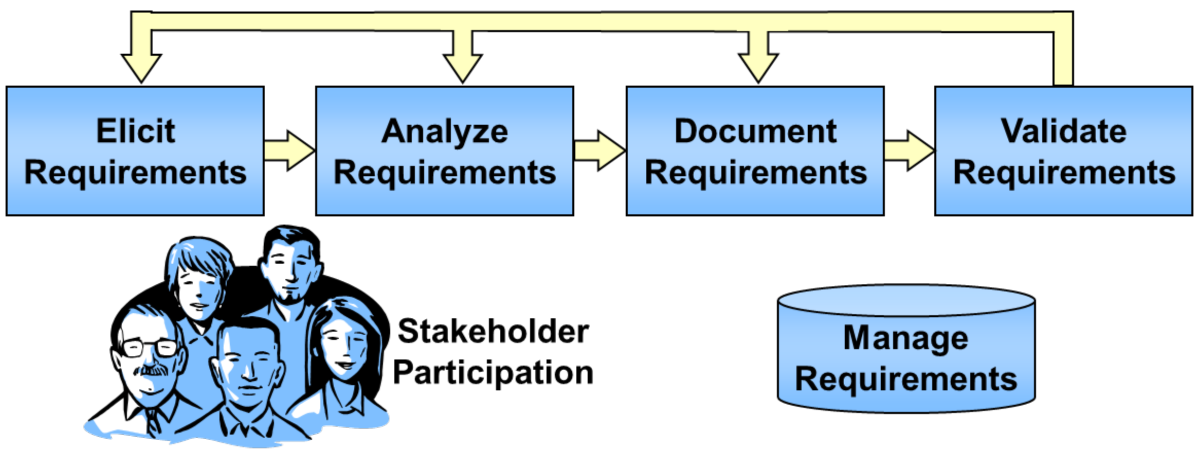


Figure 14: Requirements Definition Activities
(Source: FHWA)

Note that each of these activities shown in the figure can be highly iterative. In the course of a day, a systems engineer may do a bit of each of the activities as a new requirement is identified, refined, and documented.

Elicit Requirements – Building on the stakeholder needs and other input such as the functional requirements from the regional ITS architecture, and any relevant statutes, regulations, or policies, define a strawman set of system requirements and review and expand on these requirements, working closely with the project stakeholders. There are many different elicitation techniques that can be used including interviews, scenarios (see discussion under the Concept of Operations), prototypes, facilitated meetings, surveys, and observations. Each of these techniques can be used in combination to discover the stakeholder's requirements.

As part of this step, the requirements developers should identify the classes of requirements that will be the subject of their effort:

- Operational
- Maintenance
- Performance
- Security
- Environmental (e.g. what other systems the project will need to interface or coexist with)

Elicit and *elicitation* are words you may not run into every day. *Elicit* means to draw forth or to evoke a response. This is the perfect word to use in this case because you have to do some work to draw out the requirements from the stakeholders and any existing documentation. More work is implied by "elicit requirements" than if we said "collect requirements" or even "identify requirements", and this is intended.

Having the right stakeholders involved can make or break a requirements development effort. It isn't enough to make sure the right organizations are involved. You should ensure that the right individuals within those organizations are involved. For example, it isn't enough to engage someone from the maintenance organization – it should be an electrical maintenance person who has experience with ITS equipment maintenance for ITS projects. Finding individuals with the right combination of knowledge of current operations, vision of the future system, and time to invest in supporting requirements development is one of the key early challenges in any requirements development effort.

Analyze Requirements – The stakeholder requirements that are elicited are analyzed in detail and the stakeholders negotiate to select the requirements that must be implemented and priorities may be assigned to the requirements. This is where the requirements are cleaned up – conflicts are resolved, gaps are identified and addressed, ambiguity and redundancy are removed, and the requirements are organized and decomposed into more specific supporting requirements.

For larger systems, it can be very difficult to "get your arms around" all of the requirements. Requirements modeling tools provide a graphical way to define requirements so that they are easier to understand and analyze. Tools range from simple repositories that allow you to manage your requirements and traceability to user needs and design elements to full-featured requirements management systems that store many attributes for each requirement that allow you to track requirements changes, supporting baseline management over the life of the project. These tools are particularly useful for more complex ITS projects.

There are numerous requirements modeling tools and techniques available that can help you model the system as part of the analysis process. These tools support a variety of development methodologies (e.g. agile) and can range from simple database extensions to applications that support development of system models from different perspectives, fully integrate diagrams and text and will support creation of documentation of the requirements.

INCOSE and the Object Management Group (OMG) have collaborated on a standard System Modeling Language (SysML) that is an extension of the Universal Modeling Language (UML) specifically to support SE. INCOSE maintains a data repository of available modeling tools that is available on their web site. These tools do require effort (possibly significant in the case of the most complete applications) to set up and maintain, so their use is likely justified only in the largest ITS projects.

Document Requirements – The requirements are documented in a well-organized and approachable fashion so that the system stakeholders and system development team can all easily understand and review the requirements. Normally, a combination of plain language and diagrams are used to define the requirements.

The level of detail needed in the requirements definition phase can depend on later design choices. For example, planning to use a proven off-the-shelf solution for a component of the project design may mitigate the need for detailed requirements analysis of the selected solution. In this case only higher-level requirements covering the overall selected solution may be sufficient.

If requirements for a project are identical to requirements for a prior project – those requirements can be referenced and not necessarily redeveloped. Note that because requirements are generally technology neutral, the design for reused requirements (using newer and current technology) may still be necessary.

As the requirements are documented, a plan for verifying the system based on the requirements is defined. A verification method is identified for every requirement – normally you select one of four fundamental ways to verify each requirement: Test, Demonstration, Inspection, or Analysis. The purpose of this early assignment, long before the requirements will actually be verified, is to make sure the requirements author thinks about how the requirement will be verified from the very start. Only verifiable requirements should find their way into the system requirements.

The method of verification is only one of the attributes that should be tracked for each requirement. A rich set of attributes is particularly important for large complex projects. Consider specifying attributes like the following for each of your requirements if you are developing a large complex project: Source, author, creation date, change history, verification method, priority, and status. The historical and change tracking attributes are particularly important since they allow management to measure and track requirements stability.

Traceability is another important aspect of the requirements documentation. Each requirement should trace to a higher-level requirement, a stakeholder need or other governing rules, standards, or constraints that the requirement is derived from. As the project is developed, each requirement will also be traced to the verification test case that will verify the requirement, more detailed "child" requirements that may be derived from it, and design elements that will implement the requirement, as applicable. Establish and populate the traceability matrix at this stage and then continue its population during development; don't wait until the end.

Validate Requirements – The documented requirements are carefully checked for consistency, accuracy, and completeness. Also check for "compound requirements", which are requirements that contains two or more statements, each of which is a distinct requirement with its own individual verification path. This is a critical step that is intended to identify requirements defects as early in the process as possible when correcting defects is most economical. To support validation, requirements walkthroughs are held to review the requirements in a systematic way with the project stakeholders and project team.

Table 5 identifies an oft-repeated list of attributes of good requirements. As part of the validation process, you do your best to make sure that the requirements have all of these desired attributes. Unfortunately, computers today can only do a fraction of this validation and people have to do the rest. Techniques for validating a requirement against each of these quality attributes are also shown in Table 5. An attribute list like this can be converted into a checklist that prompts reviewers to ask themselves the right questions as they are reviewing the requirements.

Table 5: Validating Quality Attributes of Good Requirements

Quality Attribute	Validate By:
Necessary	Make sure that each requirement traces to a stakeholder need or a parent requirement. A computer can check that the traceability is complete, but people have to verify that the identified traces are valid.
Clear	Look for red flag words and constructs in the requirements e.g. "user friendly", "optimum", "real-time", "and/or", "etc.". The vast majority of this aspect of validation relies on walkthroughs and other reviews to make sure the requirements aren't subject to different interpretations. The main culprit here is ambiguity in the English language. One of the most common findings during a walkthrough is that the wording of a requirement is not clear to all the stakeholders, a condition that can usually be resolved by rewording of the requirement
Complete	Does every stakeholder or organizational need trace to at least one requirement? If you implement all of the requirements that trace to the need, will the need be fully met? A computer can answer the first question, but only stakeholder(s) can answer the last.
Correct	In general, it takes a walkthrough to verify that the requirements accurately describe the functionality and performance that must be delivered. Only the stakeholders can say whether the highest-level system requirements are correct and consistent. Traceability can assist in determining the correctness of lower level requirements. If a child requirement is in conflict with a parent requirement, then either the parent or child requirement is incorrect.

Quality Attribute	Validate By:
Feasible	Again, this must be determined by review and analysis of the requirements. A computer can help with the analysis and possibly even flag words like “instant” or “instantaneous” that may be found in infeasible requirements, but a person ultimately makes the judgement of whether the requirements are feasible. In this case, it is the developer who can provide a reality check and identify requirements that may be technically infeasible or key cost drivers early in the process.
Verifiable	Does the requirement have a verification method assigned? (This is something the computer can check.) Is the requirement really stated in a way that is verifiable? (This much more difficult check can only be performed by people.) For example, ambiguous requirements are not verifiable.

Manage Requirements – Processes and tools are established to manage the requirements and associated information that is collected, track changes to the requirements, and provide facilities that support traceability, requirements retrieval and reporting, etc.

TOOL

Like the other requirements engineering activities, the requirements management capabilities should be scaled based on the complexity and size of the ITS project. Large complex ITS projects can benefit from a tool specifically for requirements management such as DOORS or Requisite-Pro. Requirements for smaller scale ITS projects can easily be managed with a general-purpose database like Microsoft Access. A professional requirements management tool includes a long list of capabilities including change management, requirements attributes storage and reporting, impact analysis, requirements status tracking, requirements validation tools, access control, and more. Whether simple or sophisticated, every project should have some means of managing their requirements baseline.

No matter how you developed your requirements, you must document them in some consistent, accessible, and reviewable way. The requirements development process may result in several different levels of requirements over several steps in the “Vee” – stakeholder requirements, system requirements, subsystem requirements, etc. that may be documented in several different outputs. For example, stakeholder requirements might be documented in a Concept of Operations in a series of Use Cases, system requirements may be documented in a System Requirements Specification, and subsystem requirements may be documented in subsystem specifications. Ideally, all of these requirements are managed in a single repository that can be used to manage and publish the requirements at each stage of the project.

TIP

It is much easier to use a standard template for the requirements than it is to come up with your own, and numerous standard templates are available. If your organization does not have a standard requirements template, then you can start with a standard template like the one contained on this website (see Section 6.5 for the Requirements Template) or ISO/IEC/IEEE 29148:2018, Systems and software engineering - Life cycle processes - Requirements engineering. Starting with a template saves time and ensures that the requirements specification is complete. Of course, the template can be modified as necessary to meet the needs of the project.

Another good starting point is a model document. FHWA has developed a set of model documents around systems that are commonly deployed around the country. Currently there are several published model documents:

- Model Systems Engineering Documents for Central Traffic Signal Systems, FHWA-HOP-19-019
- Model Systems Engineering Documents for Adaptive Signal Control Technology (ASCT) Systems, FHWA-HOP-11-027
- Model Systems Engineering Documents for Dynamic Message Sign (DMS) Systems, FHWA-HOP-18-080
- Model Systems Engineering Documents for Closed Circuit Television (CCTV) Systems, FHWA-HOP-18-060

Each is available in HTML or PDF from https://ops.fhwa.dot.gov/publications/publications.htm.

A set of system requirements should fully specify the function, performance, and environmental characteristics of the system to be developed. The requirements document should include the following information:

- System boundary with interfacing systems clearly identified
- External interface requirements for interfacing with other systems and people
- Functional requirements and associated performance requirements
- Cybersecurity requirements
- Environmental requirements (physical as well as technology environment if appropriate)
- Lifecycle process requirements supporting production, deployment, transition, operations and maintenance, change and upgrade, and retirement/replacement, as applicable. Possibly including cost requirements of each or some of the stages of the lifecycle.
- Staffing, human factors, safety, and security requirements.
- Physical constraints (weight, form factors).

Define Acceptance Criteria- As a part of this step, the acceptance criteria are identified and documented. Acceptance is the final step taken before the system initial deployment is undertaken. One of the key aspects of Acceptance Criteria is successful verification of the system against the Verification Plan. However, the acceptance step may include additional criteria, such as a period of successful operations by the agency operation staff. Depending on the scope and complexity of the project, the Acceptance Criteria may be defined in a separate document, or included in the Verification Plan.

Tailoring This Step

In this activity, there are no shortcuts. Requirements development is a critical process for new systems. On small systems, the owner may be able to reduce the number of requirements documents by combining the system and sub-system requirements. For systems that are being expanded, the initial set of requirements may be sufficient to support the expansion, but any existing requirements should be reviewed to see if they completely address all the needs. Finally, if the project relates to the devices covered by Model SE documents, then those documents may be an excellent source of requirements that can be tailored to the specific project.

Policy or Standard for Process Step

The FHWA Regulation requires that requirements be developed for ITS projects funded with the Highway Trust Fund, including the Mass Transit Account. ISO 29148 describes the processes and products related to requirements engineering for systems and software.

Traceable Content

The key artifact determined by the System Requirements development process are the Requirements, identified in Table 6 with their backward and forward traceability to other process artifacts or external documents.

Table 6: Stakeholder Requirements Traceability

Traceable Artifacts	Backward Traceability To:	Forward Traceability To:
System Requirements	Needs in the Concept of Operations	• High-Level Design Modules • Low-Level Design Specifications • System Verification Tests

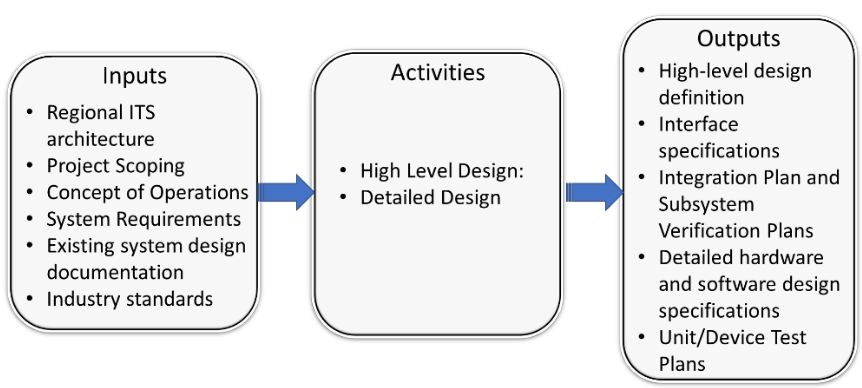
Checklist

The following checklist can help answer the question "Are all the bases covered?" once the activities above have been planned or performed. **The checklist is not sufficient by itself.**

- ☑ Were the requirements documented?
- ☑ Was a requirements walkthrough held to validate the requirements?
- ☑ Was each requirement checked to see that it met all of the following?
  - ☑ Necessary [trace to a user need]
  - ☑ Concise [minimal]
  - ☑ Feasible [attainable]
  - ☑ Testable [measurable]
  - ☑ Technology Independent [avoid "HOW to" statements unless they are real constraints on the design of the system]
  - ☑ Unambiguous [Clear]
  - ☑ Complete [function fully defined]
- ☑ Was a verification case for each requirement developed? [test, demonstration, analysis, inspection]
- ☑ Was each user need fully addressed by one or more system requirement(s)?
- ☑ Is the requirement set complete? Have the following types of requirements been defined?
  - ☑ Functional
  - ☑ Performance
  - ☑ Enabling [training, operations & maintenance support, development, testing, production, deployment, disposal]
  - ☑ Data
  - ☑ Interface
  - ☑ Environmental
  - ☑ Non-functional [reliability, availability, safety, and security].

- ☑ Were attributes [quality factors] assigned to each requirement [Priority, risk, cost, owner, date, and verification method]? Verification methods could include demonstration, analysis, test, and inspection.
- ☑ Were the requirements reviewed and approved by the stakeholders and was a baseline [reference point for future decisions] established?
- ☑ During this process step, were periodic reviews performed? Were the reviews done in accordance with the review plan documented in the SEMP?

3.3.7 Design and Specifications

OBJECTIVES	• Produce a high-level design that selects specific technologies which satisfy the system requirements • Define the major interfaces (especially interfaces crossing institutional and system boundaries), and that facilitates development, integration, future maintenance, and upgrades. • Develop detailed design specifications for the selected technologies which supports hardware and software development and where possible procurement of off-the-shelf equipment.
DESCRIPTION	The design step encompasses two topics, high-level design and detailed design. High-level design is the transitional step between WHAT [requirements for sub-systems] the system does, and HOW [architecture and interfaces] the system will be implemented to meet the system requirements. This process includes the decomposition of system requirements into alternative project architectures and then the evaluation of these project architectures for optimum performance, functionality, cost, and other issues [technical and non-technical]. Stakeholder involvement is critical for this activity. In this step, internal and external interfaces are identified along with the needed industry standards. These interfaces are then managed throughout the development process. Detailed design for software, hardware, communications, and databases describes HOW the components will be developed to meet the required functions of the system in great detail. For computer programs, this will describe the software in enough detail so the software coding team can write the individual software modules. For hardware, this step will describe the hardware elements in enough detail to be fabricated or purchased. This level of detail is best performed by the development team who writes the software code, designing the hardware and communications, then manages the design and development process starting in this phase to the end of the development of the software and hardware. Systems engineering supports this activity by monitoring and reviewing the detailed design process and clarifies the requirements when needed. Systems engineering is involved in the periodic technical reviews during the component design process. At the completion of this step, the system's owner and stakeholders will have a Critical Design Review to review and approve the "build-to" design.
CONTEXT	

INPUT *Sources of Information*	• Portion of the regional ITS architecture for the project • Project Scoping • Concept of Operations • System Requirements • Existing system design documentation • Industry standards

PROCESS
Key Activities
	• High Level Design:
	  ○ Develop and evaluate high-level design alternative
	  ○ Evaluate off-the-shelf components
	  ○ Analyze and allocate requirements
	  ○ Define interfaces and identify standards
	  ○ Document high level design
	  ○ Define integration plan and subsystem verification plans
	• Detailed Design
	  ○ Prototype user interface
	  ○ Develop detailed hardware and software design specifications
	  ○ Select off-the-shelf (OTS) products
	  ○ Create Unit/Device Test Plans

OUTPUT *Process Results*	• High-level design definition • Interface specifications • Integration Plan and Subsystem Verification Plans • Detailed hardware and software design specifications • Unit/Device Test Plans

Overview

In the systems engineering approach, we define the problem before we define the solution. The previous processes have all focused primarily on defining the problem to be solved. The system design step is the first step where we focus on the solution. This is an important transitional step that links the system requirements that were defined in the previous process with system implementation that will be performed in the next process as shown in Figure 15.

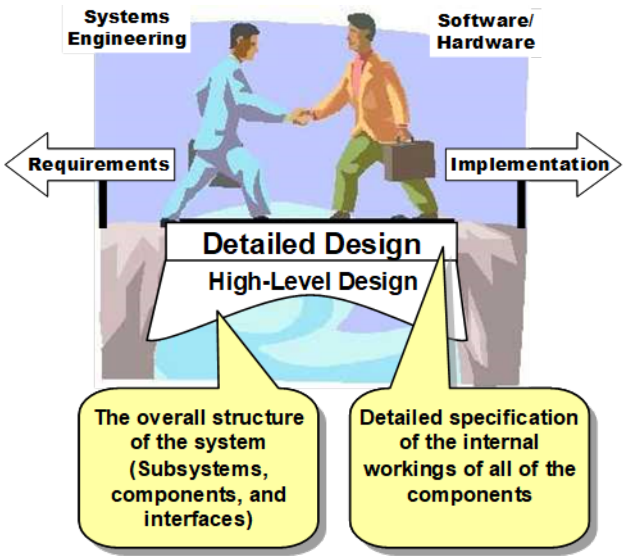


Figure 15: System Design is the Bridge from Requirements to Implementation

There are two levels of design that should be included in your project design activities:

High-level design is sometimes referred to as *architecture definition* in systems engineering handbooks and process standards. Architecture definition is used because an overall structure for the project is defined in this step. ISO/IEC/IEEE 15288 defines the purpose of architecture design as *"the process … to generate system architecture alternatives, to select one or more alternative(s) that frame stakeholder concerns and meet system requirements, and to express this in a set of consistent views."* The alternative architectures often also represent different technologies, so that high-level design is sometimes also associated with technology selection.

System requirements from the prior section should be allocated to the architecture elements (or in some cases to multiple architecture elements working together as a subassembly. In some cases, if a requirement spans multiple elements, it may be useful to decompose the requirement into more primitive sub-requirements that can be allocated to specific architecture elements or modules. Since requirements are used to verify that the system components or modules are correct, the requirements allocated to a module should be completely satisfied by that correctly built module. Some requirements may require multiple modules working together to implement, and these requirements may be allocated to the multiple modules or subassemblies for testing during the later phases of integration.

Detailed design is the complete specification of the software, hardware, and communications components (and their various technology variants), defining *how* the components will be developed to meet the system requirements allocated to them. The software specifications are described in enough detail that the software team can code the individual software modules. The hardware specifications are detailed enough that the hardware components can be fabricated or purchased.

Risks to be Managed

Design addresses all the requirements: The key risk being managed is that the system design does in fact address all of the requirements defined for the system. In other words, that all project requirements are satisfied by one or more project design technology specifications. A second risk to be managed is that the system design does not go beyond the requirements, which would likely entail additional cost and schedule for the system, which (along with scope and quality) systems engineering is seeking to control. This relationship between each system requirement and one or more design specifications is documented in a traceability matrix.

Activities

System design is a cooperative effort that is performed by systems engineers and the implementation experts who will actually build the system. The process works best when there is a close working relationship among the customer, the systems engineers (e.g., a consultant or in-house systems engineering staff), and the implementation team (e.g., a contractor or in-house team).

High-Level Design

High-level design is normally led by systems engineers with participation from the implementation experts to ensure that the design is implementable. Typical activities of high-level design are shown in **Figure 16**. Each of the activities can be performed iteratively as high-level design alternatives are defined and evaluated.

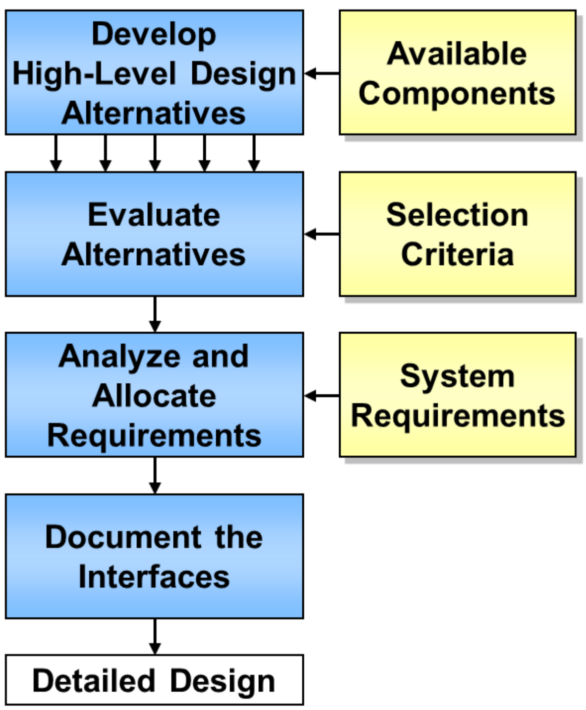


Figure 16: High Level Design Activities
(Source: FHWA)

Develop and evaluate high-level design alternatives

The system is partitioned into subsystems, and the subsystems are in turn partitioned into smaller assemblies. The process continues until system components – the elemental hardware and software configuration items – are identified. Figure 17 shows a partial decomposition of an electronic toll collection system that identifies all of the major subsystems and the components for the Video Enforcement subsystem.

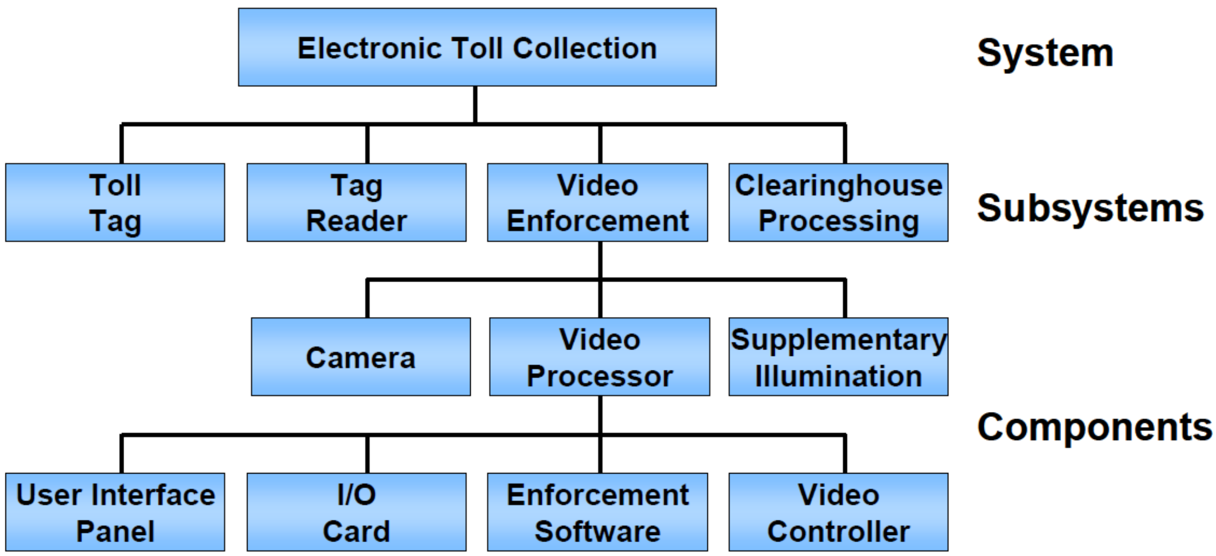


Figure 17: Electronic Toll Collection Subsystems and Components (Excerpt)
(Source: FHWA)

There are many alternative ways that a system can be partitioned into subsystems and components. In this Electronic Toll Collection example, we might consider whether the Clearinghouse Processing subsystem should be handled by a single centralized facility or distributed to several regional facilities. As another example, vehicle detectors could be included in the Video Enforcement subsystem or in the Tag Reader subsystem, or both.

Even a relatively simple traffic signal system has high-level design choices. For example, a traffic signal system high-level design can be two-level (central computer and local controllers), three-level (central computer, field masters, and local controllers), or a hybrid design that could support either two or three levels. High-level design alternatives like these can have a significant impact on the performance, reliability, and life-cycle costs of the system. Alternative high-level designs should be developed and compared with respect to defined selection criteria to identify the superior design choice.

One effective way to compare high level designs is to create a project architecture that highlights the subsystems and interfaces for each design. A regional ITS architecture may have project architectures defined that can serve as a starting point for the project architecture. In addition, the project definition from the regional ITS architecture can be imported into the Systems Engineering Tool for Intelligent Transportation (SET-IT), to create a more detailed representation of the projects systems and interfaces that are provided by the regional ITS architecture definition.

The selection criteria that are used to compare the high-level design alternatives include consistency with existing physical and institutional boundaries; ease of development, integration, and upgrading;

and management visibility and oversight requirements. One of the most important factors is to keep the interfaces as simple, standard, and foolproof as possible. The selection criteria should be documented along with the analysis that identifies the superior high-level design alternative that will be used. If there are several viable alternatives, they should be reviewed by the project sponsor and other stakeholders.

The USDOT regulation CFR 940.11 requires the systems engineering analysis for ITS projects to include an analysis of alternative system configurations.

Evaluate available components

One key aspect of high-level design is the identification of components that will be purchased, reused, or developed from scratch. The project may be required to use commerically-available hardware or software, or this may simply be the preferred solution. Specific design constraints may also require that a particular product be used. For example, a municipality that is expanding a signal control system that already includes 300 Type 170 controllers may constrain the design of the expansion to use the same controllers to facilitate operation and maintenance of the overall system. State DOTs and other large agencies often publish preapproved product lists that identify ITS-related products that meet agency requirements and/or specifications.

When commercially-available components will be used, the high-level design must be consistent with the capabilities of the target products. The designer should have an eye on the available products as the high-level design is produced to avoid requiring a design that can be supported only by a custom solution. A particular product should not be specified in the high-level design unless it is truly required. When possible, the high-level design should be vendor and technology independent so that new products and technologies can be inserted over time.

You should give commercially-available hardware and software serious consideration and use it where it makes sense. The potential benefits of off-the-shelf solutions – reduced acquisition time and cost, and increased reliability – should be weighed against the requirements that may not be satisfied by the off-the-shelf solution and potential loss of flexibility. If you have important requirements that preclude off-the-shelf solutions, determine how important they really are and what their real cost will be. This make/buy evaluation should be documented in a summary report that considers the costs and benefits of off-the-shelf and custom solution alternatives over the system life cycle. This report should be a key deliverable of the project design.

Also recognize that there is a large grey area between off-the-shelf and custom software for ITS applications. Every qualified software developer starts with an established code base when creating the next “custom solution”, accruing some of the benefits of off-the-shelf solutions. Many vendors of off-the-shelf solutions offer customization services, further blurring the distinction between off-the-shelf and custom software.

Analyze and allocate requirements

The requirements analysis described in Section 3.3.6 continues as the requirements are decomposed until there is enough granularity to allocate requirements to the system components identified in the high-level design.

The detailed functional requirements and associated performance requirements are allocated to the system components. To support allocation, the dependencies between the required system functions are analyzed in detail. Once you understand the dependencies between functions, you can make sure that functions that have a lot of complex and/or time-constrained interactions are allocated to the same component as much as possible. Through this process, each component is made as independent of the other components as possible.

Define interfaces and identify standards

Interfaces should be identified early, fully documented, and then managed throughout the project development. Interface specifications should be developed for external interfaces (i.e., interfaces between the current project and external systems) and internal interfaces (i.e., interfaces between project components). Interfaces between systems that are owned and operated by different agencies may require additional lead time to negotiate interface agreements.

This is the place to identify ITS standards and any other industry standards that will be used in detail. There are a variety of standards that should be considered at this point. Take a look at all interfaces, both external and internal. Since your regional ITS architecture and/or project ITS architecture was based on the Architecture Reference for Cooperative and Intelligent Transportation (ARC-IT), most interfaces should already have communications solutions--groups of standards that together satisfy the interfaces defined in your architecture. These solutions are defined to provide the most complete satisfaction of interface requirements, but might still have issues: mostly these will be gaps--areas where the solutions are incomplete, not fully tested or vetted. These are areas that researches and standards bodies might be working on, but point out areas where, if you are to implement, you will have additional work to do, particularly to ensure interoperability, a key issue for any interface to vehicles or handheld devices.

REGULATION

Agencies are encouraged to incorporate the ITS standards into new systems and upgrades of existing systems. The Regulation/Policy requires the systems engineering analysis for ITS projects to include an identification of ITS standards. Consult the ITS Standards Program website at https://www.standards.its.dot.gov/ for more information and available resources supporting standards implementation.

Document High Level Design

The results of the previous activities will be collected into a key output of this step- the Project Design Document. There isn't a single "best way" to present the high-level design to stakeholders and developers since different users will have different needs and different viewpoints. Over the years, high-level designs have evolved to include several different interconnected "views" of the system. Each view focuses on a single aspect of the system, which makes the system easier to analyze and understand. The specific views that are 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 information view that describes the information that will be managed by the system, and a communications view that defines the communications solutions (a related group of communications standards and specifications) that support each interface. As shown in **Figure 18**, these views are just different ways of looking at the same system.

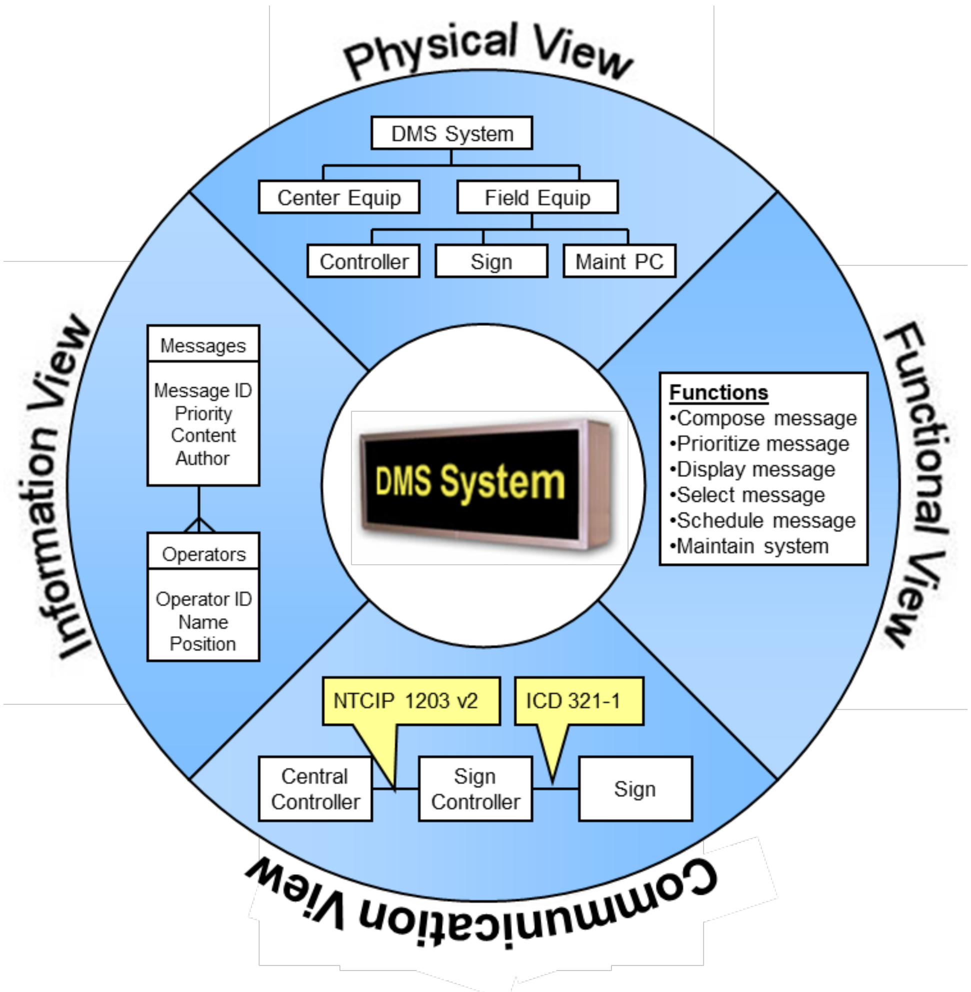


Figure 18: High-Level Design May Include Several Views

(Source: FHWA)

Create Integration Plan and Subsystem Verification Plans

An Integration Plan and Subsystem Verification Plans should be completed parallel with the high-level design. (See Section 3.3.9 for more information on integration and verification planning.)

Detailed Design

Hardware and software specialists create the detailed design specifications for each component and software module identified in the high-level design. Systems engineers play a supporting role, providing technical oversight on an ongoing basis. As you might expect, the detailed design activity will vary for off-the-shelf and custom components, as shown in **Figure 19**.

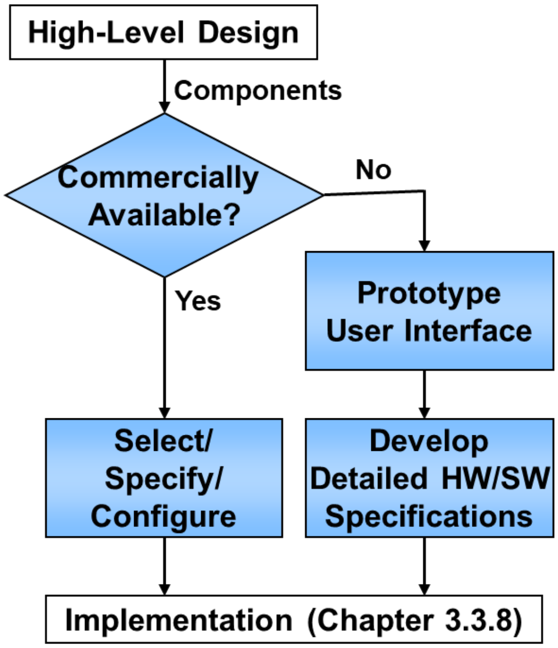


Figure 19: Detailed Design Activities
(Source: FHWA)

Prototype user interface

If a user interface is to be developed, a simple user interface prototype is an efficient way to design it. A prototype is a quick, easy-to-build approximation of a system or part of a system. A software prototype can be used to quickly implement almost any part of a system that you want to explore, but it is used most often to make a quick approximation of a user interface for a new system.

A user interface prototype should be employed to help the user and developer visualize the interface before significant resources are invested in software development. This is one area in particular where you can expect multiple iterations as the developers incrementally create and refine the user interface design based on user feedback. (You will find that it is often easier to get users to provide feedback on a prototype than on system requirements and design specifications, which can be tedious to review.)

While the user interface prototype is included here because it is an effective way to design the user interface, prototypes may actually be generated much earlier in the process, during system requirements development. The prototype can turn the requirements statements into something tangible that users can react to and comment on.

Develop detailed hardware component and software module design specifications

Detailed design specifications are created for each hardware component and software module to be developed. In the high-level design, each component is defined in terms of its allocated functional and performance requirements, with particular focus on its interfaces to external systems and other components.

The detailed design specifies exactly how the component will be implemented so that it meets the requirements. For hardware, mechanical and schematic drawings and parts lists are defined. For software, this includes specification of algorithms, data structures, and third-party software packages that will be used. For open ITS standards, this may include selection of optional standardized objects that together completely specify an interface MIB (Management Information Base).

The detailed design of each component should be reviewed to verify that it meets the allocated requirements and is fit for the intended purpose. Each specification of the detailed design specifications should be traced to higher-level requirements. This implies an expansion of the traceability matrix described under the Requirements process. Periodic or as-needed reviews can be held to monitor progress and resolve any design issues. For larger projects, coordination meetings should be held to ensure that concurrent design activities are coordinated to mitigate future integration risks. At the completion of the detailed design step, a broader stakeholder meeting is held to review and approve the detailed design before the implementation team begins to build the solution. A record of the technical reviews that were conducted should also be included in the project documentation.

Select off-the-shelf (OTS) products

One of the fundamental principles of systems engineering is to delay technology selection until you have a solid foundation for making the right choice. By waiting until this point in the process, the latest technologies and products can be considered, and these selections can be based on a thorough understanding of the requirements and the overall architecture of the system. The selections can also be made by specialists who are closest to the implementation and are therefore best equipped to make them. There are two fundamental ways that a product can be selected, depending on your procurement requirements and selected procurement strategy:

- 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 specifications.
- A competitive procurement can be used that allows vendors to propose products that will best meet the specifications. In either case, product selection should be driven by performance-based specifications of the product.

Specifications can be either performance-based or prescriptive. In a performance-based specification, you specify the functionality and the performance that are required rather than what equipment to use. In a prescriptive specification, you specify exactly the equipment that you want. A performance-based specification for a dynamic message sign would include statements like “The sign shall provide a display of 3 lines of 25 characters per line.” A prescriptive specification would be “The Trantastic LED Model XYZ sign shall be used.” Performance-based specifications tend to provide the best value because they allow the contractor or vendor maximum flexibility to propose the best solution that meets your specifications.

If a trade study is performed, then the functional, performance, and environmental requirements that are allocated to the component or module should be used to define product selection criteria. An alternatives analysis document captures the alternatives that were considered and the selection criteria that were used to select the superior product. Existing trade studies, approved product lists, and other resources can be used to facilitate product selection. The evaluation of off-the-shelf products should be reviewed to verify that the evaluation criteria were properly defined and applied fairly and that an appropriate range of products was considered.

Create Unit/Device Test Plans

Test plans should be created for each hardware component and software module to test all requirements identified in the HW/SW design specifications. The test plans will show how each design specification will be verified.

Tailoring This Step

The level of each activity should be appropriately scaled to the size and budget of the project. For example, a small project may have an analysis of alternatives that is only a page or two long, based upon qualitative comparisons. Constraining the number of sub-systems will also reduce the effort here and in the subsequent steps, such as integration and verification.

This activity is driven by the amount of custom development needed for the project. The more customized the development, the more effort there is at this step. For small systems that contain nearly all products previously used by the agency to address similar requirements, the primary activity is the evaluation of these products.

Policy or Standard for Process Step

The FHWA Regulation requires requirements to be developed for ITS projects funded with the Highway Trust Fund, including the Mass Transit Account. It also requires the analysis of alternative system configurations to meet requirements.

The IEEE 1233 Guide for developing system requirements specifications provides a standard for developing requirements.

The FHWA Regulation does not specifically mention component level detailed design practices to be followed. For software, IEEE/ISO 12207 Software Life cycle process provides specific process guidance.

Traceable Content

Table 7 identifies that the high level and detailed design specifications trace back to the system functional, performance and environmental requirements, and forward to the component, module, and interface tests.

Table 7: Design traceability

Traceable Artifacts	Backward Traceability To:	Forward Traceability To:
High-Level Design	System Requirements	• Detailed Design Specifications • Verification Tests
Detailed Design Specifications	• High Level Design • Allocated Requirements	• HW/SW Implementation • Verification Tests

Checklist

The following checklist can help answer the question "Are all the bases covered?" once the activities above have been planned or performed. **The checklist is not sufficient by itself.**

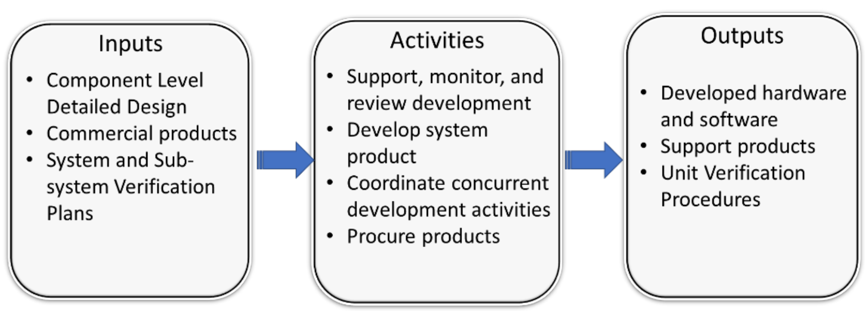
High-Level Design

- ☑ Were alternative project architectures/high level designs considered?
- ☑ Is there documented rationale for the selected project architecture/high level design?
- ☑ Are all interfaces identified and documented?
- ☑ Have industry standards been identified for the high-level design?
- ☑ Is the design clearly documented?
- ☑ Is the high-level design traceable to the system requirements?
- ☑ Do any of the requirements need to be changed based on the high-level design development effort?
- ☑ Have the integration, verification, and deployment plans been updated in the SEMP?

Detailed Design

- ☑ Did each component have a technical review?
- ☑ Did each component design trace to a sub-system requirement?
- ☑ Were all sub-system requirements satisfied by the component design activity?
- ☑ Was a verification plan for each component defined?
- ☑ Was each component design checked for performance?
- ☑ Was the component design documentation complete, up to date, and accurate?
- ☑ Was a critical design review conducted?
- ☑ Was an alternatives analysis done on the products used in the system?
- ☑ Have all system and sub-system requirements been updated at the time of the critical design review?

3.3.8 Software and Hardware Implementation

OBJECTIVES	• Develop and/or purchase hardware and software that meet the design specifications with minimum defects. Identify any exceptions to the requirements or design specifications that are required. • This step in the process develops [builds or constructs] the hardware and software for the system that matches the requirements and component level detailed design documentation. This step is primarily the responsibility of the development team, who fabricates the hardware and writes the software programs. The systems engineering activities includes the support and review of the development effort on behalf of the system's owner. • If multiple developments for the same system are underway, the systems engineering activity includes the monitoring and coordination of these developments to ensure these projects integrate together with a minimum of effort.
DESCRIPTION	The systems engineering activities include the monitoring and coordination of the hardware & software development activities. The implementation is primarily the responsibility of the implementation team, whether it is in-house or by a contracted development firm. Monitoring is accomplished by a preplanned series of reviews coordinated with the development team. This is performed by the systems engineering staff of the agency or a contracted system manager. It is essential to review the technical progress and provide technical guidance on the implementation of requirements. These reviews provide early warning that requirements are deficient, or they are not being met by the implementation. In such cases deviations or waivers may be needed or the re-evaluation of the requirement may be necessary. Also, these reviews will be needed when coordinating among concurrent developments for the same project, depending on the development strategy.
CONTEXT	
INPUT *Sources of Information*	• Component Level Detailed Design is the "build-to" documentation. The coding and fabrication team develop their products based on this documentation.

	• Commercial products are procured for the project. The intent is to wait until the last possible opportunity to procure technology to get the latest and most cost-effective products • System and Sub-system Verification Plans are used to assist the development team to fully understand the design and requirements they are building to.
PROCESS *Key Activities*	• Plan software/hardware development • Establish development environment • Develop detailed design for each component • Develop custom hardware/software • Perform unit/device testing • Develop support products (manuals, training, on-line help)
OUTPUT *Process Results*	• Developed hardware and software • Support products • Unit Verification Procedures

Overview

In the Implementation step, hardware and software solutions are created for the components identified in the high-level design. Part of the solution may require custom development of hardware or software, and part of the solution may be implemented with off-the-shelf items, modified as needed to meet the design specifications. The components are tested and delivered ready for integration and installation.

Having invested the effort in developing a clear set of requirements and a good high-level design, the systems engineering process now provides technical oversight as an implementation team of hardware and software specialists create the detailed component-level design, fabricate the hardware, and write the software programs. This is a highly iterative process, particularly for software, where key features may be incrementally designed, built, tested, and incorporated into the baseline over time. Progress is monitored through a planned series of walkthroughs, inspections, and reviews.

Although the systems engineering approach does not specify the mechanics of hardware and software development (this is left to the implementation team), the software/hardware development effort is obviously critical to project success. This is the time to build quality into the hardware/software and minimize defects. A common refrain in the software industry is that you can't test quality into the software, you must build in quality from the beginning. The systems engineering activities that are suggested in this section are intended to ensure that the implementation team builds quality into their products.

In practice, most of the hardware that is used for ITS projects is commercially available. Software development is more prevalent, but there are many ITS projects that include little or no software development. ITS projects that do not include custom hardware or software development acquire the necessary off the shelf hardware and software components at this step and configure and customize the components for the particular application. Good performance-based specifications are developed to support the acquisition, the system components are acquired, and bench testing is performed on the received components to verify that they meet their specifications. The detailed modeling and design development, hardware/software development, and unit testing described in this section are not required.

Although it isn't included in every ITS project, at least some custom software development is fairly common for ITS projects and custom software development has proven to be a relatively risky endeavor. This is why software development receives more attention than hardware development in this section. It is beyond the scope of this document to discuss specific software development techniques, but there are several clear factors that contribute to software development success:

No matter how clear and unambiguous the requirements appear, it is almost certain that the software customer and the software implementation team will interpret some of the requirements differently. Requirements walkthroughs in the previous steps help to mitigate this risk, but ultimately, the customer/stakeholders will have to monitor the software as it is being developed to ensure that the development is proceeding in the right direction. Also remember and try to compensate for the natural tendency of the software developer to interpret the specification narrowly and the customer to interpret it more broadly. Expect and plan for course corrections and requirements changes along the way, at least until we discover the way to build the "perfect specification". Ensure that the contract is flexible enough to have a couple of reviews but also to allow some visits or informal reviews with the developers to see how they are doing.

Perhaps the best way to reduce software development risk is to proceed in small steps and build incremental software releases in short time periods. Software cycle times that used to be measured in years are now measured in months or even weeks between incremental software releases. Incremental, iterative development with frequent coordination and feedback is the best way to keep the software development on track, particularly for projects where the requirements are not well understood at the outset.

Rapid prototyping should be used to help the user and developer visualize the user interface before significant resources are invested in software development. This is one area in particular where you can expect multiple iterations as the developers incrementally create and refine the user interface design based on user feedback. Prototyping is one of the best ways to design software that will be highly satisfactory to users, rather than software that merely meets the requirements. Rapid prototyping must be planned for and bounded like any other task.

Risks to be Managed

Hardware or Software doesn't satisfy design specs: The risk, which is primarily technical, in the implementation of the hardware and/or software is that they don't satisfy the design specifications allocated to them. The way this risk is managed is by testing the hardware components and software modules against the specifications allocated to the hardware components and/or software modules.

Activities

This step includes activities that the hardware and software specialists lead, beginning with the detailed design of each system component and ending with testing of each completed component. Systems engineering plays a supporting role in each of these steps, providing technical oversight on an on-going basis to identify minor issues early, before they grow into large problems. Some of the activities below will be performed only if there is new hardware or software development. Each of the activity descriptions is followed by a discussion of the technical review and monitoring of that activity.

Plan software/hardware development

This activity will be relevant if considerable hardware or software development is required. The implementation team documents their development process, best practices, and conventions that will be used. The plan should address development methods, documentation requirements, delivery stages, configuration control procedures, and technical tracking and control processes for the implementation effort, including reviews. This is one of the key documents that should be reviewed by the customer and the broader project team.

CAUTION

The implementation plan should be reviewed and approved before hardware/software development begins. Well qualified implementation teams will already have proven processes in place that can be tailored for the specific project, so this shouldn't be viewed as a burdensome activity. The intent is not to mandate a particular implementation process, but to ensure the implementation team has an established process that they will follow. Even teams that use "lightweight" processes such as agile software development or extreme programming should have documented processes. Lightweight, flexible processes can be very effective, but no documented process is definitely a red flag.

Establish development environment

This activity is applicable when new development, particularly software development is required. The development environment is assembled and integrated, including design and development tools, source control tools, third party application libraries, test simulators, etc. Every tool that is used should be documented specifically enough so that the development environment can be replicated if necessary.

TIP

Although it is sometimes overlooked, the development environment is just as critical to future software maintenance as the actual detailed design documentation and source code. Every tool that is used should be documented, including version information and complete documentation of any customization or extensions. If this is a custom development and you have paid for the tools, include the development environment as a project deliverable.

A peer review or inspection can be used to verify that the development environment is adequate and accurately documented. Once established, the development environment should be placed under configuration control so changes to the environment are tracked. Seemingly minor changes like application library upgrades or operating system service pack upgrades can cause problems later if the changes are not controlled and tracked.

Develop detailed hardware and software design specifications

Detailed "build to" design specifications are created for each hardware and software component to be developed. This step applies when there is significant hardware or software development as part of the project. A simple user interface prototype is developed as a quick way to help users visualize the software and several iterations are created based on user feedback. Any necessary requirements and high-level design changes are identified, evaluated, and incorporated as appropriate.

The detailed design of each component should be reviewed to verify that it meets the allocated requirements and is fit for the intended purpose. Periodic or as-needed reviews can be held to monitor progress and resolve any design issues. For larger projects, coordination meetings should be held to ensure concurrent design activities are coordinated to mitigate future integration risks. At the completion of the detailed design step, a broader stakeholder review is held to review and approve the detailed design before the implementation team begins to build the solution.

Procure commercial products

Off-the-shelf (OTS) products are compared and OTS solutions are selected, tailored as necessary, and procured. An alternatives analysis documents the alternatives that were considered and how the superior alternative was selected.

The evaluation of OTS products should be reviewed to verify that the evaluation criteria were properly defined, an appropriate range of products was considered, and the evaluation criteria were applied fairly.

TIP

Delay procurement of the OTS products until the products are actually required to support the implementation. Too much lead time can result in OTS that becomes outdated before it can be integrated into the project. Too little lead time could cause procurement delays that impact the project schedule.

Develop software and hardware

The software is written and the hardware is built based on the detailed design. On most projects, there is an easy transition from detailed design to software/hardware construction because the same person that does the detailed design for a specific part of the project also writes the software for that part. The current state of the practice is to develop the software incrementally and release the software in stages. The initial releases implement a few core features and subsequent releases add more features until all requirements are satisfied. This incremental approach enables early and on-going feedback between the customer and the implementation team. If this approach is used, then a staged delivery plan should define the order in which the software will be developed and the staged release process.

Releases will be developed, tested, and made available to selected users for feedback. Providing feedback on interim releases is only part of the technical oversight that should be performed. Code inspections and code walkthroughs should also be used to check the software quality – inspections and walkthroughs are the only way to check that the software is well structured, well documented, and consistently follows the coding standards and conventions identified in the implementation plan. Also verify that source control procedures and tools are in place to manage the evolving software and hardware components until they are delivered. Independent reviewers can help verify software quality on the customer's behalf if the customer agency does not have the right expertise.

Most project managers who have managed software development efforts are familiar with the "90% complete" syndrome in which software developers quickly reach "90% complete" status, but then the development effort languishes as the final 10% takes much more work than anticipated. Project tracking should be based on discrete measurable milestones. To minimize overly optimistic reporting of software completion status, credit for completed software should not be taken until the piece of code has been successfully tested and integrated into the next release.

Develop Supporting Products

Enabling products such as training materials, user manuals and on-line help, installation and conversion software, and maintenance manuals are also developed. It is natural to focus on the hardware and software in the "end product", but you also need to develop and account for all the ancillary products that are needed in a working system.

Like the end-product hardware and software components, the supporting products can also be developed in stages and released incrementally to encourage early customer feedback.

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 software developer, who should step through and test every line of code, including all exception and error paths. As the software/hardware is developed, a series of test cases are developed that will exercise the hardware/software component; these test cases are documented in a unit verification plan. After the software is complete and thoroughly debugged by the developer, the test cases are used to test the hardware/software and the test results are documented. Identified defects are analyzed and corrected and testing is repeated until all known defects are either fixed or otherwise resolved. Defect correction may be relatively simple or may include redesign of sections of code that are determined to be error-prone.

While the developer will conduct their own tests to identify and fix as many defects as possible, experience shows that the test cases and formal tests should be conducted by an independent party, either within the implementation team or an independent party from another organization. The reason for this independence is obvious if you look at the objectives of the software developer and the software tester. The primary objective for the tester is to break the software while the primary objective of the developer is the exact opposite – to make the software work. Few individuals can effectively wear both of these hats. The degree of independence between the developer and the tester and the level of formality in unit testing should be commensurate with the criticality of the software and the size of the project.

The unit verification plan should be reviewed to verify that it will thoroughly test the component. Track testing as it progresses to verify that defects are being identified and addressed properly. A testing process that identifies few defects could indicate excellent software or an incomplete or faulty testing process. Use scheduled technical reviews to understand the real project status. Various techniques can be used to estimate the number and severity of remaining defects and make an educated decision about when the software will be ready for release.

This step results in hardware and software components that are tested and ready for integration and verification. Artifacts of the development process, including the implementation plans, development environment documentation, component-level detailed design, unit verification plans and procedures and verification results, change control records, and supporting products and documentation are also delivered. A record of the technical reviews that were performed should also be included in the project documentation.

Tailoring This Step

Depending on the budget, staff resources, size, and complexity of the project or program, the number and formality of the reviews should be tailored to fit the project.

Small projects, e.g. signal system upgrades, may require only 1-2 technical reviews and the coordination meetings with communications and/or IT services only.

Large complex projects may require bi-weekly or monthly technical reviews [at a minimum], and an equal amount of coordination meetings.

The technical reviews should go in accordance with the planned reviews in the Systems Engineering Management Plan.

Policy or Standard for Process Step

The FHWA Regulation does not specifically mention general hardware/software practices to be followed. ISO/IEEE 12207 Software development life cycle processes.

Traceable Content

Table 8 illustrates that the test results of implementation should support the detailed design specifications for the project.

Table 8: Implementation Process traceability

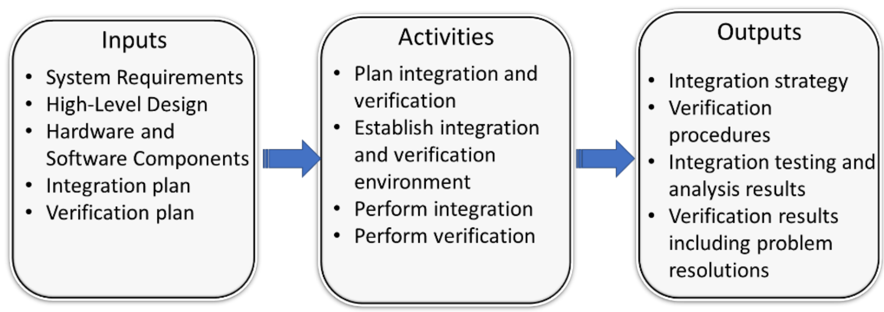
Traceable Artifacts	Backward Traceability To:	Forward Traceability To:
Implementation Test Results	High Level and Detail Design Specifications	N/A

Checklist

The following checklist can help answer the question “Are all the bases covered?” once the activities above have been planned or performed. **The checklist is not sufficient by itself.**

- ☑ Is the technical review and coordination meeting schedule established and documented?
- ☑ Has the development team established a schedule and method for measuring software and hardware progress?
- ☑ Have the significant risks been identified and is a schedule in place to monitor these risks?
- ☑ Does the development team have documented process for developing hardware, software, database, and communications?

3.3.9 Integration and System Verification

OBJECTIVES	• Integrate and verify the system in accordance with the high-level design, requirements, and verification plans and procedures. • Confirm that all interfaces have been correctly implemented • Confirm that all requirements and constraints have been satisfied.
DESCRIPTION	The software and hardware components are individually verified and then integrated to produce higher level assemblies or subsystems. These higher-level assemblies are also individually verified before being integrated with others to produce yet larger assemblies, until the complete system has been integrated and verified.
CONTEXT	
INPUT *Sources of Information*	• System Requirements • High-Level Design • Hardware and Software Components • Integration plan • Verification plan
PROCESS *Key Activities*	• Plan integration and verification • Perform integration • Perform verification
OUTPUT *Process Results*	• Integration strategy • Verification procedures • Integration testing and analysis results • Verification results including problem resolutions

Overview

In this step, we assemble the system components into a working system and verify that the system fulfills all of its requirements. Assembling a puzzle is a nice simple analogy for this step, but the challenge in an ITS project "puzzle" is that you may find that all of the pieces aren't available at the same time, some of the pieces will not fit together particularly well at first, and there will be pressure to change some of the pieces after you have already assembled them. The systems engineering approach provides a systematic process for integration and verification that addresses the challenges and complexity of assembling an ITS system.

Integration and verification is an iterative process in which the software and hardware components that make up the system are progressively combined and verified against the requirements as shown in **Figure 20**. This process continues until the entire system is integrated and verified against all of its requirements. This is the opposite of the decomposition that was performed during the requirements and design steps, which is reflected in the symmetry between the left and right sides of the Vee. Components that are identified and defined on the left side of the Vee are integrated and verified on the right.

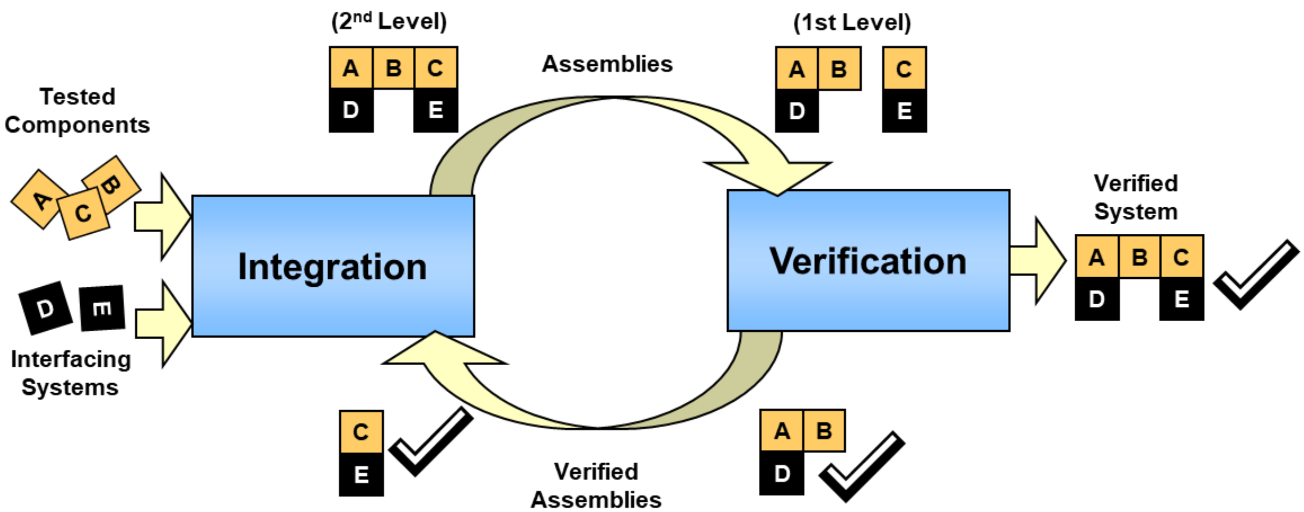


Figure 20: Iterative Integration and Verification
(Source: FHWA)

In systems engineering, we draw a distinction between “verification” and “validation”. “Verification” confirms that a product meets its specified requirements. “Validation” confirms that the product fulfills its intended use. In other words, verification ensures that you “built the product right” while validation ensures that you “built the right product”. This is an important distinction because there are lots of examples of well-engineered products that met all of their requirements, but ultimately failed to serve their intended purpose. For example, a Bus Rapid Transit system might implement a signal priority capability that satisfies all of its requirements. This system might not serve its intended purpose if the traffic network is chronically congested and the buses are never actually granted priority by the signal control system when they need it most. Verification is discussed in this section. System validation is described in Section 3.3.11.

Risks to be managed

System does not pass verification testing: The risk in this step, which is primarily technical, is that the system(s) developed for the project doesn’t function/perform as required in the expected environment.

Activities

Integrating and verifying the system is a key systems engineering activity that includes basic planning, preparation, and execution activities as described in the following paragraphs:

Plan integration and verification

Integration and verification planning actually began on the left side of the Vee. A technique for verifying every requirement was identified as the requirements were specified and a general plan for verifying all of the requirements was documented. As the overall structure of the system was defined as part of high-level design, the general strategy for integrating the system components was developed. Detail is added to these general plans based on the actual system implementation and the order in which project components and other required resources will be available is defined.

The integration strategy defines the order in which the project components are integrated with each other and with other systems that the project must interface to. Each integration step includes integration tests that verify the functionality of the integrated assembly with particular focus on the interfaces. For less complex projects, the integration strategy can be an informal plan. For complex projects, there will have to be careful planning so that the system is integrated in efficient, useful increments consistent with the master schedule.

The verification plan is expanded into verification procedures that define the step-by-step process that will be used to verify each component, subsystem, and system against its requirements. For efficiency, test cases are identified that each can be used to verify multiple requirements. Each test case includes a series of steps that will be performed, the expected outputs, and the requirements that will be verified by each step in the test case.

REGULATION

The systems engineering analysis requirements identified in FHWA 23 CFR 940.11 and FTA's National ITS Architecture Policy Section 6 require identification of testing procedures, which are synonymous with the verification procedures that are described here.

- Establish integration and verification environment – The tools that will be used to support integration and verification are defined, procured, and/or developed. For complex systems, this could include simulators that are used to simulate operational interfaces, test equipment that is used to inject failures and monitor system responses, etc. The verification environment effectively simulates the operational environment as faithfully as possible and allows portions of the system to be tested before all interfacing components are completed.

CAUTION

If test and simulation tools are used to support system verification, then these tools should be verified with the same care as the end product. Verifying a system using a simulator that has not been verified could result in invalid verification results or compensating errors where a defect in the end product is masked by a defect in the verification tool.

Perform integration

The system is progressively integrated based on the high-level design and the integration strategy. The system 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 the interface documentation in detail. The process confirms that all interfaces are implemented per the documentation. Proposed changes to the baseline high-level design, including any required changes to the interface documentation, are identified.

Perform verification

Every requirement is verified using the test cases defined in the verification procedures. System requirements and the related subsystem and component-level requirements may be verified several times as verification progresses bottoms-up from component verification to subsystem verification to system-level verification. For example, a requirement that the system "shall blank a selected dynamic message sign on user command" might be verified at several different levels. The capability of the sign to blank itself would be verified at the Dynamic Message Sign (DMS) component level. The capability of the user interface to accept and relay a "blank sign command" might be tested at the subsystem level, and finally, an end-to-end system test would be used to verify that the sign actually blanks on user command.

There are four basic techniques that are used to verify each requirement:

Test**:** Direct measurement of system operation. Defined inputs are provided and outputs are measured to verify requirements have been met. Typically, a "test" includes some level of instrumentation Tests are more prevalent in early verification tests where component-level capabilities are being exercised and verified.

Demonstration: Witness system operation in the expected or simulated environment without need for measurement data. For example, a requirement that an alarm is issued under certain conditions could be verified through demonstration. Demonstrations are more prevalent in system level verification where the complete system is available to demonstrate end-to-end operational capabilities.

Inspection: Direct observation of requirements that are easily observed such as construction features, workmanship, dimensions and other physical characteristics, software language used, etc.

Analysis**:** Verification using logical, mathematical, or graphical techniques. Analysis is frequently used where verification by test would not be feasible or would be prohibitively expensive. For example, a requirement that a web site support up to 1,000 simultaneous users would normally be verified through analysis.

As each verification test case is performed, all actions and system responses are recorded. All unexpected responses are documented and analyzed to determine the reason for the unexpected response and define a plan of action that might involve repeating the test, revising the test case, fixing the system, or even changing the requirement. Any changes to the test cases, the requirements, or the system are managed through the configuration management process.

It is important to keep strict configuration control over the system components and documentation as you proceed through verification. The configuration of each component and the test case version should be verified and duly noted as part of the verification results. It is human nature to want to quickly find and fix a problem "on the spot", but it is very easy to lose configuration control when you jump in to make a quick fix. In addition, such quick fixes can invalidate preceding verification tests since a change may have unexpected effects on another part of the system.

TIP

Resist the temptation to scale back verification activities due to budget or schedule constraints. This would be false economy because defects that slip through will be even more expensive to fix later in the system lifecycle. As previously noted, it is most efficient to identify defects early in the verification process. This approach also minimizes the number of issues that will be identified during system verification, which is the most formal and most scrutinized verification step. Issues that occur during a formal system verification that is witnessed by stakeholders can undermine confidence in the system. Be sure to run the system verification test cases beforehand to the extent possible to reduce the risk of unexpected issues during a formal system verification.

Lessons Learned: As you approach the end of the effort, it is a good idea to consider documenting Lessons Learned on the project. It is always a good practice to maintain a "Lessons Learned" memo or report which may be used by future systems engineering staff that are developing new but similar systems. The lessons learned document captures for each stage of the SE process, which decisions or approaches were unexpectedly valuable (or useless). These lessons learned can be invaluable to future system owners and the systems engineers that work to develop, operate, and maintain their systems.

Integration and verification result in a documentation trail that shows the integration and verification activities that were performed and the results of those activities. The outputs include:

Integration strategy

The integration strategy defines the sequence of steps that were performed to incrementally integrate the system. It also defines the integration tests that were performed to test the interfaces in detail and generally test the functionality of the assembly. Typically, the integration tests are less formalized and step by step test procedures will not be documented for each test.

Verification plan and procedures

The verification plan documents the approach that was used for verifying each of the system and subsystem requirements. The plan identifies test cases that were used to verify each requirement and general processes that were used to conduct test cases and deal with verification issues. Verification procedures elaborate each test case and specify the step-by-step actions and expected responses.

Integration test and analysis results

This is a record of the integration tests that were actually conducted, including analysis and disposition of any identified anomalies.

Verification results

This is a summary of the verification results. It should provide evidence that the system/subsystem/component meets the requirements and identify any corrective actions that were recommended or taken due to the verification process.

Tailoring This Step

There are a number of factors which make a project complex. The same factors that influence other steps in the systems engineering process also influence the integration process.

Integration of sub-systems with external interfaces is nearly always required.

The major impact on tailoring the integration process is the degree of formality needed to verify compliance with requirements to stakeholders. The simpler the system, the smaller the project team and the fewer the number of external stakeholders [stakeholders with systems that interface with the target system], the less formal the integration process needs to be.

Some level of verification is needed to accept the system. The formality with which verification is performed can be tailored to the budget, size, and complexity of the project. For a small simple project with few stakeholders, it only may be necessary to use the requirement document itself as a checklist and extemporize the procedures on the fly. Thus, no verification documents are needed. The system's owner determines what level for verification formality and documentation is needed to satisfy the complexity of the project.

Policy or Standard for Process Step

The FHWA Regulation does not specifically mention integration as one of the required systems engineering analysis activities. EIA 731 and the INCOSE SE Handbook have identified best practices for integration.

The FHWA Regulation does not specifically mention general verification of requirements. It does require inter-operability tests relating to use of ITS standards. IEEE std. 1012 talks about independent verification and validation. The INCOSE SE Handbook identifies best practices.

Traceable Content

Table 9 illustrates that the test results of integration and verification should support the system requirements for the project.

Table 9: Integration and Verification traceability

Traceable Artifacts	Backward Traceability To:	Forward Traceability To:
Verification Test Results	Functional, Performance, and Environmental Requirements	N/A

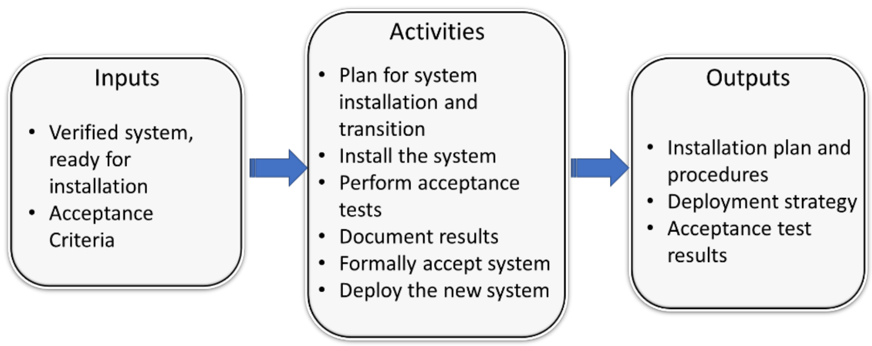
Checklist

The following checklist can help answer the question "Are all the bases covered?" once the activities above have been planned or performed. **The checklist is not sufficient by itself.**

- ☑ Are integration activities included in the master project schedule?
- ☑ Does the plan for integration and verification support the strategy for deployment?
- ☑ Based on project complexity, is a written Integration Plan required?
- ☑ Are the external systems needed to support integration available, or does the interface need to be simulated?

- ☑ Have the components to be integrated been placed under configuration control?
- ☑ Are the development teams available to promptly fix problems uncovered during integration?
- ☑ Was a Verification Plan developed and approved?
- ☑ Were all requirements traced to a Verification Plan test case?
- ☑ Were Verification Procedures developed and approved?
- ☑ Were the key participants identified and trained?
- ☑ Were all resources needed for testing in-place?
- ☑ Were all participants notified of the testing schedule?
- ☑ Was a Verification Report prepared?

3.3.10 Deployment and Acceptance

OBJECTIVES	• Successful installation • Operations team ready to operate and maintain the system • Uneventful deployment of the new system
DESCRIPTION	The system is installed in the operational environment and the system is transferred from the project development team to the organization that will own and operate the system. The transfer also includes support equipment, documentation, operator training, and other enabling products that support on-going system operation and maintenance. Acceptance tests are performed as part of this step to confirm that the system performs as intended in the operational environment before control is transferred.
CONTEXT	
INPUT *Sources of Information*	• Verified system, ready for installation • Acceptance Criteria
PROCESS *Key Activities*	• Plan for system installation and deployment • Install the system • Perform acceptance tests • Document results • Formally accept system • Deploy the new system
OUTPUT *Process Results*	• Installation plan and procedures • Deployment strategy • Acceptance results

Overview

Deployment and Acceptance involves releasing the deployed system to its users. The users could be operations, maintenance, facilities, or another function at the "owner" agency. At this point, the development of the system is complete; the development team has integrated and verified they system against the requirements. While the focus shifts to the team that will operate and maintain the operational system, the development team will often have a continuing role supporting system operation through a warranty period and the development team may also play a role in validating the system in operation. Optional elements of the deployment process might include training (and delivering training materials) for specific stakeholder groups regarding the operation and/or maintenance of the new system.

Up to this point, the system may have been tested primarily in a lab environment. In this step, the system is shipped to the actual deployment site(s), installed, accepted, and transitioned to system operations and maintenance (O&M), as shown in Figure 21. Note that the nature of deployment and acceptance can vary with the type of project. For example, cloud-based software projects will have little effort in delivery, site prep, or installation. Projects that contract O&M may not have the transition between distinct teams as shown in the figure. Projects that contract separate system integrator and systems engineering contracts may have more than one "development team" to coordinate with.

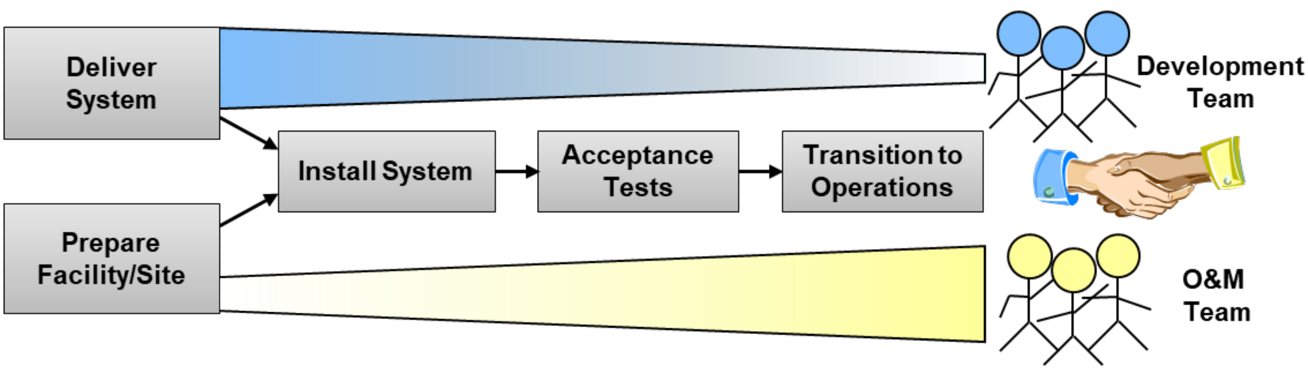


Figure 21: Transition from Development Team to O&M Team
(Source: FHWA)

Larger systems may be installed in stages. For example, a closed-circuit television (CCTV) camera network may be built out incrementally over the course of several years and several projects. This may be done to spread the costs across several fiscal years or to synchronize with other construction projects in the region. In other cases, phased deployment may be performed to mitigate risk by deploying the essential core of the system and then adding features over time. If it is necessary to deploy the system in stages, whether due to funding constraints, to mitigate risk, or to synchronize with other projects, it is important to understand the dependencies between successive deployments and to prioritize the projects accordingly.

Risks to be managed

System does not successfully become operational: The risk to be managed in the deployment process is that the system is operable and maintainable by the receiving entity staffs.

Activities

This step represents the handoff of the tested system from the project team to the operations and maintenance team in the field. The following tasks are cooperatively performed to deliver, install, and deploy the system to full operational status:

Plan for system installation and deployment

Key to the systems engineering process is the advance planning, and this is especially true for delivery and installation since the system may actually change hands from the engineering team to the system owner. The first step is to create a deployment plan that clearly defines how the site will be prepared and how system will be installed, tested, and transitioned to operational status. The plan should include the validation criteria; that is, how are you going to know that the system is operating correctly? It is a good idea to include a series of checklists in the deployment plan that identify all the key pieces that must be in place and working prior to switching over to full operation. If there are still any open issues found during system test (and there likely will be), evaluate each of them to determine whether or not they should be fixed or a work-around created prior to placing the system into full operation. A formal review of the deployment plan should be held, and include the deployer, the operations team, and other key personnel.

TIP

The Deployment Plan should take into consideration the complexity of the system, whether it will be deployed at multiple sites, and, if so, the order of the deployments. It might be a good idea to bring up a minimal configuration or a single installation at first and to add further functionality and other sites once the initial installation is operational.

There are many war stories about the delivery of a system that doesn't quite fit the installation site (e.g. server racks that wouldn't fit through the equipment room door). For this reason, part of the planning process is to perform a site survey (physical, electrical, communications, and lighting) and possibly prepare a site survey report and site installation plan. There might be some modifications required to the site or facility in order to accommodate the system, or perhaps additional seating for personnel to operate the system. You should document any necessary site modifications in a site plan, execute the plan, and make sure the facility is ready to receive the system.

If the new system is replacing an existing system, a smooth transition should be planned, including a backup strategy to revert to the existing system just in case the new system does not operate as intended.

Install the System

The system must be physically moved from the development and test labs to the actual deployment site(s). In preparation for this, a complete set of documentation will be developed by the engineering team and coordinated with the site O&M team. This documentation will include all the logistical details for transporting the hardware and software, any facility modifications that may be necessary, personnel assignments for installation, and installation instructions.

Prior to system installation, the deployment sites must be prepared, the system must be installed at each site and tested, and operations and maintenance staff must be trained. A deployment plan should address each of these steps.

Until delivery, the system's components – the hardware and software – were inventoried and under version control by the engineering team. Once delivered, however, ownership may change hands to the agency who will operate and maintain the system. Regardless, the engineering and operating agencies should come to agreement regarding who will maintain the inventory, the version of the software and hardware, any vendor maintenance agreements, and maintenance records.

When the system is installed at each deployment site, the operations and maintenance team should perform an initial inspection and preliminarily accept the system. This could be a formal review of the hardware/software inventory, a check of the documentation, or perhaps a start-up test. More extensive formal acceptance tests will be conducted once the system is fully installed.

Following delivery of the system to a site that has been properly prepared and modified as necessary, the system will be installed. Sometimes, problems occur during system installation – make sure you've included a contingency for backing out all or part of the installed system in your installation plan. Following installation, installation tests should be run to verify the system was installed correctly using documented test procedures, also included in the installation plan. You could consider including the system operators in the installation tests since they'll be objective and will get a chance to learn more about the system.

Perform Acceptance Tests

Once the systems is installed, formal acceptance tests should be run by the customer agency It's a good idea to tie some funding to a successful outcome.

The acceptance tests to be performed should be clearly documented in advance and agreed to by all parties. Detailed test procedures should be defined for each test to be performed. The acceptance test documentation should clearly define expected results for each test and cover what should be done in event of a test failure.

TIP

The team that will routinely operate the system should participate in Acceptance Testing. Ideally, they should perform most or all of the tests because developers are vulnerable to "Designer's Bias" (e.g., they will never mis-interpret a system display, or hit a wrong key, etc.). If the O&M team is to perform the acceptance tests, they should be trained on the new system in advance.

Once the system has been initially deployed, acceptance testing will be performed to show that the system meets all the acceptance criteria defined during the Requirements step. In order to do this, acceptance test plan and procedures may need to be defined and once tests are completed, the results documented. The acceptance test plan could build upon the acceptance criteria defined earlier.

Transition to Operations

After the system has been installed successfully at the final deployment site and accepted, the next step is to transition to full operation. If this is a new, standalone system, this can be a relatively uncomplicated effort. However, if the system must interoperate with other systems – such as the case when installing new AVL software on an existing computer-aided dispatch system – additional integration and testing may be necessary. Or perhaps the new system is replacing an existing system – perhaps you are replacing an older signal control system – careful deployment planning must take place to minimize the disruption to ongoing signal operations.

TIP

When transitioning to operations, especially when replacing an existing system, a contingency back-out plan should be included as part of the deployment plan so that in the event the new system does not operate correctly, you can revert to the older system until the issues have been sorted out.

All operations and maintenance staff should be in place and properly trained. The maintenance plans for the system should be reviewed by the operations and maintenance team – check to make sure that all maintenance procedures and hardware/software maintenance records are in place, and that they are adequate to properly maintain the system.

The operational procedures and any special equipment needed to operate or monitor the system should be ready, tested, and operating correctly. It's a good idea to take some performance measurements on the system at this stage so that you can estimate performance following transition to full operational status. Establish user accounts (if necessary), initialize databases or files as identified in the transition plan, and make sure all test data has been removed or erased. The system should be all set to begin operations.

TIP

Some transitions to full operation can be complex, especially when replacing an existing system that many people use. Just as we get annoyed when we can't access the Internet for a few hours, users may also get annoyed if your system is down for any period of time. You might want to consider planning the transition on a weekend or in the evening, if possible, to cause the least disruption to system users. Also consider holding a "dry-run" so that everyone knows their role during the transition period and performs their assigned task to make the transition as smooth as possible. If the public may be impacted during the transition, notify them in advance and during using all available tools (media releases, website notifications, social-media postings) and be sure to monitor public responses. If it's a new system, consider a "soft-launch" strategy.

Finally, a deployment readiness review meeting should be held with the operations and maintenance team, the support personnel who are on-hand to address last-minute issues, and representatives from other interfacing systems, the project sponsor, and other key personnel. Use the checklist in the deployment plan to assess the system readiness. Only after all checklist items have been declared as "ready" should the go-ahead be given for the system to transition to full operational status.

Following transition, the team will ramp down to include only the operations and maintenance personnel with potential continued support from the development team if there is a warranty period. It might be advisable to keep a few support personnel around through the validation period so that any issues that come up in the early stages can be resolved quickly.

The primary output of this step is a completely installed product or system in a facility or site, modified as needed to meet the requirements of the product or system, and transitioned to operational status. To support this effort, the following outputs should be generated:

- A hardware and software inventory, under configuration control, including versioning information, maintenance records and plans, and other property management information
- Delivery and installation plan, including shipping notices
- Acceptance test plan and procedures
- Deployment plan with checklists
- Contingency plans
- Test issues and resolutions
- Trained O&M personnel
- Operational and maintenance procedures

Tailoring This Step

Depending on various factors of the project, deployment can range from very simple to very complex. The number of deployment steps and the number of stakeholders involved in deployment are the best indicators of complexity, although there may be others of equal importance. If either of these factors warrant, then project management may decide that the expense of preparing, reviewing, and approving a Deployment Plan document is justified. For simple projects, the guidance in the PMP and in the SEMP, plus a qualified person in charge of deployment, may be sufficient.

Policy or Standard for Process Step

The FHWA Regulation (23 CFR940.11) does not specifically mention initial system deployment as one of the required systems engineering analysis activities.

Traceable Content

Acceptance documentation that shows all system requirements have been met and the O&M team is prepared to support operation and maintenance of the new system in the expected operational environment.

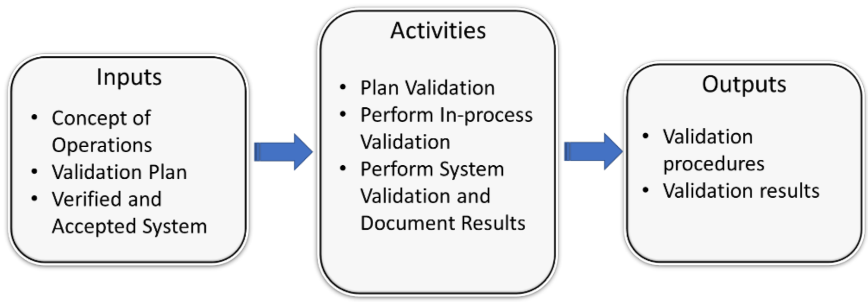
From a systems engineering perspective, there will be traceability or testing artifacts of this process for the system acceptance. The key artifact or deliverable of this process will be results of the acceptance testing following which there will likely be a written agreement stating that the ownership, responsibility for operation and maintenance of the system has transitioned from the development team to the system owner, subject to warranty periods and ongoing maintenance requirements.

Checklist

The following checklist can help answer the question “Are all the bases covered?” once the activities above have been planned or performed. **The checklist is not sufficient by itself.**

- ☑ Has a comprehensive set of deployment goals been developed?
- ☑ Can those deployment goals be traced into the deployment strategy?
- ☑ Does the deployment strategy consider available funding?
- ☑ Does each step in the deployment strategy produce an operationally useful and maintainable deployed system?
- ☑ Does the deployment strategy minimize the risk of interference to on-going operations?
- ☑ Does the deployment strategy offer a viable operational fallback at each step of the process?
- ☑ Are all stakeholders in a deployment step aware of their roles and responsibilities?
- ☑ Are all resources needed for a deployment step available?
- ☑ Has a work-around plan been developed in case a needed resource is not available?
- ☑ Has acceptance testing been defined based on the acceptance criteria?

3.3.11 Validation

<table>
<tr><td>OBJECTIVES</td><td>Confirm that the installed system meets the user’s needs and is effective in meeting its intended purpose.</td></tr>
<tr><td>DESCRIPTION</td><td>System validation is an assessment of the operational system. Validation ensures the system meets the intended purpose and needs of system’s owner and stakeholders. In addition, validation refers to the actions taken at each step in the development process to ensure that the outputs of the step are validated by the stakeholders.</td></tr>
<tr><td>CONTEXT</td><td>
</td></tr>
<tr><td>INPUT
Sources of Information</td><td>• Concept of Operations
• Validation Plan
• Verified and accepted system</td></tr>
<tr><td>PROCESS
Key Activities</td><td>• Plan validation
• Perform in-process validation
• Perform system validation and document results</td></tr>
<tr><td>OUTPUT
Process Results</td><td>• Validation procedures
• Validation results</td></tr>
</table>

Overview

Validation is the confirmation that the need(s) identified in the Concept of Operations have been met by the new operational system. Recall that needs are different from requirements in that requirements identify the correct operation and performance of the system in the specified environment, and needs identify the expected impact of the new system on the environment.

For example, an off-board fare collection system will have requirements regarding accepting payment from travelers and dispensing payment instruments with associated stored value or trip contracts. That same off-board fare collection system may have the need to reduce the dwell time of a bus at the bus stop (compared to, for example, an on-board fare collection system that the new system is replacing). Verification testing involves verifying the correct operations of the system to collect fares and dispense tickets/contracts in the expected environment. Validation is testing that in fact the bus dwell time (e.g. average or median time stopped at a bus stop normalized by passenger volume at the bus stop) has in fact been reduced by the expected amount.

Validation really can't be completed until the system is in its operational environment and is being used by the real users. For example, validation of a new signal control system can't really be completed until the new system is in place and we can see how effectively it controls traffic.

Of course, the last thing we want to find is that we've built the wrong system just as it is becoming operational. This is why the systems engineering approach seeks to validate the products that lead up to the final operational system to maximize the chances of a successful system validation at the end of the project. This approach is called in-process validation and is described in the activities section.

Risks to be managed

System does not meet a key need: The risk being managed is that the system has the desired impact on the environment into which it is deployed, i.e. it meets the needs.

Activities

Validation is testing that the needs documented in the Concept of Operations have been satisfied by the deployment of the new system. The validation process has two primary activities:

Plan Validation

With stakeholder involvement planning starts at the beginning of the project timeline. The plan includes who will be involved, what will be validated, what is the schedule for validation, and where the validation will take place. An additional aspect of planning is the definition of the validation strategy. This defines how the validation will take place and what resources will be needed. For example, whether a before and/or an after study will be needed. If so, the before study will need to be done prior to deployment of the system.

An initial Validation Plan was created with the Concept of Operations earlier in the life cycle (see Section 3.3.5). The performance measures identified in the Concept of Operations forced early consideration and agreement on how system performance and project success would be measured.

It is important to think about the desired outcomes and how they will be measured early in the process because some measures may require data collection before the system is operational to support "before and after" studies. For example, if the desired outcome of the project is an improvement in incident response times, then data must be collected before the system is installed to measure existing response times. This "before" data is then compared with data collected after the system is operational to estimate the impact of the new system. Even with "before" data, determining how much of the difference between "before" and "after" data is actually attributable to the new system is a significant challenge because there are many other factors involved. Without "before" data, validation of these types of performance improvements is impossible.

In addition to objective performance measures, the system validation may also measure how satisfied the users are with the system. This can be assessed directly using surveys, interviews, in-process reviews, and direct observation. Other metrics that are related to system performance and user satisfaction can also be monitored, including defect rates, requests for help, and system reliability. Don't forget the maintenance aspects of the system during validation – it may be helpful to validate that the maintenance group's needs are being met as they maintain the system.

Detailed validation procedures may also be developed that provide step-by-step instructions on how support for specific user needs will be validated. At the other end of the spectrum, the system validation could be a set time period when data collection is performed during normal operations. This is really the system owner's decision – the system validation can be as formal and as structured as desired. The benefit of detailed validation procedures is that the validation will be repeatable and well documented. The drawback is that a carefully scripted sequence may not accurately reflect "intended use" of the system.

In-Process Validation

One of the key principles of systems engineering is stakeholder involvement in the development process. This is why the systems engineering approach seeks to engage stakeholders to validate the products that lead up to the final operational system to maximize the chances of a successful system validation at the end of the project. This approach is called in-process validation and is shown in Figure 22. As depicted in the figure, validation was performed on an ongoing basis throughout the process:

- The business case for the project was documented and validated by senior decision makers during the initial project identification and scoping.
- User needs were documented and validated by the stakeholders during the Concept of Operations development, i.e., "Are these the right needs?"
- Stakeholder and system requirements were developed and validated by the stakeholders, i.e., "Do these requirements accurately reflect your needs?"
- As the system was designed and the software was created, key aspects of the implementation were validated by the users. Particular emphasis was placed on validating the user interface design since it has a strong influence on user satisfaction.

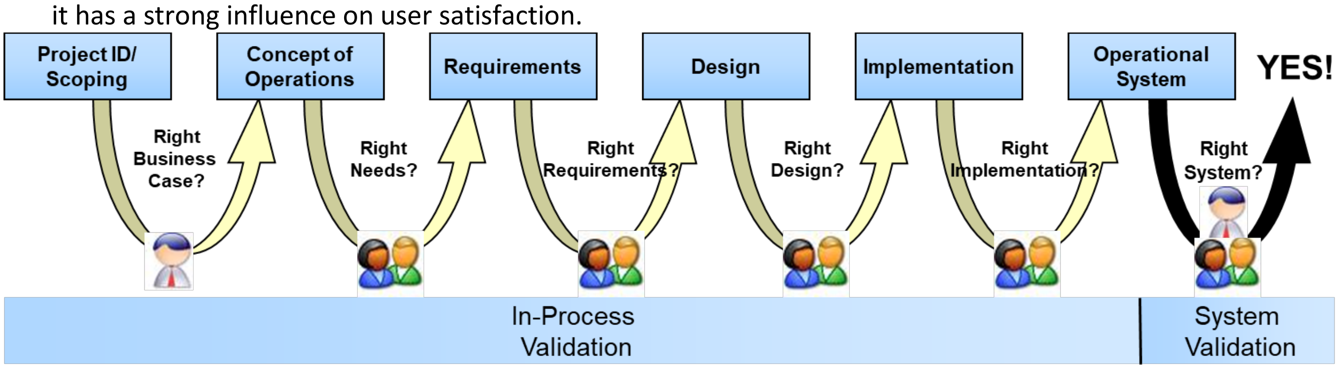


Figure 22: In-Process Validation Reduces Risk
(Source: FHWA)

Since validation was performed along the way, there should be fewer surprises during the final system validation that is discussed in this step. The system will have already been designed to meet the user's expectations, and the user's expectations will have been set to match the delivered system.

Perform system validation and document results

The system is validated according to the Validation Plan. The system owner and system users actually conduct the system validation. The validation activities are documented and the resulting data, including system performance measures, are collected. If validation procedures are used, then the as-run procedures should also be documented.

TIP

The measurement of system performance should not stop after the validation period. Continuing performance measurement will enable you to determine when the system becomes less effective. The desired performance measures should be reflected in the system requirements so that these measures are collected as a part of normal system operation as much as possible. Similarly, the mechanisms that are used to gauge user satisfaction with the system (e.g., surveys) should be used periodically to monitor user satisfaction as familiarity with the system increases and expectations change.

The data resulting from the system validation is analyzed, and a validation report is prepared that indicates where needs were met and where deficiencies were identified. Deficiencies can result in recommended enhancements or changes to the existing system that can be implemented in a future upgrade or maintenance release. If an evolutionary development approach is used, the validation results can be a key driver for the next release of the product.

Tailoring this Step

There is great latitude in system validation. It is dependent on institutional agreements (State and FHWA requirements) on a per project basis. In signal upgrade systems a simple before and after study on selected intersections may be sufficient to validate. In a more complex system, a number of evaluations may be needed. This validation may be needed for each stakeholder element, each sub-system [e.g., camera, CMS, and detection system]. It may be done on a sample area of the system or comprehensively. Getting this addressed with the stakeholders in the planning stage is very important.

Policy or standard for Process Step

The FHWA Regulation does not specifically mention general validation practices to be followed. IEEE-1012 Independent verification and validation and CMMI identify best practices.

System validation should result in a document trail that includes the Validation Plan, Verification Procedures (if written), and the Validation Results including disposition for identified deficiencies. There are several industry and government standard outlines for validation plans including FIPS Publication 101 and IEEE Standard 1012. Note that both of these standards cover both verification and validation plans with a single outline. Consider maximizing commonality between the verification and validation documentation in your project for efficiency.

Traceable Content

Table 10 shows the Validation process artifacts and their backward traceability to the need(s) in the Concept of Operations document.

Table 10: Validation traceability

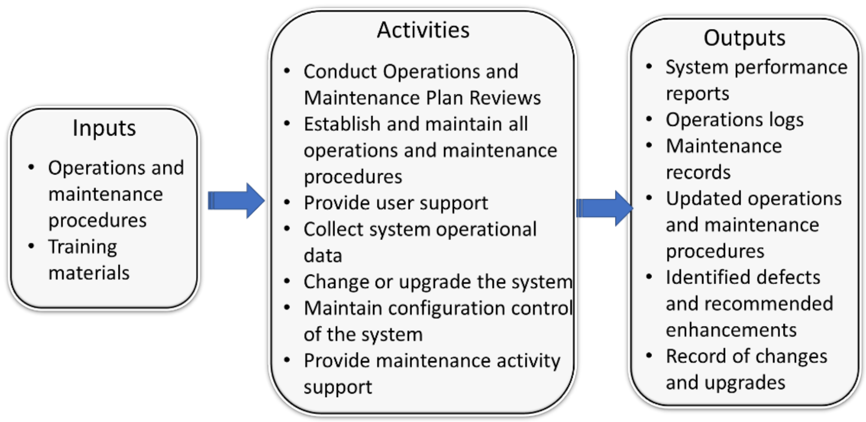
Traceable Artifacts	Backward Traceability To:	Forward Traceability To:
Validation Test Plan	Needs in the Concept of Operations	N/A
Validation Test Results	Needs in the Concept of Operations	Recommended System Changes, Enhancements, Upgrades

Checklist

The following checklist can help answer the question "Are all the bases covered?" once the activities above have been planned or performed. **The checklist is not sufficient by itself.**

- ☑ Were all the needs clearly documented?
- ☑ With each need, goal, and objective is there an outcome that can be measured?
- ☑ Are all the stakeholders involved in the validation planning and the definition of the validation strategy
- ☑ Are all the stakeholders involved in the in-process validation of the systems engineering products?
- ☑ Are all the stakeholders involved in the performance of the validation and is there an agreement on the planned outcomes?
- ☑ Are there adequate resources to complete the validation?

3.3.12 Operations and Maintenance

OBJECTIVES	Use and maintain the system over the course of its operational life.
DESCRIPTION	Once the customer has accepted the ITS system, the system operates in its typical steady state. System maintenance is routinely performed and performance measures are monitored. As issues, suggested improvements, and technology refreshes are identified, they are documented, considered for addition to the system baseline, and incorporated. An abbreviated version of the systems engineering process is used to evaluate and implement each change. This occurs for each change or upgrade until the ITS system reaches its end-of-life
CONTEXT	
INPUT *Sources of Information*	• Operations and maintenance procedures • Training materials
PROCESS *Key Activities*	• Conduct Operations and Maintenance Plan Reviews • Establish and maintain all operations and maintenance procedures • Provide user support • Collect system operational data • Change or upgrade the system • Maintain configuration control of the system • Provide maintenance activity support
OUTPUT *Process Results*	• System performance reports • Operations logs • Maintenance records • Updated operations and maintenance procedures • Identified defects and recommended enhancements • Record of changes and upgrades

Overview

Now that the ITS system is up and running, it enters a “steady state” period which lasts until the system is retired or replaced. The operations and maintenance personnel have all been trained on the system and are ready to perform their duties. During this period, operators, maintainers, and users of the system may identify system issues, suggest enhancements, or identify potential efficiencies. New releases of hardware and software will be installed and routine maintenance will be performed. Approved changes and upgrades are incorporated into the system baseline using the systems engineering process, as shown in Figure 23. Operations and maintenance personnel might also identify process changes that may streamline operations or maintenance activities. All changes to the processes should be documented.

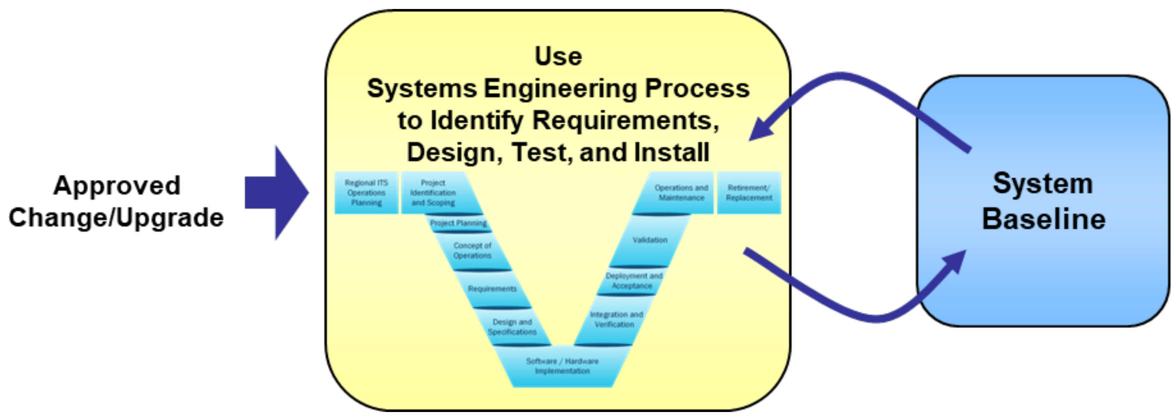


Figure 23: Changes/Upgrades Performed Using Systems Engineering
(Source: FHWA)

The operation process involves the collection of data to support continuous or periodic retesting of the system against the system requirements (System Verification) and against the needs (System Validation) to confirm that the system continues to operate as expected and to have the impact on the environment as expected. If at any time the system fails to meet all system requirements or all needs at the level it did during system verification and system validation, then the system should be fixed or adjusted, or an understanding developed as to what has changed to cause the failed (or partially failed) verification and/or validation (see next section on system Maintenance Process).

If staff operating or maintaining the system changes during operations, then the operation or maintenance training used during the transition process may be reused to get the new staff up to speed on the system operation or maintenance processes.

Risks to be Managed

The risks to be managed are that the system fails to work as required, or that the expected impact on the environment in which the system is operating has changed in some way.

Activities

Operations and maintenance do not contain major milestones, although any changes incorporated certainly will, since they will follow the systems engineering process. Therefore, the key activities below are performed periodically, at a pre-determined interval unless a change is considered severe and affects system performance dramatically.

Conduct Operations and Maintenance Plan reviews

Operations and maintenance personnel and the system sponsor should all be in agreement on the level of support to be provided. This could include the staffing profile, frequency of technology refreshes (e.g., how often the software or hardware are upgraded to a new release), performance monitoring and reporting, processes for handling identified issues, level of support provided to the end-user, and so forth.

Establish and maintain all operations and maintenance procedures

Although the processes to be used for identifying, tracking, resolving, recording, and providing feedback on all system issues will have been established during the Initial Deployment step, specific detailed procedures will be further developed and maintained as efficiencies are identified. All personnel will be trained on the procedures and are responsible for their use.

Provide user support

End users of your system, whether they are traffic management center operators or a person whose farecard is not working in the new farecard reader, need to be able to contact someone for user support. This support could be handled by a formal call center or perhaps only a person who performs the task during spare time via e-mail, depending on the type and complexity of the system to be supported. Either way, the user support personnel should be properly trained, should document all calls from initiation through final resolution, and should have access to system experts if needed. These user support personnel should also provide periodic updates on user inquiries and resolutions.

TIP

A trouble ticket system or database that holds information about all user support inquiries can help you to review the types of calls that were received and to notice trends. If there seems to be a recurring problem or confusion about some aspect of the system, it could mean that a system modification should be considered.

Collect system operational data

During earlier phases in the system life cycle, you will have determined how to collect system performance metrics and will have used the performance data to validate the system. During operations, you should collect sufficient performance data to help you determine how well the system is operating over time. For example, in a transit management center, the on-time arrival performance data might be collected from the AVL software. If you are providing a website that displays incident and speed information, a positive response from a user who is asked whether the information was "helpful and accurate" could be collected. Feedback from operators and travelers will provide a measure of customer satisfaction. In-process reviews can be held periodically to review collected metrics, assess system performance, and identify potential system improvements. Some of the outputs that can be created with system operational data are:

- System performance reports both from any installed automated performance monitors and from user support calls received.
- Operations logs identifying the various operations actions undertaken
- The current system configuration, including hardware, software, and operational information.
- A complete record of all system changes performed with version information

Change or upgrade the system

The system will evolve over its lifetime. This could be as a result of issues reported by the users, system improvements identified from the review of operational data, or upgrades to the system. If you decided to deploy only part of the system during the Initial Deployment step, this is when you'll incrementally add the rest of the system – whether it's additional functionality or equipment at additional sites (e.g., additional CCTV deployment).

All proposed changes will be prioritized, and will require careful cost estimates, schedules, planning, testing and coordination with operations and maintenance prior to installation. Each approved change will require a new system release level and should be coordinated between the operations and maintenance and development teams.

Each potential change to the system should be assessed by the affected stakeholders and the project sponsor to determine whether or not it should be incorporated. You should clearly understand and document the effect each change will have on other parts of the system, on the operation of the system as a whole, and on the maintenance of the system before approving the change. If you make this assessment early on by following the systems engineering process for the change, you will not discover a problem months later in the lab when schedule and budget impact is significantly higher.

You should use the systems engineering process, from requirements through design and verification and installation to add any approved change to the system. Approved changes are typically aggregated into builds or releases, although you may want to introduce particularly complex changes individually.

Each build or release should be subjected to thorough verification testing prior to installation. There are many stories of "changes that affected only a few lines of code" that ultimately resulted in operational failure. Regression tests are also important that verify that a seemingly minor change in one part of the system didn't have an unexpected effect on another part of the system. Statements like "I didn't change that area so there is no need to test it." should be a red flag.

In many cases, the development and test lab that was available during the initial system development may not be available once the system has been deployed. (It might even be the system that was deployed!). Therefore, it's common to establish a test environment for the Operations and maintenance to test software product upgrades or minor fixes without interfering with the current operational system.

Some of the outputs that can be created as part of the system upgrades are:

- Updated operations and maintenance procedures
- Identified defects and recommended enhancements

- Record of changes and upgrades

Maintain configuration control of the system

The deployed system is under configuration control, so every time the system changes, even if only a minor software patch was added, the system baseline must be updated. This means that all documentation, databases, and any other operational data must also be updated.

This is one area where state of the practice lags a bit in ITS. It is common for agencies to require good configuration management practices by their contractors during system development, but then lose configuration control after the system is delivered. For example, if you want to know the configuration of a field controller at a particular location, you will have to take a trip to the field and have a look inside the cabinet at many agencies.

Provide maintenance activity support

A fully functional system should be available for use at all times except for minimal prescheduled maintenance periods during off-hours. Maintenance records on all equipment should be documented. Sufficient equipment, materials, supplies, and spares should be in place, inventoried, and working properly. The suggested quantities for each of these items should be included in the maintenance plan, prior to transitioning to full operational status.

TOOL

Consider using a database tool or a similar property management application to help you keep track of all equipment, together with maintenance records, date due for next maintenance activity, and so forth. Check it weekly and schedule the maintenance required. A key output of this activity is are the Maintenance record.

Tailoring

Operations & maintenance are necessary for all systems of any size or complexity. After the ITS system is built, it is made operational and maintained in operational condition for as long as is needed. However, some systems, such as traffic signals, operate autonomously with little routine human input. They need only initial configuration and periodic review and fine-tuning of the settings. Others, such as a closed-circuit television system, require hands-on involvement by a human operator as part of normal operation. But a traffic signal system may involve more intensive maintenance than a CCTV system.

The Operations & Maintenance Plan and associated documents, such as manuals, operating procedures, and system configuration records, should record all the information needed for employees to keep the system operating effectively and for managers to plan for future resource needs. Information provided should include what is needed for day-to-day activities, and also what is needed to plan for occasional activities, such as periodic preventive maintenance and system upgrades. The Concept of Operations, System Requirements, and design documents should be consulted as a checklist of all the system elements and operational aspects that may need coverage in operations & maintenance documentation.

Policy or Standard for Process Step

The FHWA Regulation requires that the identification of procedures and resources necessary for operations & maintenance of the system be determined in the systems engineering analysis for ITS projects funded with Federal money from the Highway Trust Fund, including the Mass Transit Account.

Traceable Content

Table 11 shows how routine operations records trace back to system requirements relevant to system operations.

Table 11: Operations Traceability

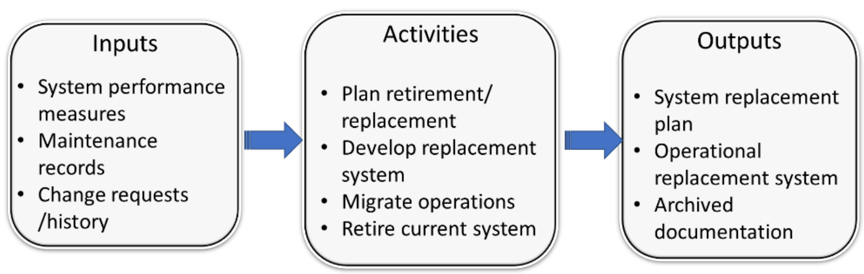
Traceable Artifacts	Backward Traceability To:	Forward Traceability To:
Operations Records (Status of each operations activity.)	System operation requirements	N/A
Maintenance Records (Status of each maintenance activity and status of each repair activity.)	System maintenance requirements	N/A

Checklist

The following checklist can help answer the question “Are all the bases covered?” once the activities above have been planned or performed. **The checklist is not sufficient by itself.**

- ☑ Is management support in place for on-going operations & maintenance (O&M)?
- ☑ Has funding for O&M been identified?
- ☑ Has an O&M Plan been developed and approved?
- ☑ Were all key stakeholders involved in development of the O&M Plan?
- ☑ Are resources and training in place for system start-up?
- ☑ Are established procedures for continually monitoring the effectiveness of operations & maintenance developed and approved?
- ☑ Is there a plan for long term upgrades?

3.3.13 Retirement/Replacement

OBJECTIVES	• Determine when a system needs to be retired or replaced. • Migrate to the replacement system with minimum disruption. • Remove the system from operation, gracefully terminating or transitioning its service. • Dispose of the retired system properly
DESCRIPTION	Eventually, almost every system will face retirement or replacement, no matter how well it was developed and maintained. This step in the process describes how to determine the end of life for a system. The objective is to make this end of life a planned event so that a replacement system can be procured if necessary and the preparations can be made so the system retirement is graceful with minimum stakeholder impact. When a system or subsystem needs to be replaced, a strategy must be developed to migrate to the new system gracefully, without interruption of service. Following successful migration, the old system is deactivated and disposed of.
CONTEXT	
INPUT *Sources of Information*	• System performance measures • Maintenance records • Change requests/Change history
PROCESS *Key Activities*	• Plan system retirement/replacement • Develop/procure the replacement system • Migrate operations to replacement system • Retire and dispose of current system
OUTPUT *Process Results*	• System replacement plan • Operational replacement system • Archived documentation for current system

Overview

Systems are retired and removed from service for a variety of reasons:

- High cost of operations & maintenance
- High cost of upgrades and changes due to system limitations
- Technology obsolescence makes the system unsupportable.
- Capabilities of the system are no longer needed

In all cases except the last, the system retirement must be planned well in advance so that a replacement system can be developed and deployed to meet continuing stakeholder needs with minimum disruption. When a replacement system is in place, transition can begin from the system to be retired to the replacement system. For critical systems that require very high availability, some period of parallel operation may be required during a transition period.

Regardless of the reason for the retirement of the system, you should make sure that everything is "wrapped up" properly (e.g., hardware and software inventory identified for disposal is audited, final software images are captured, and documentation is archived, etc.), the contract is closed properly, and the disposal of the system is planned and executed.

Risks to be managed:

<u>Negative operational impacts due to retirement/replacement</u>*:* The risk managed is that the system retirement has a negative impact on operational effectiveness, possibly impacting stakeholders, budget, and schedule.

Activities

This step represents the end of the system life cycle – the retirement and/or replacement of the ITS system. Characteristic of the whole systems engineering process is the planning of all events, and system retirement should be planned as well.

Plan system retirement/replacement

As the current system is operated and maintained, key data is collected that will inform a timely decision to consider and plan for retirement/replacement of the current system. Operational performance of the system (Is it still meeting stakeholder needs?, What is system availability?, How is user satisfaction?), change requests in the queue (Can the system meet new and developing stakeholder needs?), and maintenance data (How is system reliability?, Are adequate replacement parts available? Are maintenance costs increasing over time?) are collected and used to inform the decision to retire or replace the system.

Perform Gap Analysis: Current system capabilities versus capabilities needed. The trade studies process (see Section 3.4.5) can be used to evaluate the cost/benefit of upgrading the current system with replacement of the entire system or some major subsystem[s]. Can the current system evolve to meet the new needs? Is the technology that was used in the current system obsolete and no longer supportable? Have the operations and maintenance costs increased to the point where a replacement system is more cost effective? To answer the last question, the trade study should include life cycle cost analysis, including the operations and maintenance costs of the current system, and replacement costs. For comparison, the life cycle cost of the proposed replacement system is estimated. Will the replacement system have an improved cost/benefit ratio in operations & maintenance cost over its life? The replacement system should work better and cost less to maintain if the current system is truly approaching the end of its useful life. Other issues to consider include vendor support of commercial products embedded in the current system and license costs. The trade study should also consider the quality of the current system. For example, is the cost of documenting the current system prohibitively expensive, if documentation is not adequate?

Develop the replacement/retirement strategy. If this trade study analysis supports system replacement, a strategy for system replacement is defined. This strategy may require the upgrade of facilities, floor space, air conditioning, communications, furniture, and other such facilities. Because some systems are safety critical, they have to be operational full-time. In this case, the new system would need to be deployed in parallel with the current system. In this case, a switch-over plan needs to be created to allow the legacy system to act as a back-up while the new system is being verified and validated. There is a cost and deployment impact from having both systems operate in parallel for that period of time. In other cases, where the system is not safety critical, removing the current system prior to the deployment of the new system may be more cost effective.

Develop/procure the replacement system

The same systems engineering process described in the previous sections is used to develop the replacement system. This is another good opportunity to capitalize on the systems engineering documentation that was developed for the current system. The current system documentation provides an important input to the replacement system documentation, since many of the same needs and requirements will likely carry over to the replacement system. When defining the replacement system, it is also important to consider any lessons learned experienced in operating the current system. The goal is to learn from issues encountered with the existing system so that the replacement system will not suffer from the same issues. Following the systems engineering process, improved systems engineering documents that leverage the current system documentation will be developed for the replacement system:

- New Concept of Operations-
- Requirements
- Design documentation
- Verification plans
- Support documentation on development, training, maintenance, and user manuals

Migrate operations to replacement system

As discussed in section 3.3.10, the replacement system is deployed and formally accepted, and operations transitions to the replacement system. Depending on the criticality of the system being replaced, a period of parallel operation may be required, as defined in the replacement strategy.

Retire and dispose of current system

Planning for system retirement includes development of a disposal plan, which should include a complete inventory of all software and hardware, final system and documentation configurations, and other information that captures the final operational status of the system. This should include identification of ownership so that owners can be given the option to keep their equipment and use it elsewhere. It should also include how the system and documentation will be disposed, including an assessment and plan if special security measures should be in place or if there are environmental concerns that might dictate where the equipment should be disposed. You should also plan to erase the content of all storage devices to protect any personal data that might pose privacy concerns. The disposal plan should be reviewed and approved by all parties, including the agency or contractor providing O&M, the owner of the system (if different), and other key personnel.

The next activity is to deactivate the system. execute the disposal plan and record the results. It's also a good idea to hold a "lessons learned" meeting, including suggested system improvements. All recommendations should be archived for reference in future system disposals. The O&M contract should be officially closed out if one exists.

Tailoring This Step

The replacement strategy can be tailored for the project but factors that constrain this will be if the current system is critical and needs to be operational nearly full time. Are there alternatives to the legacy system that can allow it to be inoperable until the new system is in place, verified, validated, and operational?

As with other steps, the amount of formal documentation that is necessary can be much reduced for simple systems. For example, if the current system is a phone app and the replacement system is a new and improved phone app, "disposal" might simply entail uninstalling the old application.

Policy or Standard for Process Step

The FHWA Regulation does not place specific requirements on this step, but if, as part of the replacement effort another systems engineering set of steps are initiated, then the requirements for those steps would apply.

Traceable Content

Table 12 shows how disposal records trace back to system requirements relevant to system disposal.

Table 12: Disposal Traceability

Traceable Artifacts	Backward Traceability To:	Forward Traceability To:
Disposal Records (Status of each disposal activity.)	System disposal requirements	N/A

Checklist

The following checklist can help answer the question "Are all the bases covered?" once the activities above have been planned or performed. **The checklist is not sufficient by itself.**

- ☑ Was a trade study done on the cost/benefit of upgrading the legacy system against the cost/benefit of developing or procuring a new system?
- ☑ Did the trade studies include the operations & maintenance costs of the current and replacement system?

- ☑ Is the replacement system well documented? Does it have a concept of operations, requirements, and documentation necessary to support operations and maintanence.
- ☑ Is there a replacement strategy to switch out the current system with the replacement?
- ☑ Have all of the affected stakeholders been involved in the replacement/retirement decision, and the planning and replacement strategy for the new system?

3.4 Cross-Cutting Processes

Several important processes support more than one step in the system lifecycle. These processes provide continuity and support the incremental development process steps covered in the previous section.

3.4.1 Project Management

Introduction - Project Management and SE

Project Management is defined by the Project Management Institute (PMI) as "the use of specific knowledge, skills, tools, and techniques to deliver something of value to people". This is accomplished through the application of project management processes throughout the project's life. These processes are defined within five process groups covering the beginning to conclusion of a project. This section will describe the basic processes of project management across the five process groups defined by PMI.

Initiation

The Project Management Body of Knowledge (PMBOK) defines the project initiation group as "the process of formally recognizing that a new project exists or that an existing project should continue into its next phase." The Initiation group of project management processes authorizes the project to start, and provides the charge to the project team. This includes determining the vision for the project, document what the project hopes to accomplish, and secure approvals from the organization to develop the project. The primary output of this process group is a Project Charter which provides a high-level motivation and definition of the project. The Charter may also serve as an authorization of the project. ITS projects are typically initiated (programmed) through transportation planning activities such as the Transportation Improvement Program (TIP), or the statewide equivalent (STIP). A Project Charter can build upon the limited project information usually found in the TIP/STIP to provide an initial scoping of the project or for multi-phase projects, this process can be used to validate or refine the decisions made during the previous phases. The Project Charter as defined in the PMI Body of Knowledge (BOK) addresses the following topics related to the project:

- Define the purpose, goals and objectives of the project
- Define Project Scope and Deliverables
- Define High level Needs for the product or service
- Define Project High Level Budget, Schedule
- Identify Sponsor, Key Stakeholders, Responsibilities
- Project Approvals

Planning

Once the project initiation is complete and the project has the green light to go ahead, the project planning processes can begin. Project planning starts with the project's goals and objectives as defined by the Project Charter, the regional ITS architecture, and the needs and constraints elicited from the project's stakeholders. It identifies all relevant agency policies and procedures used in managing and executing such a project. It uses these to identify the project tasks [both administrative and technical], their interdependencies, estimates of needed resources, and budget for each task, the project schedule and the project's risks. The result of this planning is the Project Management Plan. This plan identifies the detailed work plans for both the administrative and technical tasks. The plan estimates the resources [people, equipment, and facilities.] needed for each task along with an estimated budget for each task. It identifies key events and the technical and program milestones, and establishes a schedule for the project. Each task's detailed work plan is developed to identify its needed inputs and outputs and a description of the process used to carry out the activity. Based on project complexity, additional technical plans [e.g., a Systems Engineering Management Plan] and additional administrative plans [e.g., Configuration Management, Risk Management and Procurement] may be included as part of the PMP, or defined as separate outputs. A template for the PMP is described in Section 6.1.

The Planning processes also include specific activities relevant to the systems engineering aspects of a project which are documented in a Systems Engineering Management Plan (SEMP). SEMP is the top-level plan for managing the systems engineering effort to produce a final operational system from initial requirements. It can be used in conjunction with a Project Management Plan which defines how the overall project will be executed, to define how the engineering portion of the project will be executed and controlled. It describes how the efforts of system designers, test engineers, and other engineering and technical disciplines will be integrated, monitored, and controlled during the complete life cycle. For a small project, the SEMP might be included as part of the Project Management Plan document, but for any project of greater size or complexity a separate document is recommended.

The information contained within a SEMP can be organized in different ways, but in general it should include an introductory section (including system description, top-level schedule, and relevant documents), technical planning and control, systems engineering processes tailored specifically for the project, and plans for coordinating the efforts of multiple engineering disciplines to accomplish the project tasks. Make sure that the SEMP and the PMP are consistent – it's fine to reference the PMP in the SEMP and vice versa. The following sections provide additional details on what would be included in the different main sections of a SEMP. In addition, a template for the SEMP is contained in Section 6.2

Technical Planning and Control – This section describes how the project will be controlled from a systems engineering point of view. It includes the engineering organization and responsibilities, identification of technical and performance monitoring reviews to be held during the project life cycle, the system test strategy, technical performance measurements to be monitored, the configuration and data management strategy, the risk management strategy, and identification of any critical items that may require special risk management. In addition, there are a host of other plans that should be created near the beginning of the project life cycle. Depending on the size and complexity of the project, plans may be small and included as part of the SEMP or they may be referenced by the SEMP as standalone documents. Minimally, the SEMP should identify all of the relevant project documents.

Plans to be described or referenced in the SEMP include:

- Interface Control Plan, describing the nature of external interfaces and responsibilities of organizations on each side of the interface.
- System Integration Plan, describing the strategy for how the software and hardware that comprise each subsystem will be integrated with other subsystems to form the overall system and including the dependencies and operational capabilities at each stage in the integration.
- System and Subsystem Verification Plan, describing how requirements will be verified for the system and each subsystem. This plan may also include the test lab environment(s), tools required, and dependencies.

- System Validation Plan, describing the approach that will be used to validate the project delivery.
- Software and Hardware Development Plans, describing the facilities, tools, and processes to be used to produce the project's software and hardware including development of custom software/hardware and procurement of commercial software/hardware products.
- Installation Plan, describing logistics for system deployment and installation procedures.
- Operations and Maintenance Plan, describing the organization, staffing, and processes for operating the deployed system, including maintenance, technical refresh plans, enhancement process, and procedures.
- Other plans, such as a Training Plan, a Safety Plan, or a Security Plan, may also be needed to address special issues of the project.

Systems Engineering Processes – This section of the SEMP describes the activities to be used for execution of the various systems engineering processes covered in Chapter 3, as tailored for your project. This is a good place to include a discussion of the project's approach to meeting the requirements of FHWA 23 CFR 940.11/FTA Policy Section VI. This section should include a definition of all high-risk areas, including critical technologies that might pose some challenge for your system. The SEMP will include a list of the tools that will be employed during the development (e.g., a requirements traceability tool).

Coordination of Engineering Disciplines – This section describes how the various inputs into the systems engineering effort will be integrated and how appropriate disciplines will be coordinated with that effort. In a complex project, there will be multiple engineering disciplines contributing to the success of the project. For example, for projects that have a user interface, operability/ human engineering will provide input during the development cycle to ensure that the design is user-friendly and intuitive. If system reliability is a major issue, specialists should assess the design to make sure that it will meet performance requirements. In the SEMP, the dependencies between these various engineering disciplines and the project life cycle will be documented. This will help the systems engineer to make sure that input is solicited from each engineering discipline at the appropriate time and that the right people are at the various technical reviews.

Preparation of the SEMP is a multi-step process that involves the system owner, systems engineer, and the Development Teams. First, the system's owner (or systems engineer) develops a framework for the SEMP before any process work starts. This includes the organizational structure, a master schedule for the system implementation, and identification of the technical tasks. For each task the SEMP framework identifies the required outputs, and to the extent possible at this stage, the inputs and processes to be performed. The SEMP framework may define a number of other items including a candidate set of supporting plans, metrics to measure technical performance, and the criteria for technical reviews. The SEMP framework will also tailor the technical processes commensurate with the scope and risk level of the project.

Then, the systems engineer and selected Project Development Teams, will take the SEMP framework and supply the needed detail for the processes to be used. This will include preparing any supporting plans, for instance, a Software Development Plan or an Interface Control Plan.

Execution

The Execution Process group defines the processes that should be performed to complete the work defined in the project management plan to satisfy the project specifications. This Process Group involves coordinating people and resources, as well as managing and performing the activities of the project in accordance with the PMP. Some of the key activities of the process group are:

- Direct and Manage the work performed per the PMP
- Acquire and Manage Project Team
- Conduct Procurements, if needed, by creating Procurement Documentation and getting vendors on-board to perform tasks per the PMP
- Manage communications, distribute information and manage stakeholder expectations

Monitoring and Control

Monitoring the progress of an engineering project requires a combination of management and engineering knowledge. The project plan requires an assessment of progress to adjust schedule, along with budget and staffing levels to meet project goals and customer requirements. The ability to assess the merit of reported progress entails knowledge of the underlying engineering efforts, with the possibility of related engineering discipline changing as the project progresses. A method to anticipate project progress is to examine not only concurrent measures of progress, but to consider leading indicators that identify potential issues prior to the project experiencing schedule slips. Table 13 shows representative potential monitoring approaches for phases of an ITS project. Other monitoring approaches may be appropriate depending on the specifics of the project.

Table 13 contains the following information in its columns:

- Phase: the phase of the SE process that is being monitored
- Acquired Service: key engineering activity during the phase
- Measure of Progress (MOP): quantifiable measures that can be assessed to give an indication of the progress in the phase
- Assessment: suggested method of monitoring the MOP
- Leading Indicator: additional quantifiable measures that can be an indicator of potential issues with maintaining the schedule, budget, and scope of the systems engineering effort.

Table 13: Progress Monitoring Alternatives

Phase	Acquired Service	Measure of Progress	Assessment	Leading Indicators
Project Conceptualization	• Stakeholder Engagement • Project Scoping	• Cost variance • Schedule variance • Level of consensus • Participation level	• Cost vs plan • Progress vs plan • Qualitative assessments	• # of meetings • # of participants • % budget change
SE Tailoring	• SE assessment • SEMP development	• Cost variance • Schedule variance • Analysis value • Correspondence to tailoring	• Cost vs plan • Progress vs plan • Qualitative assessments	• # of prior similar projects • % Schedule prior to draft
Concept of Operations	ConOps Development	• Cost variance • Schedule variance • Document completeness	• Cost vs plan • Progress vs plan • Qualitative assessments • Review comments	• # of prior similar projects % Schedule prior to draft
Requirements Development	Requirements Development	• Cost variance • Schedule variance • Document completeness • Correlation to Needs • Requirements quality	• Cost vs plan • Progress vs plan • Topics vs plan • Qualitative assessments • Review comments	• # of prior similar projects • % Schedule prior to draft
System Design	Design Development	• Cost variance • Schedule variance • Correlation to Requirements • Design quality	• Cost vs plan • Progress vs plan • Qualitative assessments • Review comments	• % of proven technology • % of custom products • # of requirements revisions
Component Design	Design Development	• Cost variance • Schedule variance • Correlation to Requirements • Design quality	• Cost vs plan • Progress vs plan • Qualitative assessments • Review comments	• % of proven technology • % of custom products • # of requirements revisions

Phase	Acquired Service	Measure of Progress	Assessment	Leading Indicators
Implementation	Implementation/installation	• Cost variance • Schedule variance • Test plan completeness • Component testing	• Cost vs plan • Progress vs plan • Qualitative assessments • Review comments	• # of design revisions/RFIs • % of custom products • Schedule variance
System Testing	System testing	• Cost variance • Schedule variance • Test plan completeness • System testing	• Cost vs plan • Progress vs plan • Qualitative assessments • Test results	• # of test plan comments • % of failed tests
System Evaluation	Evaluation testing	• Cost variance • Schedule variance • Evaluation plan completeness • System evaluation	• Cost vs plan • Progress vs plan • Qualitative assessment • Test results	• # of evaluator prior projects • Volume of 'before' data
System Retirement	Retirement planning	• Cost variance • Schedule variance • Plan adequacy	• Cost vs plan • Progress vs plan • Qualitative assessment	# of dedicated staff

Closing

The Closing process group includes processes performed to finalize all activities across all Project Management Process Groups to formally close the project or phase. Key among those processes is performing system acceptance and getting sign-offs. After review and acceptance of the project deliverables, the Project Manager obtains acceptance of the project. Another key activity of this process group is post-project or phase review. Examples of key deliverables for the post project review are the following:

- "Lessons learned" document that can be used to improve the agency's project management procedures
- List of potential future enhancements to the project's deliverables (e.g., a newly implemented center-based systems may trigger users to ask for more improvements)

Two additional activities that are needed for closing all projects are:

- Close out Procurements, ensuring financial and legal closure
- Archive project documentation, which includes collecting all final project documentation.

3.4.2 Configuration Management

Configuration management (CM) is one of the cross-cutting activities that occurs throughout the system life cycle. There are 2 principles involved: Establishing System Integrity which includes setting the baseline and Maintaining System Integrity through monitoring and managing changes to the produce baseline.

Introduction - Configuration Management

Configuration Management (CM) is a cross-cutting activity that isn't strictly a Systems Engineering (SE) activity but rather supports SE activities. CM can be thought of as a process for establishing and maintaining consistency of baselines, approving and controlling changes, and recording and reporting changes in status of a system/product under development.

CM recognizes that *Change Happens*. This is about establishing the baseline definition of the product and its documentation then managing the changes as they happen.

Description

Configuration management [CM], in conjunction with other systems engineering activities, is used to establish system integrity [integrity is defined as all system functionality, physical characteristics, and design match its documentation] and then maintain this integrity throughout its life.

The Electronic Industries Alliance (EIA) standard 649 defines CM as …

> *A management process for establishing and maintaining consistency of a product's performance, functional, and physical attributes with its requirements, design, and operational information throughout its life.*

The general CM process is demonstrated graphically below.

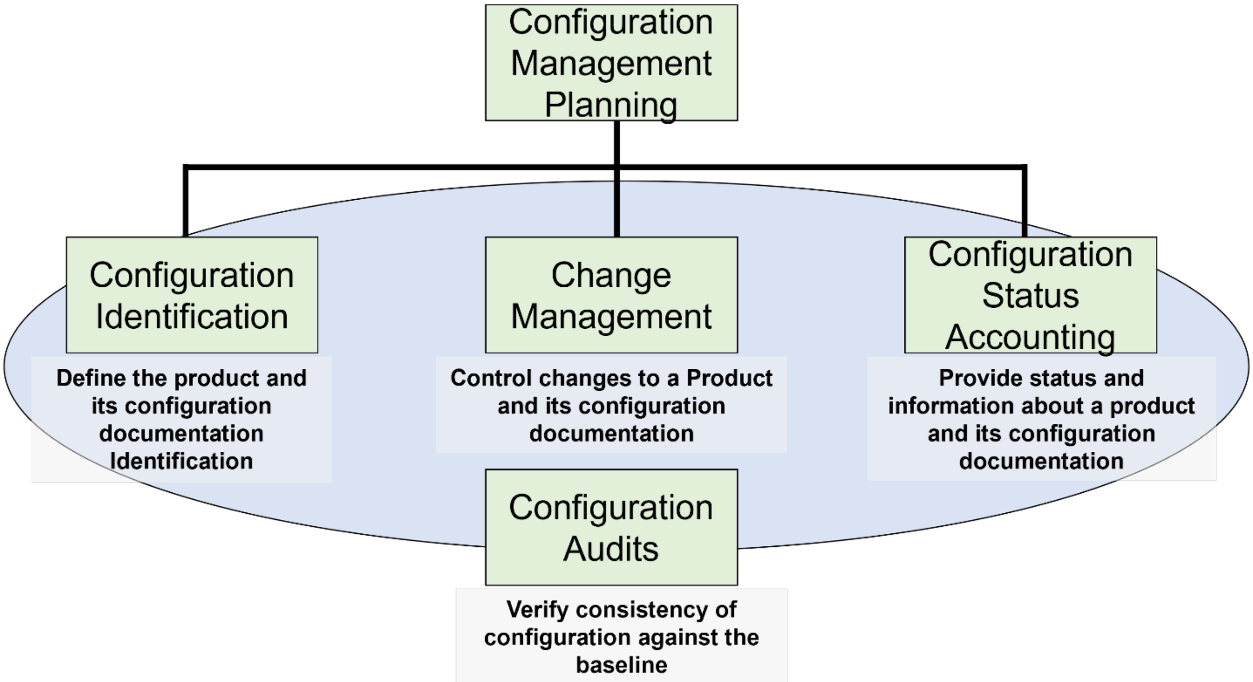


Figure 24: CM Process Overview
(Source: FHWA)

Configuration Management Planning

During Project Planning one of the activities should be to develop a plan for how Configuration Management will be carried out. A CM plan is a document that will guide the CM program of a particular group. Typical contents of a CM plan include items such as:

- Personnel
- Responsibilities
- Resources
- Training requirements
- Administrative meeting guidelines
- Definition of procedures
- Tools/tool use
- Organization configuration item (CI) activities
- Baselining
- Configuration control
- Configuration status accounting
- Naming conventions
- Audits and Reviews
- Subcontractor or vendor CM requirements

There are three application areas that need planning. The agency's CM plan for the life of the system, the implementation team's CM plan for development, and the CM Plan for the product vendors. The agency's CM plan should identify the requirements for the development team's CM plan and vendor's CM plan and the needed outputs to support the life of the system.

See Section 6.3 for a template for a Configuration Management Plan

Configuration Identification

Identify *what* needs to be independently identified, stored, tested, reviewed, used, changed, delivered and/or maintained. Identifying the configuration items and what identifiers will be used during the product life cycle. Configuration Identification defines the product and identifies its configuration documentation. What is it that makes up the baseline for the product?

> Baseline: An agreement at a given point in time that is under configuration control and used for measuring progress and as the basis for defining change. The term itself is meaningful only when preceded by another noun that specifies the type of agreement (e.g., schedule baseline, cost baseline, requirements baseline, etc.).

Configuration identification is the process of documenting and labeling the items in the system. Depending on the scale of the particular CM program, this simply may involve software versions or, in the case of a large program, all hardware, software, documentation, and the CM plan itself. The goal of configuration identification is to provide a unique identifier for each item to help track the changes to that item and to be able to understand its place in the system. Often, identification involves recording the identifier, maintenance history, relevant documents and other information that will simplify the change process in the future.

The benefits of configuration identification are to provide a way to uniquely identify the system components to support traceability and change management processes. This minimizes confusion over various versions of configuration items and facilitates the change control process. It allows items to be more easily tracked as they undergo change.

Change Management

With a product development underway or in use it is important to manage the changes to the baseline.

This is the process to manage changes to the configuration items. This involves a change management board and documentation that identifies the change, rationale, cost, risk, and priority.

The basics steps in a Change Management process are shown below:

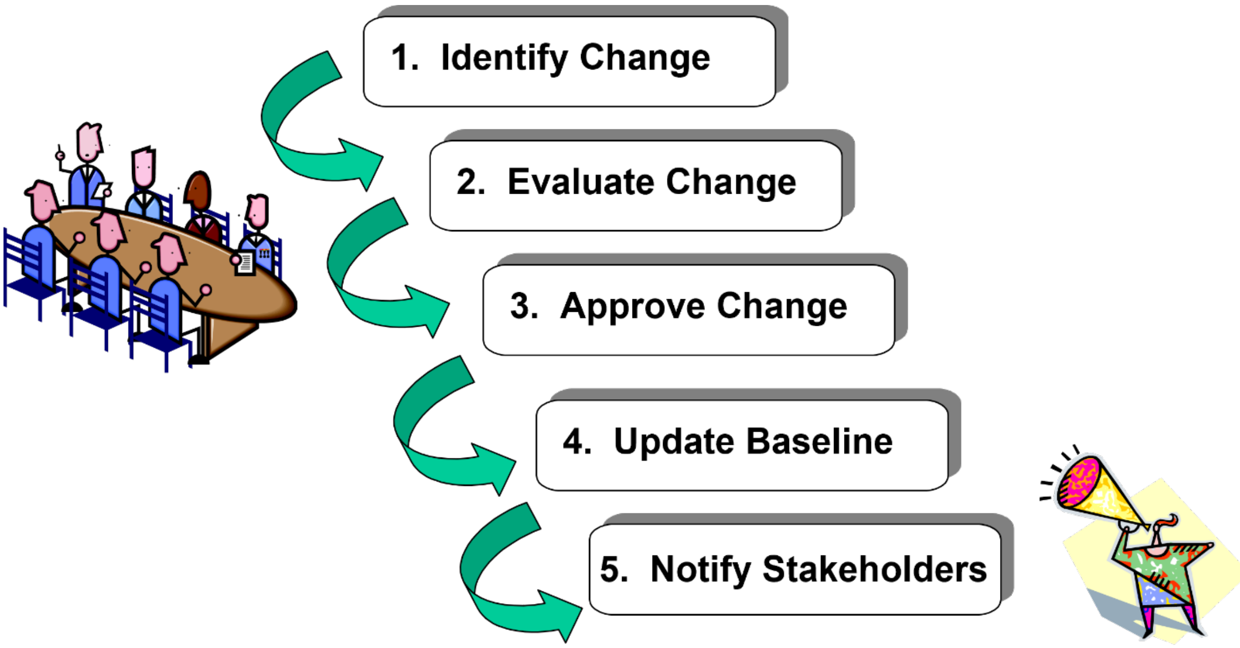


Figure 25: Change Management Process
(Source: FHWA)

1. Identify the Change - Step 1 is to identify the Changes. The CM Plan should establish who can suggest changes and how that I managed – either informally with the project staff or a formal. Also capture how those change requests will be documented – online forms, spreadsheet, database, etc.
2. Evaluate the Changes - This is more applicable to the Incremental Change approach – as changes are identified or requested someone needs to look at it and determine its impact. Include the person requesting the change but also include someone from the group that really understands the details of the architecture. Depending on the change, you may need to inform impacted stakeholders possibly even scheduling a mini-review between some of the stakeholders to discuss the potential change.
3. Approve the Change - Present one or more incremental changes to the Configuration Control Board or Maintenance Committee/Team for review - face-to-face, email, teams, etc. Approve/accept, reject, or defer the change and document the decision.
4. Update the Baseline - Now that the changes have been approved, they need to be rolled into the baseline products. If the same team isn't around to make the changes then the new team members will need access to the source files AND the knowledge to use the tools – may need to plan for this ahead of time. Make sure that you update the versions tracking all of the output's products.
5. Notify the Stakeholders - Last but not least, let your community of stakeholders know what's been done. This applies equally to incremental changes and full updates but you may want to tailor how much publicity is done for each type. This can be done in a number of ways or a combination of: email, press releases, presentation at committees, working groups, etc.

Effective communication of baseline changes to all affected parties is critical to effective configuration management. To that end, make sure your contact list is complete and current.

Configuration Status Accounting

Configuration Status Accounting is the process by which the project provides status and information about a product and its configuration documentation. Keeping track of updates as they occur.

Configuration Audits

The configuration audit is used to verify consistency of configuration against the baseline. Occasionally, a project team will perform an audit or have an outside group audit the records and products and compare them to the documented baseline. This may result in corrective actions or a need to update the CM Plan.

There are two types of audits, [functional and physical]. Functional audits match the product to the functional and performance requirements [acceptance verification]; and physical audits match version numbers and physical identifiers with the documentation.

When Is CM Done in a Project?

When is Configuration Management done? As shown below, the CM Plan will be developed during the initial stages of a project. It will be one of the plans developed, typically as part of the Project Management Plan or a Systems Engineering Management Plan.

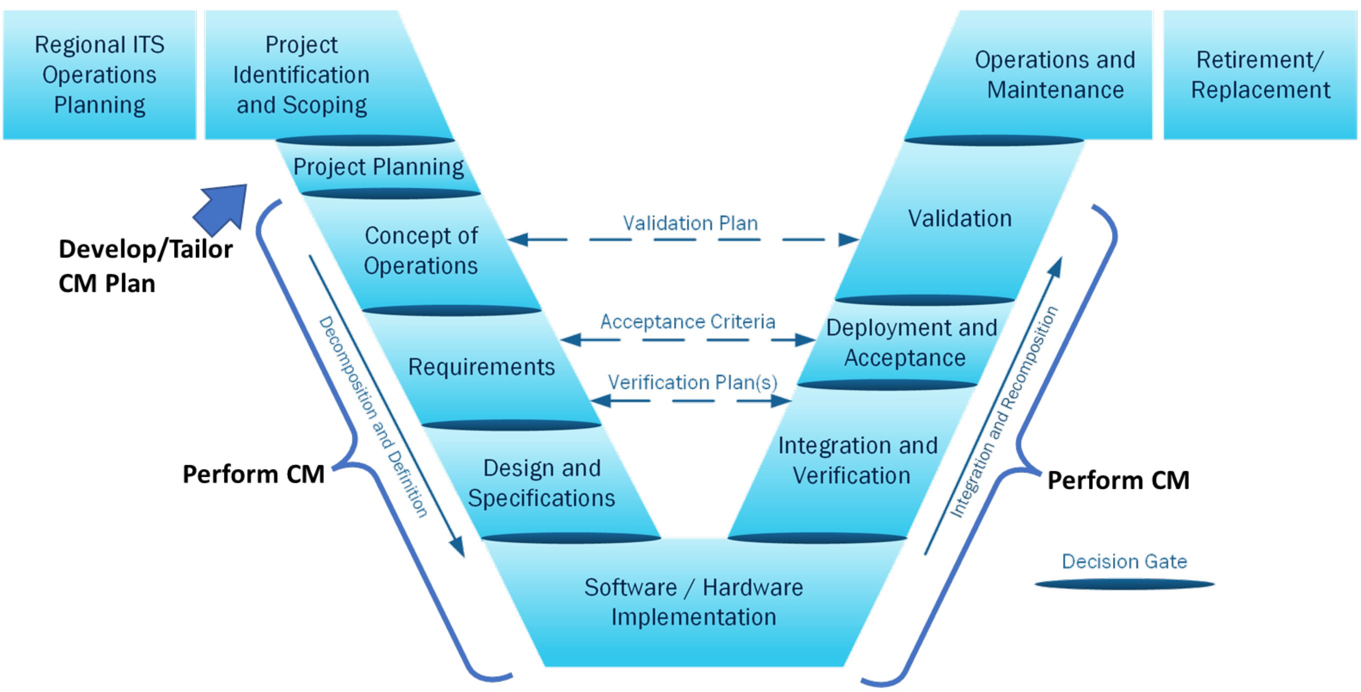


Figure 26. Where does the Configuration Management take place in the project timeline?
(Source: FHWA)

As the project development progresses, CM will be carried out against the documentation – the Needs in the ConOps, system requirements, the design specifications, and system architecture information all become part of the baseline and configuration management should be done throughout this part of the project. Then as the hardware and software are developed, purchased, installed, tested, and handed off any changes to the product baseline will also need to be managed.

Configuration Management Policy/Standards

Is there a policy or standard that includes Configuration Management?

FHWA Regulations do not specifically mention general Configuration Management practices to be followed. EIA 649 National Consensus Standard for Configuration Management provide a great deal of applicable information.

Metrics for CM to Reduce Project Risk

What should I track in this process step to reduce project risks and get what is expected? [Metrics]

On the *technical* side:

- Changes to the specific area of the system. A high number of changes may indicate a design weakness
- Monitor the impact of a change: who will be affected and how much of the system will need to be changed?

On the *project management* side:

- Growth in the number of change requests. This is an indication that the baseline was established too early
- Monitor the types of changes. Determine if the changes are critical to meet the initially stated requirements or if this is new functionality that can be deferred to the next phase of work

Are there any other recommendations that can help?

Configuration management for systems development is a management process for the project products.

TIP

Configuration management works together with a good systems engineering process. The systems engineering process provides the orderly establishment of the project products and documentation and Configuration Management is used to maintain consistency between the system changes to its documentation.

Checklist

The following checklist can help answer the question “Are all the bases covered?” once the activities above have been planned or performed. **The checklist is not sufficient by itself.**

- ☑ Is there a CM plan for the project?
- ☑ Was the plan reviewed and supported by all the stakeholders?
- ☑ Is the organization for CM in place for the project?
- ☑ Is there sufficient funding to sustain the CM activities throughout the life of the system?
- ☑ Does the development team have a CM process and was it reviewed by the system's owner and stakeholder?
- ☑ Is the product documentation complete to the extent that the system's owner can use another qualified development team to upgrade and maintain the system independent of the initial development team? [extremely important]

- ☑ Does the vendor have a CM process for their products?
- ☑ Does the vendor provide a notice of design changes?
- ☑ Does the vendor provide a notice of obsolescence?
- ☑ Does the vendor provide on-going maintenance support?

3.4.3 Traceability

Introduction – Traceability

Traceability is a cross-cutting activity that connects the various systems engineering activities with each other. Traceability is a key principle of systems engineering; it documents bi-directional relationships of project artifacts that allow backwards traceability to points of origin and forward traceability to the final system. The goal of traceability is to provide better quality and consistency of system/product development. It brings the ability to verify the history, location, and application of an item by means of documented identification.

A non-SE example of traceability is the food supply chain. If there is a problem with a food product, there is documented traceability back to the source of origin of that food product as it traversed the food supply chain. This may result in a food recall that goes beyond that particular food products creation.

Description – Traceability

Traceability follows the life of a requirement throughout the life of the system. The user needs are traced or related to the requirements which are traced or related to the design, which are traced or related to the implementation, which are traced or related to the testing, which are traced or related to the final system. The traceability is bi-directional; the initial items are traced forward to the latter items and conversely, the latter items are traced backward to the initial items. User needs and requirements are also traced to their associated validation and verification plans. Traceability is maintained after delivery of the system, supporting changes and upgrades, as well as replacement activities.

Checklist

The following checklist can help answer the question "Are all the bases covered?" once the activities above have been planned or performed. **The checklist is not sufficient by itself.**

- ☑ Has the extent of traceability been defined?
- ☑ Are all user needs traced to system requirements and vice versa?
- ☑ Have the concept of operation scenarios been traced to the system requirements and the validation plan?
- ☑ Have the user needs been traced to the system validation plan?
- ☑ Is a requirements management tool needed for the project?
- ☑ If a requirements tool is needed, has it been procured and configured for the project?
- ☑ Is the staff trained on the use of the tool?
- ☑ Is access to the requirements management tool available to all stakeholders and the development team?
- ☑ Have the system requirements been traced to the system verification plan?
- ☑ Have the system requirements been traced to the design?
- ☑ Has the design been traced to the verification plans?
- ☑ Has the design been traced to implementation artifacts [SW source code, HW documentation etc.]?

- ☑ Have the verification procedures been traced to the verification plan?
- ☑ Have the validation procedures been traced to the validation plan?
- ☑ Has all needed supporting project documentation been traced to?
- ☑ Has traceability been maintained during the operations & maintenance, changes & upgrades, and retirement & replacement?

Are there any other recommendations that can help?

TIP

For projects that have roughly 100 system requirements or more, procure and use a requirements management or a database tool to capture, trace, and manage the project requirements.

- The tool should be installed and configured in the early stages of the project
- Staff should be trained in the use of the tool
- The tool should have the capability such that all staff have access to the tool
- The tool should be able to trace within and between classes of the schema.
- The tool should support document generation, or interface with a document generation tool.
- The tool should provide a change management capability where stakeholders can recommend changes to requirements and traceability.

For small project [less than 100 requirements], a spreadsheet may be used to capture and trace requirements. A schema must be defined on what are the naming conventions and how the links will be identified. This is a low-cost approach but in the long term it may be more labor intensive. The choice of the tools should be determined on the long-term growth of the system.

3.4.4 Risk Management

Introduction - Risk Management

Although ITS projects come in many shapes and sizes, they all use technology (computers, communications, sensors, etc.) and frequently include the exchange of information between systems. The technology and integration that sets ITS projects apart also creates challenges for the ITS project manager. Every ITS project manager wants a successful result at the end of the project, with "success" measured by:

- how well the implementation satisfies the needs of the people who use it, and
- how closely the project stayed within the budgeted cost and schedule.

The effects that can compromise success are collectively referred to as "risk". A key value of systems engineering is to manage the risk on a project. The following is a brief discussion of risk.

ISO 17666:2016 defines risk as an "undesirable situation or circumstance that has both a likelihood of occurring and a potentially negative consequence on a project". Other definitions equate risk to variability or to the chance that desired outcomes will not be achieved. The New Zealand transportation agency is an international leader in risk and asset management and it defines risk as "the chance of something happening that will have an impact on objectives. it is measured in terms of a combination of the likelihood of an event and its consequence.”[2]

There are many areas of risk that should be considered when deploying an ITS project. These areas, along with some questions relevant to understanding how significant the risk might be, include:

- Technical (e.g., Is the project using any technologies that have not been widely deployed or that the project team is unfamiliar with? Are the requirements well defined? Are the development or test facilities inadequate? Is all technical documentation receiving adequate review?)
- Schedule (e.g., Is the schedule aggressive? Are there particular tasks for which small schedule slips will have a major impact on the final deliverable? Is the schedule dependent on timely delivery of equipment, information, or review from parties that may not be bound by the project schedule?)
- Cost or Funding (e.g., Is the budgeted cost realistic for the planned systems? Is funding for the project secure, or is only part of it in place? Are there pending agency budget cuts that might impact development or operations?)
- Institutional (e.g., Does the project require agreements related to agency data sharing that haven’t yet been created? Are there regulations or agency hurdles that must be overcome for the project to succeed?)
- Personnel (e.g., What will happen if there is a loss of key agency or contractor personnel? What will be the impact if key personnel do not have adequate experience?)
- Environmental (e.g., Does the deployment schedule take into account typical seasonal conditions- call for installations at a typically rainy time of year, or during winter months? Are there environmental restrictions that might impact system deployment?)

Description

Risk management seeks to help a project avoid impediments to successful completion in keeping with the project environment. Similar to the management of the project, the management of risks needs to be planned and executed. Unlike project management, the different tasks associated with risk management are expected to be performed concurrently and repeatedly. A representative risk management process is shown conceptually in Figure 27. Risk management is a developed area with existing standards and reference material. (Technical Committee ISO/TC 262, 2020)

[2] Risk-Based Transportation Asset Management: Evaluating Threats, Capitalizing on Opportunities, FHWA, June 2012.

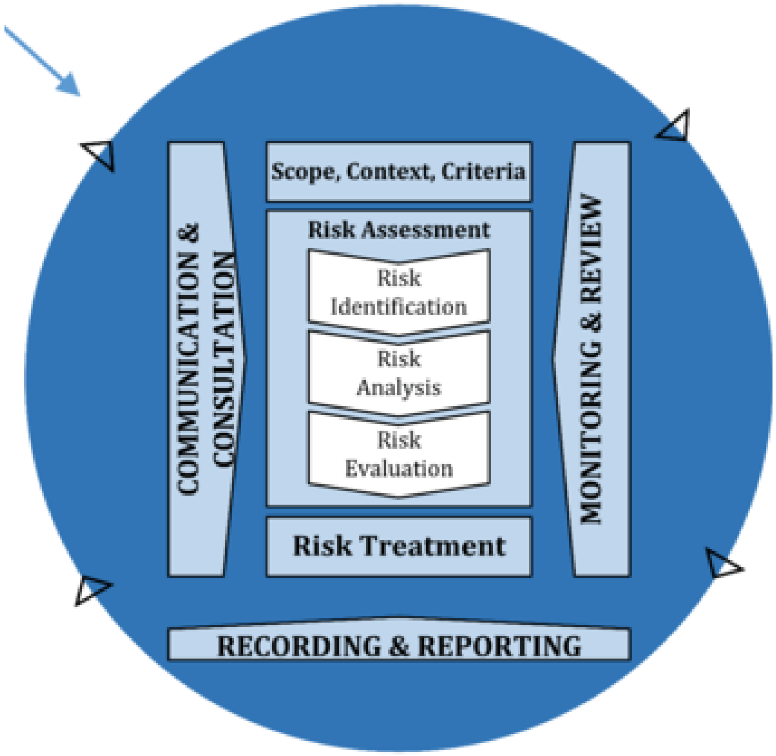


Figure 27: Representative Risk Management Process
(Source:ISO/TC262)

To be valuable to a project, risk management must identify the relevant risks, assess the threat that the risk poses to the project, and mitigate risks that are determined to be of significant risk to the project. The value of mitigating a risk must be assessed against the impact, likelihood, and urgency of the risk occurring compared against the available resources of the project to mitigate the risk.

Risk management requires the support of the project team, including both management and staff. A successful risk management effort receives the support of project management to assign risk management activities with sufficient staff resources, to make risk management an ongoing effort, and to foster an environment where project staff communicate accurately with risk management staff. The risks to the project are most directly known by the line employees who will design, implement, or operate the systems or services implemented by the project. The staff assigned to risk management are most effective in risk identification when the line employees are empowered and encouraged to directly communicate with risk staff.

Risk management inherently has an adversarial component with the project team. The risk analysis performs a critique of the ability of the project team. Any risk that is identified can be interpreted as a criticism of the resources of the team. The assessment of a risk as highly likely to occur can be interpreted as questioning the skills or decisions of team members. Risk mitigation can also be viewed as an additional assignment that comes on top the assignments of an already-overworked staff member.

The value of mitigating a risk should be assessed against the resources to be applied to the mitigation effort in determining if the mitigation effort should be undertaken. Any time that management determines that a risk must be mitigated, team resources are diverted from their ongoing assignment to the mitigation assignment, which can impact the cost or schedule of the project. Mitigation efforts can represent an opportunity cost related to the portions of the project delivered late or foregone.

In a full effort for risk management for a project with sufficient resources, dedicated staff should be assigned to perform activities related to risk identification, risk analysis, and risk mitigation, along with maintaining a record of each risk throughout the project. Such a level of effort should be implemented for projects with high-risk characteristics both in likelihood of risk occurrence and impact of occurrence. Risk management in this scenario requires ongoing commitment from project management to regularly review the evolution of the risk register and apply project resources to resolve issues determined as posing a threat to the success of the project.

The efforts in performing risk management should be tailored in response to a lower-risk project environment. For a project that initially is determined to be low risk, the risk identification may be performed by project management staff speaking with designers, implementers, and future operators. Risk management should continue throughout the project, with the possibility of risk mitigation continuing until risks posing significant threats to the project are no longer relevant. Limiting the risk management effort for a project with limited resources is a risk in itself.

Are there any other recommendations that can help?

All useful systems incur some risk. The goal is a balance between system performance and risk. That is why the focus is on only the most critical risks. Lesser risks will and should be accepted.

From a management viewpoint, there are four ways to handle risk.

1] Mitigate the risk by allocating contingency funds to its resolution if it becomes necessary

2] Accept a risk that cannot realistically be mitigated, such as an earthquake

3] Avoid the risk by changing the requirements or design

4] Transfer the risk [e.g., to an insurance company or to a developer under a fixed price contract].

Even if a dedicated risk management team is in-place, *everyone on the team must be encouraged to identify potential risks*. A “shoot the messenger” atmosphere will only allow hidden risks to grow out of control.

Uncertainty is what makes risk management both difficult and essential. There are statistical techniques such as probabilistic decision theory for reasoning under uncertainty. The most basic technique is expected value. Risk is computed as the probability of occurrence multiplied by the consequence of the outcome. Probability is between 0 [minimal] and 1 [certain]. Consequence is expressed in terms of dollars, features, or schedule. Multiplying probability of occurrence and consequence [impact analysis] together gives a risk assessment value between 0 [no risk] and 1 [definite and catastrophic].

When exact data is not available for expected costs and probabilities. One can get reasonably good results simply by rating risks qualitatively relative to three to four categories in each of impact (rows) and likelihood (columns). Below is an example of the matrix used for such an evaluation. The numbers are the order in which the risks are to be considered. The lower the number the higher the priority of the risk. This table is just for illustration purposes, and in the real world its unlikely any project would proceed with risks in the categories 1-3 in the table.

Impact	Likely 0.7-1.0	Probable 0.4 to 0.7	Improbable 0.0 to 0.4
Catastrophic 0.9 to 1.0	1	2	6
Critical 0.7 to 0.9	3	4	8
Marginal 0.4 to 0.7	5	7	10
Negligible 0 to 0.4	9	11	12

A closer look at definitions and examples of impacts and probability ratings

Here are definitions to firm up the impact levels used in the matrix [from INCOSE Systems Engineering Handbook]. Here the "mission" is the purpose of the system such as traffic management.

- **Catastrophic**: Failure would result in project failure meaning a significant degradation/non-achievement of technical performance.
- **Critical:** Failure would degrade system performance to a point where project success is questionable, for example: a reduction in technical performance.
- **Marginal**: Failure would result in degradation of secondary system functions, a minimal to small reduction in technical performance.
- **Negligible:** Failure would create inconvenience or non-operational impact. No reduction in technical performance.

Here are examples of some of the characteristics that would impact the probability of failure [adapted from INCOSE Systems Engineering Handbook].

Maturity:

- Existing system, probability is 0.1
- Minor redesign, 0.3
- Major change [feasible], 0.5
- Complex design [technology available], 0.7
- State of the art [some research done] 0.9

Complexity:

- Simple design, 0.1
- Minor increase in complexity, 0.3
- Moderate increase in complexity, 0.5
- Significant increase in complexity, 0.7
- Extremely complex, 0.9

Note that if there are multiple risks. The overall probability will be at least as high as the highest of them. Often it will be even higher.

Managing Risk with Systems Engineering

At its core, systems engineering is a set of activities that seek to reduce risk, particularly the risk that the resulting system will not support the intentions of the agency, but also including the risk of schedule and cost overruns. Systems engineering, like all good engineering processes, increases the likelihood that the implementation will fulfill the user's requirements. Systems engineering processes do impose initial costs to a project, and the level of systems engineering required by any given project should be proportional to the risk imposed by the project.

The discussion below provides examples of risk mitigation strategies for four key categories of projects, as well as providing a guide to the type and amount of systems engineering that needs to be done. A further discussion of how to tailor systems engineering activities to the risk level of a project is given in Section 4.3. The four categories of projects are:

- Projects that will develop new software or hardware technology, develop new agency operational practices, or develop new interagency interfaces.
- Projects similar to previous successful projects for which systems engineering was performed.
- Projects that fit into a category of projects for which the same systems engineering documents can be developed and applied to the whole category
- Projects that do not require development work that will select from existing technology products only, including those covered by Model Systems Engineering Documents (see discussion below).

New Development Projects

These represent the greatest risk, particularly when an agency is undertaking a new operational activity as well as developing a new technology. For these projects a comprehensive approach for performing and documenting systems engineering from scratch is important for the management of risk for projects in this group. While traditional ITS technology is now common and available as finished products, systems software is still subject to considerable integration, and newer operational approaches are emerging, such as technologies related to connected and automated vehicles. Many of these technologies are not developed to the stage of commonly available products and software, and the detailed custom systems engineering described in Chapter 3 will be applicable.

Projects That Re-Use Systems Engineering Documents

Agencies often implement technology projects that extend or duplicate work done in prior projects. They support the same activities that are performed by the same stakeholder agencies. In these cases, those documents provide a *complete* and *correct* statement of needs and, particularly, *requirements* for the new projects. In these cases, agencies should simply reuse the previous documents.

Agencies may take the documents for a prior project and make the necessary minor changes, and then use them specifically for the project at hand. The extent to which the needs and requirements contained in those documents avoid being specific about technology governs their reusability to later projects.

Projects that fit into a category of projects

Another way to spread the applicability of one set of documentation is through the use of *categorical systems engineering.* For example, if an agency plans to build a number of systems in different locations as a series of disconnected projects, but all the projects respond to the same needs and fulfill the same requirements, then one set of systems engineering documents can be applied to all the projects. In one example, the Ohio Department of Transportation defined categories of projects, such as small-scale traffic signal systems, all of which are owned and operated by one agency whose arterial operational approach using those systems is the same for all projects. The needs and requirements for the category fully apply to each project. Minnesota DOT has adopted a similar approach, where lower-risk projects in defined categories can use systems engineering documents that are already developed, while more complex and risky projects can use templates to guide the development of project-specific systems engineering documents.

Product Selection Projects

Many projects do not involve development of product technologies, either hardware or software, and the principal decision during deployment is selecting the products to be used. These sorts of projects deploy the most common basic technologies used in operations, including traffic signals, dynamic message signs, closed-circuit traffic monitoring cameras, and detection and traffic sensor systems. For these, FHWA has developed a library of Model Systems Engineering Documents. These documents provide a model concept of operations, from which an agency user will simply select the needs that are appropriate to the project at hand, with minimal required tailoring. Once the needs are selected, appropriately traced and worded requirements are associated with those needs that can be used for procurement and acceptance of the products in a project setting. As of this writing, FHWA has published the following Model Systems Engineering Documents:

- Model Systems Engineering Documents for Traffic Signal Systems (working draft), available for download here: https://ops.fhwa.dot.gov/publications/fhwahop19019/index.htm
- Model Systems Engineering Documents for Dynamic Message Signs (final), available for download here: https://ops.fhwa.dot.gov/publications/fhwahop18080/index.htm
- Model Systems Engineering Documents for Closed-Circuit Television (final), available for download here: https://ops.fhwa.dot.gov/publications/fhwahop18060/index.htm
- Model Systems Engineering Documents for Transportation Sensor and Detection Systems (in development as of this writing)

These approaches help minimize the development of unneeded documents, but still provide meaningful benefits for mitigating the risk of technology deployment projects.

Checklist

The following checklist can help answer the question “Are all the bases covered?” once the activities above have been planned or performed. **The checklist is not sufficient by itself.**

- ☑ Is the risk management plan included in the Project Plan/SEMP?
- ☑ Have all sources of risks been identified?
- ☑ Technical [e.g., new detectors do not perform as expected]
  - Institutional [e.g., agency data sharing, new regulations, public opposition]
  - Funding [delays or cuts]
  - Environmental [e.g., temperature levels for outdoor field equipment, restrictions on building]
  - Personnel [e.g., loss of key personnel, substandard performance]
  - Commercial [e.g., vendor does not deliver the commercial product]
- ☑ Were experts and stakeholders queried in all the areas of risk to develop a broad list of credible risks?
- ☑ Are the risks prioritized and the most critical ones identified?
- ☑ For each high priority risk, are there ways to eliminate the risk? Or, reduce its likelihood and/or impact?
- ☑ For each high priority risk, have the symptoms of the problem and a means for monitoring them been identified?
- ☑ Are the high priority risks regularly monitored throughout the project?
- ☑ For each high priority risk, is there a risk resolution plan?

3.4.5 Trade Studies

Introduction

Trade studies compare the relative merits of alternative approaches to ensure that the most cost-effective system is developed. They maintain traceability of design decisions back to fundamental regional operations strategies, needs, and requirements. Trade studies do this by comparing alternatives at various levels for the system being developed. They may be applied to concept, design, implementation, verification, support, and other areas. They provide a documented, analytical rationale for choices made in system development.

Description

Trade studies can be used in various phases and at different depths throughout system development to select from alternatives or to understand the impact of a decision. The inputs vary depending on what is being analyzed. For example, early in system development, the alternatives may be broad concepts. Later, in the design phase, they will be design alternatives. During hardware and software implementation, the implementation team may look at alternative hardware components or language and development environment alternatives for software. In each case, the alternatives are compared against a set of criteria and the decisions are documented. Stakeholder input may be solicited to define and rate the criteria and to validate the results. The analysis may be done qualitatively, or by a model or simulation. Trade study activities typically include the following activities.

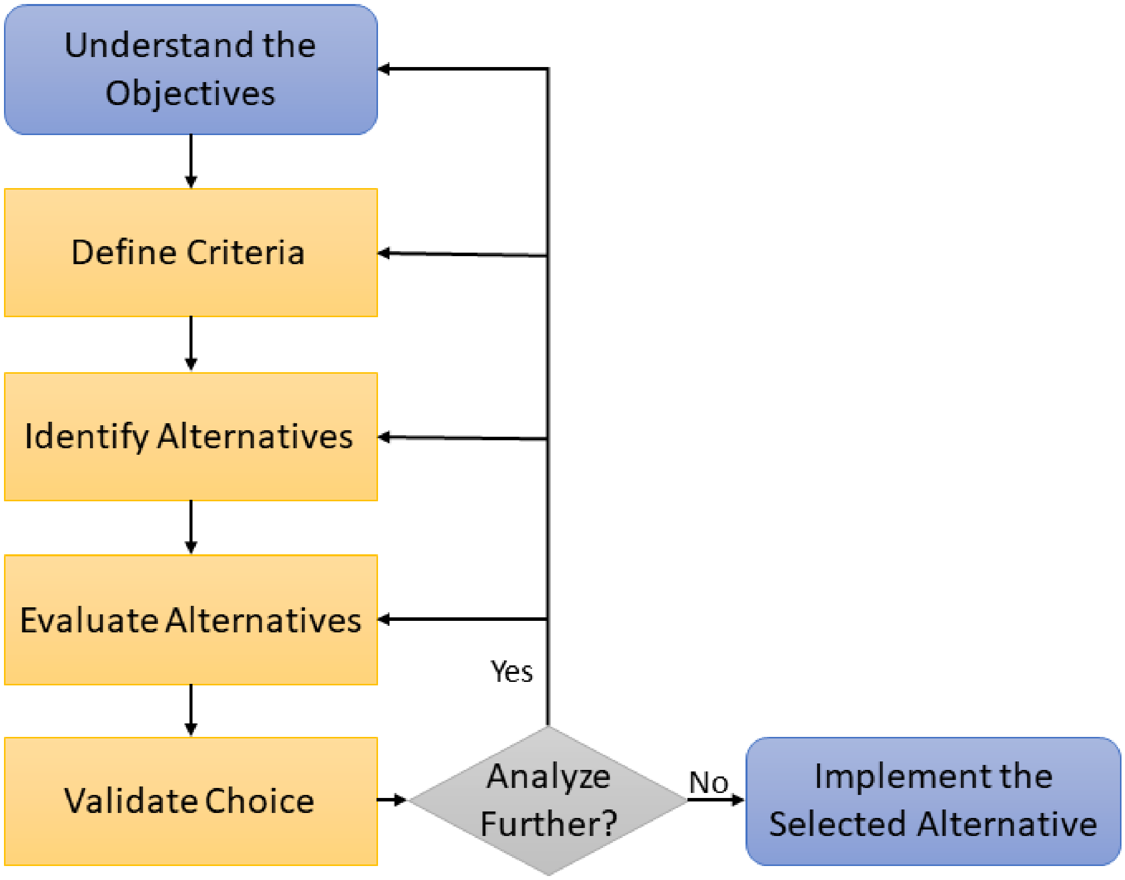


Figure 28: Trade Study Activities
(Source: FHWA)

Understand the objectives
First, define the question the trade study is to answer. This may be the selection of the most cost-effective concept or design. It may be to narrow down choices to support a more detailed evaluation. It may be to demonstrate that the choice made is the best one.

Define criteria and weightings
Experienced staff will draw from the available inputs to identify the key evaluation criteria for the decision under consideration. These are typically measures of effectiveness, metrics that compare how well alternatives meet the needs and requirements. Examples are capacity [vehicles per hour], response time, throughput, and expandability. Ideally, the criteria are developed early in the trade study process, before the alternatives are selected, to avoid subconsciously biasing the criteria to favor a pre-determined preferred alternative.

Generally, there are multiple criteria and they aren't always correlated. For example, an alternative that minimizes schedule impact might increase risk or cost. It is important to rank or weight the criteria so that the selected alternative scores best against the most important criteria. Stakeholder input may be solicited to assist with the identification and weighting of the criteria based on relative importance.

Identify alternatives

A trade study starts with alternative concepts or designs that are to be evaluated. Be sure that all reasonable alternatives are on the table. It is often good practice to include a "Do Nothing" alternative as the base case, depending on the nature of the trade study.

Evaluate alternatives

The identified alternatives are evaluated against the defined criteria. Generally, the emphasis is on performance criteria such as speed or effectiveness. For each alternative, the criteria may be evaluated quantitatively or qualitatively supported by methods including simulation, performance data gathered from similar systems, surveys, and engineering judgment. These disparate evaluations are merged using the weighting factors to give a measure of overall effectiveness for each alternative.

In most trade study analyses, the cost and schedule impacts associated with each alternative are estimated. Estimate the cost of each alternative including the development cost and the life cycle cost, which includes operation and maintenance. Use the techniques of risk assessment to compare the alternatives relative to technical or project risk. Determine the impact of each alternative on the schedule. Eliminate those that introduce too much cost or schedule risk.

For complex trade studies with many alternatives, the overall evaluation may involve multiple passes where a limited set of criteria is applied to a broad set of alternatives in order to focus additional analysis on the most promising alternatives. A more complete analysis can then be performed on the subset of alternatives that scored well in the first pass analysis. Sensitivity analysis may also be used, especially with simulation, to see the effect of changes in sub-system parameters on overall system performance. The sensitivity analysis and the evaluations may suggest other, better alternatives.

Select and document the preferred candidate, typically using a trade study matrix that lists the alternatives on one axis, the criteria on the second axis, and the relative scores of each alternative versus each criteria in the body of the matrix. Plotting each alternative's overall effectiveness, based on the combined weighted metrics, cost, or the other factors is also useful for evaluating the relative merits of each.

Validate the choice

The trade study documentation should support review of the decision by stakeholders, so the trade matrix should be easily comprehensible to management and other stakeholders who may not be as familiar with the technical details. Document the decision, the parties involved, and the rationale behind it to provide traceability back to the higher-level objectives. The documented trade study is also a repository of alternatives in case a change is needed down the road. It is also possible that stakeholder input may require iteration and refinement of the trade analysis to reflect evolution in criteria, additional identified alternatives, or desired evaluation changes.

Checklist

The following checklist can help answer the question "Are all the bases covered?" once the activities above have been planned or performed. **The checklist is not sufficient by itself.**

- ☑ Has a broad and reasonable selection of alternatives been examined?
- ☑ Was a fair and balanced set of criteria used to evaluate the alternatives?
- ☑ Is the selection rationale documented?

- ☑ Does the rationale for the trade study conclusions flow out of the trade study inputs –strategies, needs, and/or requirements?
- ☑ Is the sensitivity of system effectiveness to changes in key parameters well understood?
- ☑ Have key stakeholders reviewed and support the trade study results?

4 Managing SE Projects

This chapter provides an agency perspective on managing the systems engineering for an ITS project. It discusses evaluating the risk associated with a project and considers the actions an agency might engage in depending on the level of risk identified for the project. The chapter concludes with a discussion of the FHWA regulation/ FTA policy relating to Architecture and Standards, including a discussion of the requirements for Systems Engineering Analysis.

4.1 Intro to the Agency Perspective

This section provides an overview of SE management from the perspective of transportation agencies. Successful SE management, and thus successful project deployment, depends on a clear understanding of the roles of ITS participants, many of which are related to SE processes:

- An agency is the owner if it is financially responsible for and has control over the system. Most traditional, non-connected ITS components are typically owned by a transportation management agency.
- An agency is the operator if it is responsible for interacting with the ITS and controlling its state. This might be refined to manager and operator roles, where the manager is accountable for one or more human interface roles, while the operator performs the actual ITS device interactions during system operations.
- An agency is the maintainer if it is responsible for keeping the hardware and software that comprise system in an operational state. The maintainer typically is delegated authority by the owner. The maintainer interacts with the target Resource so as to keep that Resource in the Operational state
- The installer is the organization that performs the initial delivery, integration and configuration of the target Resource. This might be a system integrator, an agency performing its own installation, or a device supplier that performs on-site installation.
- The developer is the organization that creates the system and its documentation. In most cases this organization will be a system integrator or device supplier.
- The certifier is the organization that verifies that the system meets relevant performance, functional, environmental and quality requirements. This could be an independent third party or it could be the same entity that has the developer role. In the case of connected vehicle projects this role includes the verification of the security management system.

The agency that will be the owner of the operational system will acquire a set of development services to develop the ITS project. The services can be either in-house or contracted. The system's owner and operating organization will ultimately be responsible for the system and its operations & maintenance. The system's owner needs to supply clear requirements and expected project outcomes to the development team. These outputs must be compatible with the long-term operations & maintenance goals of the system's owner & stakeholders.

Each project has an inherent level of risk that will help determine how a systems engineering process is applied to the development. In order to decide what that level of risk is, the agency needs to do a risk evaluation, which is discussed in Section 4.2 below. Once the level of risk has been determined, then the agency needs to consider their approach to systems engineering given the defined level of risk. This is discussed in Section 4.3 below.

Section 4.4 provides consideration for how an agency should manage the SE effort on a project. Agencies possess differing levels of systems engineering expertise and experience, which will affect their ability to manage ITS projects. The section will provide insights into the role of the agency in different types of projects. The section will set their expectations about the project and what they should expect to see.

The final Section 4.5 describes how agencies can interpret the FHWA regulation/ FTA policy on systems engineering.

4.2 Risk Evaluation

A successful ITS project is one which completes on schedule, within budget, and delivers all capabilities required in a manner that meets the needs identified by its owner. ITS projects that are prone to failure typically involve systems not previously deployed by the agency, a new process that the lead agency has not done before, and/or the need to coordinate with other agencies. The project also might include new technology or new software, or new communications, or joint efforts with new partners. A key first step in the management of an ITS project is to evaluate the level of risk which the project will entail.

Project risks relate to the general categories of risk defined in Section 3.4.4 and include technical, schedule, cost, institutional, etc. When considering the overall risk associated with a project the likelihood that these risks listed above can increase or decrease significantly based on several identified factors associated with ITS projects. The factors are:

1.) Number of jurisdictions and modes

2.) Extent of software creation

3.) Extent of proven hardware and communications technology used

4.) Number and complexity of new interfaces to other systems

5.) Level of detail in requirements and documentation

6.) Level of detail in operating procedures and documentation

7.) Service life of technology applied to equipment and software

These risk factors are discussed in more detail in Table 14.

Table 14: Identifying Risk Factors

Factor	Type of Risk	Low-Risk Project Attributes	Medium-Risk Project Attributes	High-Risk Project Attributes	Risk Factors
1 Multiagency involvement	Institutional	Single agency and single transportation mode (highway, transit or rail)	Multiple sections, departments or disciplines within an agency or a small number (e.g. 2or 3) agencies.	Multi-Jurisdictional or Multi-modal	With multiple agencies, departments, and disciplines, disagreements can arise about roles, responsibilities, cost sharing, data sharing, schedules, changing priorities, etc. Detailed written agreements can be an important way to reduce the risk for higher risk projects. Even within a single agency, agreements may be relevant if the project spans departments or disciplines.
2 Software creation	Technical, Schedule	No software creation; includes software already used by agency to address similar requirements	Primarily software / hardware used by the agency or existing software /hardware based with some new software development or new functionality added to existing software - evolutionary development.	Custom software development is required	Even existing ITS software-based systems have many optional capabilities and require customization and configuration. Custom software requires additional development, testing, training, documentation, maintenance, and product update procedures --all unique to one installation. This is very expensive, so hidden short-cuts are often taken to keep costs low. Additionally, integration with existing software can be challenging, especially because documentation is often not complete and out-of-date.

Factor	Type of Risk	Low-Risk Project Attributes	Medium-Risk Project Attributes	High-Risk Project Attributes	Risk Factors
3 Technological maturity	Technical, Schedule	Hardware and communications technology already used by the agency with documentation to address similar requirements	Hardware and communications technology being used to address requirements that go beyond those that agency has previously defined. May involve multiple technologies to be implemented.	Hardware or communications technology are “cutting edge” or not in common use.	New technologies are not “proven” until they have been installed and operated in a substantial number of different environments and have It has been demonstrated (and documented) to fulfill the same set of requirements as are being used for the project in question. New environments often uncover unforeseen problems. New technologies or new businesses can sometimes fail completely. Multiple proven technologies combined in the same project would be high risk if there are new interfaces between them.
4 Interface maturity	Technical	No new interfaces	System implementation includes one or two major subsystems. May involve significant expansion of existing system. System interfaces are well known and based primarily on duplicating existing interfaces.	New interfaces to other systems are required.	New interfaces require that documentation for the “other” system be **complete and up-to-date.** If not (and often they are not), building a new interface can become difficult or impossible. Duplication of existing interfaces reduces the risk. “Open Standard” interfaces are generally well-documented and low risk.

Factor	Type of Risk	Low-Risk Project Attributes	Medium-Risk Project Attributes	High-Risk Project Attributes	Risk Factors
5 Requirements completeness	Technical, Schedule	System requirements fully-detailed in writing	System requirements are only partly detailed- some additional definition work required	System Requirements not detailed or not fully documented	System Requirements are critical for an RFP. They must describe in detail all of the functions the system must perform, performance expected, plus the operating environment. Good requirements can be a dozen or more pages for a small system, and hundreds of pages for a large system. When existing systems are upgraded with new capabilities, requirements must be revised and rewritten.
6 Operational preparedness	Personnel, Schedule	Operating procedures fully-detailed in writing	Some operating procedures are detailed in writing, but revisions or new portions are required.	Operating procedures not detailed or not fully documented	Standard Operating Procedures are required for training, operations, and maintenance. For existing systems, they are often out-of-date.
7 Technological obsolescence	Technical, Schedule, Cost	None of the technologies used are near end of service life	One of the technologies included are near end of service life	Multiple technologies included are near end of service life	Computer technology changes rapidly (e.g. PC's and cell phones become obsolete in 2-4 years). Local area networks using internet standards have had a long life, but in contrast some mobile phones that use proprietary communications became obsolete quickly. Similarly, the useful life of ITS technology (hardware, software, and communications) can be short. Whether your project is a new system or expanding an existing one, look carefully at all the technology elements to assess remaining cost-effective service life.

There are other factors that can affect project risk including the experience of agency staff implementing the project, and the clarity of the user needs defined for the project.

An example of a **Low-Risk** ITS project is the addition of 30 full matrix dynamic message signs to an existing system that has five identical signs already deployed. Project sponsor's needs and project requirements are well defined. No changes are needed to the existing central or field equipment. The system was initially designed to accommodate these additional signs so no additional software is needed. Assumptions are: 1) the initial system has been completed and the system is working well, 2) the contractor will deploy the signs, poles and foundations, controllers, and wire the controllers into the signs, and 3) the agency will add configuration information about the signs at the central computer. Updates to the existing plans have been reviewed to ensure that the original design and implementation is not adversely affected as a result of adding the elements.

During the design process, it may be discovered that a number of changes to the existing system are needed in addition to adding the expansion elements. This need could arise because of new and better technologies (or the old hardware is no longer available), or because of the desire to improve or expand the functionality of the "previous" system, or because of the need to use the system in a different way, e.g., sharing control with another party. **Any of these instances would elevate the project to a Medium-Risk implementation.**

Additional examples of **Low-Risk** ITS projects include:

- Adding new or existing signals to a new or coordinated signal system.
- Adding five identical CCTV cameras to the existing 20 – with no other changes to the system or how it's used.
- Adding 50 identical new loops to the existing 200 – no other changes.
- Installing an existing parking management system at 2 additional garages – with no changes
- Expanding the pre-existing system/network by adding several more XXXX units – with no changes. (XXXX can be almost any ITS element).
- Expanding existing communications systems – this consists of extending existing fiber-optic or wireless communications systems, using the same technology and specifications as the pre-existing system.
- Leasing turnkey services only (e.g., website-based information service) – with no hardware or software purchases.

High-Risk projects imply that some (or all) of the seven risk factors have attributes in the medium or high-risk range. This can often occur when the project is a new system deployment such as creation of Transportation Management Centers, introduction of regional Integrated Corridor Management, development of new parking management systems, and creation or expansion of toll lanes. To give a more explicit example, a High-Risk ITS project might result from adding the following new requirement to the previously described Low-Risk project: "The changeable message signs will have shared control with a partner Agency B." For this example, Agency B manages events at two activity centers. As part of the installation, Agency A will be installing six signs that would assist agency B for their event management. Agency B would use the CMS to divert traffic to get the attendees in and out of the event faster and more safely. To enable this shared control, new software may need to be developed and integrated into the existing system. With this requirement for new functionality (shared control), new risks and complexity are introduced. Although the traditional roadway Design/development and construction process is needed for the signs and controllers at each location, there will be a need for systems engineering to address the software development and integration efforts. In this example, revisions to the existing “concept of operations” and development of agreements for interagency coordination will be especially important to clarify expectations and avoid future disputes.

Additional examples of High-Risk ITS projects include:

- Development and/or deployment of applications for mobile computing devices that involve safety or liability considerations or integration with larger systems.
- Local agency using consultants to operate a TMC and/or centralized signal management facility.
- Multi-jurisdictional or multi-modal system implementation --Because of the external interfaces required, these projects generally include substantial software development. For example:
    - A traveler information system that collects data from multiple agencies or modes
    - A Bus Traffic Signal Priority system between City Traffic and Regional Transit, or one that crosses multiple jurisdictions.
- The first stage of an “umbrella” system implementation. During this first stage, the full systems engineering process would be used to develop the overall system framework plus the first implementation of that framework. For example:
    - New Traffic Signal Coordination system design plus implementation at an initial number of signals, with more signals added in later project(s).
    - New Traffic Information System design plus the first implementation in Cities X and Y, with more cities added in later project(s).
    - New Electronic Fare-Payment System design and initial implementation on Metro buses, with other transit agencies added in later project(s). If subsequent stages replicate the initial implementation, they would not be high risk. Instead, they fit the definition of a low risk ITS project, expanding the existing system with no new capabilities, and no new interfaces.

To evaluate the agency’s overall level of risk of an ITS project, consider the 7 areas of risk described above. The result to this evaluation is not a binary (or trinary) choice, but should be evaluated based upon which areas of risk are higher and which lower. On one end of the spectrum, if all seven areas are judged to be have low risk attributes, then it’s clear that the overall risk would be judged as low. On the other end of the spectrum, a project that has all (or most) of the attributes that are high risk, and the project is clearly a high-risk project. If the seven areas are a mix of risk attributes levels, then an agency should consider customizing their process (see section below) around the level of risk to each of the questions. For example, if there are no requirements documented for the system, but other factors veer toward the low risk side of the spectrum, then the project should pay particular attention to requirements.

4.3 Process Tailoring

The systems engineering process is laid out in Chapter 3. This section discusses how to apply that process in three general cases- for low, medium, and high-risk projects, providing agency guidance for the tailoring of the different steps of the systems engineering process for the project

When tailoring the systems engineering process for different projects, a given step may be performed very informally (e.g., on the back of an envelope, or in an engineer's notebook); on other projects, the activity may be performed very formally, with interim products under formal configuration control. This document is not intended to advocate any level of formality for the different levels of risk, but provide approaches that can be applied to relevant situations as applicable.

The systems engineering analysis requirements identified in FHWA 23 CFR 940.11/FTA Policy Section VI allow the systems engineering analysis effort to be tailored so that it is on a scale commensurate with project scope.

INCOSE also stresses variation in the systems engineering process:

Like all processes, the Systems Engineering process at any company should be documented, measurable, stable, of low variability, used the same way by all, adaptive, and tailorable! This may seem like a contradiction. And perhaps it is. But one size does not fit all.

This message is particularly important for ITS projects because so many of our projects are smaller, less complex, less risky projects like signal system upgrades. Even for small projects, you still should have documented requirements, design, and verification procedures. Tailoring allows you to adjust the amount of formal documentation and reviews and to focus the process on those steps that are most critical to your project's success. Ultimately, you want to define a process that will address the project's risks, no more and no less, so a preliminary risk analysis is a good way to determine how much process is appropriate.

Tailoring is not an invitation to skip steps.

The table below summarizes approaches for tailoring the process based on the three types of projects. These approaches should take into account your own environment, process requirements, and staff experience when you tailor the process for your own project. The best approach is to think about the risks of your particular project and determine how to best mitigate those risks with a tailored systems engineering process. Think about the process ahead of time and write down what you are going to do so that everyone on your team and your stakeholders understand and agree on the right steps to follow and the level of detail that is needed. Whether you call it a project plan, a systems engineering management plan (SEMP), or something else, it's critical to put your process and your plan into writing. A further discussion of how a SEMP supports the management of a project in contained in Section 3.3.4.

Table 15: Tailoring the Systems Engineering Process

Process Step	Low Risk Project	Medium Risk Project	High Risk Project
Regional ITS Operations Planning (Regional ITS Architecture)	Effort: None-low Existing architecture mapping applies	Effort: Low Add new services or interfaces to existing mapping	Effort: Low New mapping
Feasibility Study, Concept Exploration	Effort: None May be an existing study, but if not, none needed due to well understood system upgrade or expansion (e.g. a common low risk project)	Effort: None No new feasibility study likely to be needed.	Effort: Medium Depending on nature of risks, a concept exploration might be desirable that would survey existing system and alternatives
Stakeholder Needs (Concept of Operations)	Effort: None Existing ConOps should apply	Effort: Low Depending on nature of risks, some new needs likely and should be defined. If an existing ConOps, it likely needs update.	Effort: Medium If ConOps does not exist, one should be developed, or if one exists, it may need a more extensive update
System Requirements	Effort: None Existing SRS applies – review for any changes needed	Effort: Medium Define new requirements either modifying existing SRS or creating a new one.	Effort: Medium Develop requirements document and verification plan
Design Definition	Effort: None OTS products. Existing specs apply	Effort: Low Limited design definition likely needed. Define project architecture illustrating any new interfaces.	Effort: Medium Design definition needs to be developed, or if existing, updated.
Implementation	Effort: Low Purchase existing and previously documented HW/SW and Identify any configuration needed	Effort: Low-Medium Purchase existing HW/SW that will be used in a way beyond current agency implementations.. Develop any custom SW	Effort: Medium-High Develop Custom SW. Level of effort depends upon complexity of project and level of new development.

Process Step	Low Risk Project	Medium Risk Project	High Risk Project
Integration/Testing	Effort: Low Verify factory tests performed. Verify devices and comm working. Reuse original procedures	Effort: Medium Unit test custom SW. Checkout purchased HW and SW. Host/integrate custom SW on HW. New procedures for new capabilities. All documentation ready.	Effort: Medium-High Tasks similar to Medium Notes, difference is in the level of effort based on complexity of project.
Initial Deployment	Effort: Low HW/SW previously used by the agency to address similar requirement, which has normal construction issues	Effort: Low-medium Need system installed and configured, staff trained in operation	Effort: Medium Need system installed and configured, staff trained in operation
System Validation	Effort: None System validation already performed	Effort: Low-medium Validate new needs are met	Effort: Medium Validate needs, performance, user and maintainer satisfaction

Many of the projects in the Low to Medium risk category relate to deployment of ITS field devices that are readily available from many manufacturers. In these projects, the risk can often be addressed by proper bounding of the procurement process- ensuring that all needs and requirements are properly described and documented prior to procurement. These types of risks are particularly relevant when the agency is smaller or less experienced with ITS deployments. To help address these conditions, FHWA has developed a set of Model Systems Engineering documents which support procurement of the following field devices:

- CCTV
- Dynamic Message Signs
- Central Traffic Signal Systems (recently completed, but not yet available on line)
- Transportation Sensor & Detection System (TSDS) (currently under development)

These documents provide a set of sample statements that can be used to create SE documentation that supports deployment of these ITS systems. Specifically, the documents support development of the following SE documentation:

- Concept of Operations
- System Requirements
- Verification Plan
- Validation Plan

In addition, these Model SE documents provide suggested approaches for using the documentation to support the procurement of these systems, emphasizing the importance of defining key requirements as part of the procurement process.

4.4 ITS Procurement

In keeping with applicable Federal, state, and local acquisition regulations, transportation agencies have a variety of options for issuing soliciting support for Systems Engineering activities. For an individual project, several different contracts can be issued using several contracting approaches. Coordination with the contract administration staff of the procuring transportation agency is vital in making good choices in the contracting process. Errors in the procurement process can lead to bid protests and lawsuits, which can extensively delay progress on project implementation. The topics discussed below are frequently used in ITS procurements, but represent only a subset of the available approaches to gaining support in project implementation. In all cases, the contract language needs to closely coordinated with legal and administrative departments to assure that proper selections can be made.

4.4.1 Request for Information

A Request for Information (RFI) is a pre-contract process that allows the contractor community relevant to a specific project interact with the procuring transportation agency and stakeholder community. The transportation agency requests input from interested parties on plans for a specific project. The procuring agency has wide latitude in the detail included in the project plans that are part of an RFI. The agency may be interested in gaining industry input into the products and services available in the marketplace, the current market conditions, and the vendors available to supply the needs of the project. In many cases, any type of organization can respond, with responses frequently coming from contractors that would implement the project under a future contract, consultants that are available to advise the transportation agency, vendors with applicable products, and industry societies with topic expertise. Responses also frequently come from advocacy groups that feels that their constituency can be impacted by the project such as environmental groups, neighborhood associations, and disability rights groups. Private citizens may be allowed to submit responses.

An RFI is one of the exchanges with industry prior to proposals that is identified in Federal Acquisition Regulations (48 CFR § 15.201). Responders are frequently invited to one-on-one meetings to further discuss information provided. An RFI cannot lead directly to a contract award. While RFIs are frequently issued with the intent of preparing for a future contract, the issuing of an RFI does not guarantee any future contracting opportunity.

Ideally, an RFI is issued with sufficient time prior to the scheduled project development to allow information from responders to provide input on the project. The procuring agency needs to consult with their contract administrators to assure that the RFI is in keeping with applicable regulations. Topics to assure compliance with include extent of publication of the RFI, acceptable RFI responders, necessary handling and publication of RFI responses, extent of consideration of RFI responses, and requirements for RFI response for notification of future contract correspondence or qualification for RFP responses.

A Request for Qualifications (RFQ) is a formal process to determine the ability of responding bidders to perform a desired piece of work. The RFQ details the required capabilities, resources, experience, and credentials required to perform an upcoming project along with plans for the project. In some cases, the RFQ can include specific requirements for an individual project to be designed. In other cases, the RFQ can be targeted at needed capabilities for the winning firm to be able to create a design or to accomplish tasks earlier in the Systems Engineering process, such as developing a Concept of Operations or producing requirements.

RFQs can lead either directly to a contract or lead to additional steps to acquire contracted support. One type of selection process resulting from an RFQ is Qualifications Based Selection (QBS). QBS selects the most qualified bidder to perform a designated piece of work to enter into negotiation for a contract. Cost is not a factor in selecting the most qualified bidder, but is a major topic in contract negotiations. If an acceptable contract cannot be negotiated with the firm determined most qualified, the contracting agency may enter negotiations with the firm determined to be next most qualified. The use of QBS was institutionalized with the introduction of the Brooks Act (Public Law 92-582) related to the selection of Architects and Engineers, which was enacted into US law in 1972. Many states have laws similar to the Brooks Act for state contract work.

4.4.2 Request for Qualifications

RFQs are also used as a basis for selection of contract support where details of upcoming work are not fully determined consistent with the Federal Acquisition Regulations (48 CFR § 16.504). Many transportation agencies offer the opportunity for engineering firms to establish contracts where the scope of work to be performed is undetermined, but the skills needed for typical upcoming work can be adequately described. Indefinite quantities contracts, which are also called indefinite delivery/indefinite quantities (IDIQ), provide a framework for negotiating scope for future work items in a task order. Such contracts are frequently used for initial project work using an existing IDIQ contract either assigned to a single firm or competed among a small number of firms with indefinite quantities contracts. The specifics of the administration of the task-order contracts related to limits in scope, value, allowed assignments, and competition vary based on the regulations of the transportation agency.

RFQs can be used as a prequalification for responding to a future contract solicitation. RFQs are typically used as a way to limit the number of proposers for major infrastructure projects, which frequently include significant content in addition to ITS tasks. The development of a reduced bidders list both limits the resources required by the owner to evaluate the full proposals and attempts to improve the investment potential bidders are will to make to produce high-quality proposals. Once a reduced bidders list is determined based on qualifications, subsequent phases request either detailed build proposals or significant efforts to develop a proposed design. In some cases, bidders are supported in the development of proposals with a monetary stipend.

4.4.3 Request for Proposal

A Request for Proposal (RFP) is a formal process to determine the ability of responding bidders to perform a desired piece of work. The RFP details the required capabilities needed in a system or service to be procured. While the RFP can include skills, experience, and qualifications of staff members, the majority of the RFP presents the requirements of the procurement. In build contracts, the RFP also includes the design to be implemented.

While proposals are usually evaluated based on the lowest responsive bid, other evaluation factors including qualifications and proposed design can be included in the selection criteria to determine a best value. The best value approach combines elements of qualification scoring with cost factors. Federal Acquisition Regulations (48 CFR § 15.101-1) acknowledge the advantages of considering award to other than the lowest priced offeror. To establish the best value, evaluation factors that will affect the selection and their relative importance need to be presented in the solicitation. Significant latitude is routinely allowed in solicitations in determining the important factors, based on knowledge of the involved agencies and their support contractors, provided that the criteria is clearly stated and a rationale for the factors is documented. Best value contracting has a history of protests against award lodged by a lower-cost offeror, with some protests having proven successful.

4.4.4 Invitation to Bid

An Invitation to Bid (ITB) is a formal process to determine the ability of responding bidders to provide a desired product or service. The ITB details the required capabilities needed in a system or service to be procured. While the ITB can include skills, experience, and qualifications of staff members, the majority of the ITB presents the details of the product or service to be provided. As compared to an RFP, an ITB is designed to be more specific about the requested product or service. In transportation applications, an ITB is frequently used to purchase specific devices for use in existing or developmental systems.

4.4.5 Public-Private Partnership

Public-Private partnerships (PPP or P3) are used to establish a unique relationship between public sector organizations and private sector entities. The purpose of such a relationship is to accomplish public-sector goals by leveraging efficiencies available in the private sector. In transportation applications, PPP use public resources such as roadway right-of-way to facility construction of toll lanes installation of communication infrastructure. The public sector achieves a public good such as improved transportation options or communication to ITS devices at little or no cost. The private sector offers a new or improved service for a fee, with the hope of turning a profit. Any type of contracting approach can be used with a PPP, depending on the laws and regulations of the jurisdictions involved. Creation of PPPs require significant coordination with acquisition and legal departments.

4.4.6 Implementation Models

This section will discuss the use of the design-bid-build process and the design-build process for the design and implementation portion of ITS projects and projects with significant ITS content. An introduction to the Construction Manager At Risk (CMAR) will be included.

Implementation Models
Design-Bid-Build
Design-Build
Construction Manager

Construction of transportation facilities have traditionally been focused on the physical construction of roadways. departments of transportation have developed the necessary resources to construct such facilities with skill and efficiency. The projects are selected from those developed from on ongoing transportation planning process, such as one resulting in a Transportation Improvement Plan, which considers the existing roadway infrastructure and deficiencies in the condition or capacities of the infrastructure.

The advent of implementation of electronic systems has necessitated additional approaches to implement transportation infrastructure. The roadway inventory has expanded to include communications, networking, and processing assets. The skill set required for such implementations has evolved, suggesting that additional implementation models can be advantageous.

Design-Bid-Build

Traditionally, transportation departments accomplish construction projects using a design-bid-build process. In this process, an engineering design organization develops a set of plans for a project along with related documentation such as estimates, specifications, and special provisions. After the department that will own the constructed facility reviews and accepts the design, the department asks for bids from qualified firms for construction of the facility. A construction contract is executed with the firm bidding the lowest price, using the services of in-house or contract engineers using inspections, tests, and/or evaluations to assure that the facility is adequately constructed.

This model has proven highly successful for road-building contracts. The skills needed for design and construction are typically at different firms. Many employees of the transportation department have experience in similar construction and can manage the projects.

Use of the Design-Bid-Build model allows an owner to issue one or several contracts to leverage the skill set available in both in-house staff and support contractors. The final contract prior to the bid is issued to an engineering or architecture firm for the creation of required documents including biddable design documents. Prior to this contract, the owner may acquire services from other members of their support contractors to develop project architectures, develop or refine project concepts, interact with stakeholders including the public, develop Systems Engineering approaches, and create a stable environment for the creation of the biddable design documents.

Design-Build

Design-Build is a method of project delivery where the system owner issues a single contract for the execution of design and construction services. The approach hopes to both streamline the administration of the contract by reducing the number of contracts and to streamline the technical aspects of the project by improving the flow of information and the incentives for cooperation between the designers and the builder.

Design-Build development processes are typical of technology projects where the designers and the implementers share significant skills and knowledge. In this process, the owner issues a single contract with a team of contractors who collectively offer all of the skills necessary to progress a project through the implementation phase. While a contractor may bid individually for a Design-Build contract, it is typical that all of the specialties required for a contract would be most efficiently acquired from several firms.

Most frequently, a Design-Build process is initiated once the conceptual phases for a project have reached a high level of stability. In this situation, the Design-Builder can offer a reliable approach to progressing the work along with an aggressive pricing basis. In some circumstances, a Design-Build process can be initiated earlier in the process of stabilizing the project concepts. The earlier initiation has some advantage in progressing the work more quickly, but introduces risks in the ability to accurately estimate costs of the project and introducing broader skill sets into the contract.

Construction Manager at Risk

The Construction Manager at Risk (CMAR) is a delivery method that entails a commitment by a system manager or general contractor to deliver the project within a Guaranteed Maximum Price (GMP)), which is based on the design or requirements documents and specifications at the time of the GMP plus any reasonably inferred items or tasks. The CMAR provides professional services and acts as a consultant to the owner in the design development and construction implementation phases. Often times, the CMAR also provides some of the actual construction of the project depending on the availability of bidders and the expertise the company has. CMAR is similar to the Construction Manager/General Contractor (CMGC) process.

4.4.7 Staffing Options

Most agencies involved in ITS are not able to retain sufficient in-house engineering talent to perform Systems Engineering for significantly-sized projects. Moreover, Systems Engineering is a talent that is not routinely present in a Department of Transportation or a Public Works department, giving rise to difficulty in hiring and career development for the Systems Engineering specialty.

Even when agencies responsible for ITS project development acquire technical assistance for performance of Systems Engineering activities, some activities will be of sufficient size and complexity that the agency staff will not have the specialized knowledge or available staff effort to effectively manage the project activities. In these situations, the agency should consider contracting for the execution and management of Systems Engineering activities. Depending on the scope of the project and agency resources, a combination of self-performance, contracted performance, and contracted monitoring support can be considered, with each phase in the Systems Engineering process considered independently.

The tasks to be performed within a Systems Engineering process have been presented in Section 3.3. Systems Engineering requires that certain activities are completed. While the lead agency for a project, who is referred to frequently as the “owner”, should assure that the activity is satisfactorily completed, the lead agency has options in terms of which activities that it would like to perform and which activities it would like to acquire from outside. Table 16 summarizes the responsibilities that rest with the lead organization along with alternatives for performing the activities. Table 13 shows potential considerations in assessing the success of the activities in each phase.

Self-Performance

The simplest manner in which to perform Systems Engineering tasks is to assign staff from the lead agency responsible for a project to perform the tasks and submit the Systems Engineering products for acceptance by the agency managers. The most attractive characteristics of this approach are the direct influence that an owner can exert over the process and the ability to execute a project without coordination with departments responsible for the contracting process. However, this approach has several technical and organizational challenges that must be overcome for self- performance to be completed at low risk. The challenges discussed below include in-house staffing, disciplined management, and objective evaluation.

Staffing is the most common problem encountered by agencies that would like to self-perform an ITS project. The tasks associated with efficiently using a specialized Systems Engineer or Systems Engineering department presents significant management obstacles. Providing a workload that matches available resources places constraints on the ability of a transportation agency to respond to the unavoidable variability in project volume. Staff turnover in a small department can also present issues in completion of projects and finding replacement staff with a similar skill set.

When in-house staff are used without participation of technical, administrative, or user representatives from outside a transportation organization, project managers can succumb to pressure to complete a project by shortcutting prudent processes. Significant discipline is required on the part of the project management and the staff serving as Systems Engineer to persist in producing a product to the quality mandated in the project or Systems Engineering plan. In cases where Systems Engineering processes are curtailed to meet cost, schedule, or political constraints, placing an item into a risk register as part of risk management can keep the incompleteness of the SE products a visible topic.

Evaluation done by agency staff have both the appearance of bias and the incentive to report success. An evaluator who was involved in the project has knowledge of the development history, giving him insight into the desired results and operation. An evaluator who works for the system owner may be reluctant to criticize the product of his supervisor. Since the evaluation is performed following completion of the project, schedule pressure is usually not a cause for a questionable evaluation.

Self-performance addresses the issue of institutional knowledge retention frequently found in contracting approaches. A team member from a previously completed project that is retained on staff vastly aids in updating that project when revisions are needed, whether from exposed defects in the original implementation, from efficiency improvement opportunities based on improved technology, of from evolving needs that the project can meet.

Contracting for SE Performance

Most ITS projects will acquire project services for execution of one or more SE activities by contracting with an outside party. The practices used to issue the contract and the scope of the contract(s) can vary based on the nature of the project and the capabilities of the lead agency.

Most contracts to perform SE tasks are procured using Qualifications Based Selection (QBS). QBS is designed for selection of architects and engineers. The approach attempts to acquire services of the most skilled and qualified staff, with cost being a secondary consideration.

Some Design-Build contracts incorporate Systems Engineering into the design phase of the contract. This approach can result in incentives to underfund SE or use less experienced staff to reduce costs, since cost is usually a factor in Design-Build contracts. While this process is in conflict with the concept of acquiring engineering, services based on skills and qualification, the cost of the SE work is commonly minor in the budget of the overall contract.

Contracting for SE Performance and SE Monitoring

For some ITS projects that require specialized design expertise or that have scope beyond the ability of the owner to monitor, the responsibility to monitor the performance of Systems Engineering will be acquired for one or more SE activities by contracting with an outside party. The practices used to issue the contract and the scope of the contract(s) can vary based on the nature of the project and the capabilities of the lead agency.

Table 16: SE Performance Options

SE Task	Owner Objective	Owner may	Contractor or owner may	Owner must
Involve all stakeholders	Convene stakeholders	Acquire stakeholder engagement help	Interact with stakeholders	Participate and approve findings

SE Task	Owner Objective	Owner may	Contractor or owner may	Owner must
Determine schedule and budget	Acquire adequate funding	Perform estimation and seek cost sharing	Negotiate with stakeholders for consensus	Approve schedule, budget, and agreements
Apply appropriate rigor to SE efforts	Decide on appropriate SE rigor	Acquire SE assessment services	Recommend appropriate SE rigor	Decide on appropriate SE rigor
a. Know what problem is being solved	a. Know what problem is being solved	Acquire project conceptualization assistance	Develop project concepts and goals	Approve project concepts and goals
b. Know how the system will operate	b. Know how the system will operate	Acquire concept of operations support	Develop concept of operations	Approve concept of operations
c. Detail specific functions and behaviors	c. Approve required functions and behaviors	Acquire requirements development support	Develop detailed requirements	Approve detailed requirements
d. Know how the system will be implemented	d. Know how the system will be implemented	Acquire design and implementation services or engineering review services	Design and implement system	Approve design
e. Assure the system is implemented properly	e. Assure the system is implemented properly	Acquire testing services or engineering review services	Test system	Approve test results
f. Assure the problem was solved	f. Assure the problem was solved	Acquire evaluation services	Evaluate system	Approve evaluation results
g. Know how the system will be retired	g. Know how the system will be retired	Acquire decommissioning planning	Determine decommissioning process	Approve decommissioning plan
Assure that all phases are complete	Approve completion of all phases	Acquire project management support	Asses project completeness	Approve completion of all phases
Assess in house skills	Assess in house skills	Acquire capability assessment	Evaluate staff and organizational capability	Select SE performance approach
Track progress quantitatively	Track progress quantitatively	Acquire management services	Acquire capability assessment	Evaluate staff and organizational capability

4.5 Agency Implementation of Systems Engineering Processes

This section will consider how agencies might implement the requirements of FHWA 23 CFR 940 and corresponding FTA Policy as they relate to developing a regional ITS Architecture and the systems engineering analysis for ITS projects. The section will first consider the requirements of 23 CFR 940 and the FTA policy, and then will consider ways in which agencies have implemented these requirements.

4.5.1 Regulation Requirements

23 CFR 940 and the FTA Policy on ITS Architecture and Standards define a set of requirements for regions regarding regional ITS architecture, and a set of requirements for agencies to undertake good, documented engineering processes that help ensure that ITS projects attain the integration and operational objectives embodied in their regional ITS architectures. The requirements of the 23 CFR 940/ FTA/Policy are summarized below.

Regional ITS Architecture

According to FHWA 23 CFR 940, a regional ITS architecture is a "regional framework for ensuring institutional agreement and technical integration for the implementation of ITS projects or groups of projects". Developing a Regional ITS Architecture is a part of ITS Operations Planning (See Section 3.3.2). As specified in the FHWA 23 CFR 940.9 and corresponding FTA Policy Section V, the purpose of a Regional ITS Architecture is to guide the development of ITS projects and be consistent with ITS strategies and projects contained in applicable transportation plans.

Regional ITS Architecture are based on the National ITS Architecture, now called the Architecture Reference for Cooperative Intelligent Transportation (ARC-IT). ARC-IT provides a framework for planning, defining and integrating ITS in the United States. Some other countries have a similar framework for their ITS. Tailoring a Regional ITS Architecture from ARC-IT is best accomplished with the Regional Architecture Development for Intelligent Transportation (RAD-IT) software tool. The Regional ITS Architecture should be defined on a scale commensurate with the scope of the ITS investment in the region. The Regional ITS Architecture development should also include the participation of all the significant ITS stakeholders in the region. A Regional ITS Architecture is a regional plan showing both existing and planned systems and their interfaces for surface transportation system integration.

Refer to the Architecture Use section of the ARC-IT website (https://www.arc-it.net) for more information on developing or updating your Regional ITS Architecture.

ITS Project Systems Engineering Analysis (SEA)

The primary purpose of the Regional ITS Architecture is to develop a plan for how ITS projects are integrated together. ITS projects need to be understood and developed in the context of the region. The FHWA 23 CFR 940.11 (excerpted below in italics) and corresponding FTA Policy Section VI require the following for ITS Project Implementation:

- *All projects funded with highway trust funds shall be based on a systems engineering analysis.*
- *The analysis should be on a scale commensurate with the project scope.*
- *The systems engineering analysis shall include, at a minimum:*
    1. *Identification of portions of the regional ITS architecture being implemented (or if a regional ITS architecture does not exist, the applicable portions of the National ITS Architecture;*

2. *Identification of participating agencies roles and responsibilities;*
3. *Requirements definitions;*
4. *Analysis of alternative system configurations and technology options to meet requirements;*
5. *Procurement options;*
6. *Identification of applicable ITS standards and testing procedures; and*
7. *Procedures and resources necessary for operations and management of the system.*

- *Upon completion of the regional ITS architecture required in §§ 940.9(b) or 940.9(c)*[3]*, the final design of all ITS projects funded with highway trust funds shall accommodate the interface requirements and information exchanges as specified in the regional ITS architecture. If the final design of the ITS project is inconsistent with the regional ITS architecture, then the regional ITS architecture shall be updated as provided in the process defined in § 940.9(f) to reflect the changes.*

One more important area to understand from the Regulation/Policy is the establishment of ITS project administration and oversight. The FHWA 23 CFR 940.13 and corresponding FTA Policy Section VII established project administration and oversight responsibilities to FHWA and FTA respectively. The Regulation/Policy reads as follows:

- For FHWA it says: *Prior to authorization of highway trust funds for construction or implementation of ITS projects, compliance with FHWA 23 CFR 940.11 shall be demonstrated*. For FTA, it says: *Prior to authorization of Mass Transit Funds from the Highway Trust Fund for acquisition or implementation of ITS projects, grantees shall self-certify compliance with sections V and VI. Compliance with this policy shall be monitored under normal FTA oversight procedures, to include annual risk assessments, triennial reviews, and program management oversight reviews as applicable.*

- For FHWA it says: *Compliance with this part will be monitored under Federal-aid oversight procedures as provided under 23 U.S.C. 106 and 133.* For FTA it says: *Compliance with the following FTA Circulars shall also be certified: C5010.1C, Grant Management Guidelines and C6100.1B, Application Instructions and Program Management Guidelines.*

For reference, 23 CFR 940 defines "ITS" as:

> *Intelligent Transportation System (ITS) means electronics, communications, or information processing used singly or in combination to improve the efficiency or safety of a surface transportation system.*

And 23 CFR 940 defines "ITS Project" as:

> *ITS project means any project that in whole or in part funds the acquisition of technologies or systems of technologies that provide or significantly contribute to the provision of one or more ITS user services as defined in the National ITS Architecture.*[4]

[3] 940.9 is the 23 CFR 940 section defining Regional ITS Architecture requirements.

[4] Note the current version of ARC-IT does not define a set of "user services" but is defined around a set of Service Packages. So, consider the definition above to be equivalent to "provision of one or more ITS service packages as defined in the National ITS Architecture".

Generally, state departments of transportation define projects that carry out an ITS user service (as embodied in the ARC-IT Service Packages) as those requiring integration, given that the purpose of the regional ITS architecture is to provide an integration plan for ITS within a region.

The FHWA Division and FTA Regional Offices determine how the systems engineering analysis requirements in the Regulation/Policy should be applied to ITS projects in each region and how compliance should be demonstrated by each project sponsor. Federal oversight is provided based on Stewardship and Oversight agreements that are defined with each state. Several states have established checklists that prompt project sponsors to consider the systems engineering analysis requirements as part of the project development process. Other states have developed template documents, or finished documents for common (and specific) project types (see discussion below). FHWA has also provided a range of Model Systems Engineering Documents that can be tailored to specific project needs by implementing agencies. These model documents cover traffic signal systems, dynamic message signs, CCTV systems, and (in progress at time of publication) traffic sensors. Each of these approaches is further discussed below. Contact the ITS specialist in your FHWA Division Office or FTA Regional Office for more information.

4.5.2 Implementing the SE Process

Agencies have applied a variety of approaches and tools to support the implementation of the SE process in order to address the requirements described above. These include use of a Systems Engineering Review Form (SERF), developing sample systems engineering documents for use in commonly deployed systems, and, at the federal level, use of Model Systems Engineering Documents. All three of these approaches are discussed below.

Systems Engineering Review Form (SERF)

Some states have created a form that can serve as a checklist to address the SEA requirements of 23 CFR 940.11. Project sponsors fill out the form as a way to indicate their compliance with the requirements. The forms provide responses to the seven requirements for systems engineering analysis within 23 CFR 940.11. Some states (e.g. California and New Jersey) call this a SERF, other states (e.g. Arizona) call it a Systems Engineering Checklist. There is no specific format for these forms, with each state that has developed one creating something slightly different. Michigan DOT and Arizona DOT have created pdf forms that can be filled out. Caltrans and NJDOT have Word documents that can be filled out and submitted. Some states provide explanation for each question to clarify the information that should be collected. Since these are "checklists", what is often requested are links to additional documentation (e.g. concept of operations or requirements documents) that have been created to support the project.

In addition to the seven requirements listed above, some states include sections that provide general descriptions of the project, or connections from the project to planning documents (such as a Metropolitan Transportation Plan). Some states tie the form/checklist to an assessment of risk (which was addressed in Section 4.2). The idea of tying the checklist to risk is to identify if the project falls into a category that is exempt from systems engineering analysis, is a low risk project that requires minimal analysis, or is a higher risk project requiring a more complete delineation of the systems engineering process.

In the absence of a state form to use, the following discussion provides the seven requirements along with some clarifying information on what to collect:

1. Identification of portions of the Regional ITS Architecture being implemented:

Contact your MPO or State DOT to get this information from your Regional ITS Architecture ("RA"). Review the portions of the RA that define the project. If your architecture has a defined project architecture for your project, then you can copy that. If not consider the following portions of the architecture as they relate to your project:

- Service Packages
- Elements
- Information flows

If there is no information in your RA, arrange with your MPO to provide them this information when your project is designed; they will use it in the next update of the RA.

2. Identification of participating agencies roles and responsibilities:

Can you identify all stakeholders that must participate in the implementation or operations phases of this project? What are their roles/responsibilities? Have they committed to the responsibilities? Some of this information might appear in your RA (e.g., "Operational Concepts" or other sections). If this will be defined in later phase of the project (e.g., Concept of Operations), the RA may be a good source to start definition.

3. Requirements definitions:

Are the system requirements (functional and performance) already well-defined in writing? If yes, indicate where they can be found (e.g., Standard or Specs). If they will be defined in later phase of the project, the applicable high-level functional requirements in the RA may be a good starting point for writing them. The focus is on "what" functions must be performed – not on "how" the technology will be used to perform them.

4. Analysis of alternative system configurations and technology options to meet requirements:

Have you considered alternative designs yet? This could include system configurations, different organizational roles; alternative hardware, software, or communications technology. Alternatives may be considered at several points in a project lifecycle. For large systems, concept exploration considers broad alternative concepts early as defined in Section 3.3.3. More detailed design/technology choices wait until High-Level design as described in Section 3.3.7.

5. Procurement options:

Have you considered different procurement options for each of the project phases (design, implementation, operation, and management)? These options could include: off-the-shelf vs. custom, lease vs buy, fixed-price vs. cost-reimbursable, purchase-of-services contract etc. Procurement options must consider the level of staff technical expertise, existing agency procurement practices, who will be the project manager, and what level of systems engineering expertise is needed for the project. A more detailed discussion of procurement options and considerations for support during development can be found in Section 4.4.

6. Identification of applicable ITS standards and testing procedures:

Do you know yet if any ITS Communications Standards are applicable to this project? If they are applicable, will you use them? If your RA identifies specific Information Flows, you can ask your MPO to produce a "Standards Report" for those Flows; it will identify ITS Standards to consider.

7. Procedures and resources necessary for operations and management of the system:

Can you identify all stakeholders that must participate in operations, management and maintenance of the system throughout its life cycle? What are the roles, responsibilities, and resources required from each stakeholder? Examples include: money, special equipment, staff time, O&M capabilities, special expertise, provision of data, and many more. You should consider hardware, software, and communications issues.

In general, these checklists are meant to be used for state DOT projects, but where Highway Trust Fund money is used for county or municipal ITS projects, Division offices may recommend use of the statewide forms (which have been coordinated with the Division Office) as the basis for addressing systems engineering.

5 Systems Engineering Resources

This chapter lists many excellent books, reports, training courses, and other systems engineering resources that you can use to learn more about any of the systems engineering topics that are introduced in this document.

5.1 ITS Specific Publications

FHWA 23 CFR 940/FTA Policy: https://www.ops.fhwa.dot.gov/its_arch_imp/policy.htm

Guidance from Federal Transit Administration (FTA) on their policy: https://www.transit.dot.gov/research-innovation/national-its-architecture-consistency-policy-transit-projects

Systems Engineering Review Form (SERF), Caltrans Local Assistance Procedures Manual - https://dot.ca.gov/-/media/dot-media/programs/local-assistance/documents/lapm/c07/07i.pdf

The Guide to Contracting ITS Projects, National Cooperative Highway Research Program (NCHRP), https://onlinepubs.trb.org/onlinepubs/nchrp/nchrp_rpt_560.pdf

Florida DOT SEMP and Systems Engineering Templates: https://www.fdot.gov/traffic/its/projects-deploy/semp.shtm

Systems Engineering for ITS Projects Memorandum, Federal Highway Administration, July 2022, https://ops.fhwa.dot.gov/int_its_deployment/sys_eng.htm [ops.fhwa.dot.gov]

Model Systems Engineering Documents

- Traffic Signal Systems, https://ops.fhwa.dot.gov/publications/fhwahop19019/index.htm
- Dynamic Message Signs, https://ops.fhwa.dot.gov/publications/fhwahop18080/index.htm
- Closed-Circuit Television, https://ops.fhwa.dot.gov/publications/fhwahop18060/index.htm
- Transportation Sensors and Detection Systems, (forthcoming)

5.2 General Systems Engineering References

International Council on Systems Engineering (INCOSE): https://www.incose.org

ISO/IEC/IEEE 15288:2015, Systems and software engineering-System life cycle processes: https://www.iso.org/standard/63711.html

Risk-Based Transportation Asset Management: Evaluating Threats, Capitalizing on Opportunities, FHWA, June 2012: https://www.fhwa.dot.gov/asset/pubs/hif12035.pdf

National Aeronautics and Space Administration (NASA) Systems Engineering Handbook: https://www.nasa.gov/connect/ebooks/nasa-systems-engineering-handbook

Department of Defense (DOD) Systems Engineering Plan (SEP) Preparation Guide: https://www.acq.osd.mil/ds/se/publications/pig/sep_prepguide_v1_2.pdf

Buede D. M., The Engineering Design of Systems: Models and Methods, Wiley Inter-Science, 2000

Applying Scrum Methods to ITS Projects, Final Report — August 2017, Publication Number: FHWA-JPO-17-508

Federal Highway Administration (FHWA) Office of Operations page on Configuration Management: https://ops.fhwa.dot.gov/freewaymgmt/pubs.htm#config_mgmt

5.3 Systems Engineering Training

Below is a list of organizations providing systems engineering training focused on transportation. The only general systems engineering resource is INCOSE.

National Highway Institute (NHI) – refer to https://www.nhi.fhwa.dot.gov/home.aspx, search on "systems engineering", there are both Instructor-led Training (ILT) and Web-based Training (WBT)..

Professional Capacity Building (PCB) ITS JPO Program – refer to https://www.pcb.its.dot.gov/default.aspx, search PCB website for "systems engineering".

Consortium for innovative Transportation Education (CITE) – refer to https://www.citeconsortium.org/ , search CITE website for "systems engineering".

U.S. DOT ARC-IT National ITS Reference Architecture – refer to https://www.arc-it.net/html/resources/training.html, delivery options range from convenient web-based training for individuals to facilitated workshops for one or more regions.

International Council on Systems Engineering (INCOSE) – refer to https://www.incose.org/. INCOSE is a not-for-profit membership organization founded to develop and disseminate the interdisciplinary principles and practices that enable the realization of successful systems. INCOSE is designed to connect SE professionals with educational, networking, and career-advancement opportunities in the interest of developing the global community of systems engineers and systems approaches to problems.

5.4 Systems Engineering Tools

The following tools relevant to systems engineering are provided by USDOT.

The Architecture Reference for Cooperative and Intelligent Transportation (ARC-IT) is used as a template to create regional ITS architectures that are tailored for a specific state, metropolitan area, or other region of interest (e.g., a major corridor or a National Park). ARC-IT provides the fundamental building blocks: the physical objects (subsystems and terminators), interfaces (as defined by the information flows), service packages, functional objects, and functional requirements that are selectively included in the regional ITS architecture and customized as necessary to fully reflect the envisioned regional transportation system.

The Regional Architecture Development for Intelligent Transportation (RAD-IT) software provides an easy way to personalize and customize ARC-IT for a specific region.

The Systems Engineering Tool for Intelligent Transportation (SET-IT) software provides extensive drawing capabilities to produce project architectures and support Concept of Operations and Requirements definition. SET-IT content is based on ARC-IT and is compatible with the RAD-IT software.

6 Systems Engineering Documentation

This chapter provides a series of templates for each of the systems engineering-related documents that could be developed on a specific project.

6.1 Project Management Plan

6.1.1 Purpose of this Document

The Project Management Plan (PMP) is the governing document for the conduct of a project. All other plans and technical documents follow from the Project Plan. Most agencies have project management procedures which call for the creation of a Project Plan. Obviously, those need to be followed. The PMP described here shows the most commonly needed elements of a PMP.

The purpose of the PMP is to define and describe all of the tasks that need to be performed to accomplish the project. Each task is described in enough detail that the assigned personnel can do it satisfactorily. It is also critical that the products of each task, the schedule for each task, and the available budget are established. Further, the assigned personnel need to “buy-in” to this plan and believe they can do their task on time and within budget.

Also, the PMP establishes and identifies the environment in which the project will operate. It identifies all the players in the project including management, responsible teams or organizations for each task, supporting organizations, and all stakeholders.

6.1.2 Tailoring This Document to Your Project

Although almost always required, the size of the PMP can vary considerably depending on the complexity of the project and the breadth of its environment. If needed, the PMP can be supplemented with a variety of supporting plans. Depending on complexity, it may be more efficient to document all this support plan information in the PMP itself.

The more expensive a project, the more that management will want to see that it is well planned.

The technical complexity of a project translates directly into technical risk that must be managed through good planning.

The stakeholders will use the PMP to understand and plan their roles and it provides a means for them to review and comment on their ability to perform the needed tasks.

6.1.3 Checklist: Critical Information

- ☑ Are all of the necessary tasks included in the plan [perhaps in the form or a Work Breakdown Structure] along with identification of the personnel or team that is responsible for performing the task?
- ☑ Is the sequence of the tasks correct so that the necessary precursor work is done for each task?
- ☑ Is the budget assigned to each task sufficient to get the task done as defined? Does the team that will perform the task agree?
- ☑ Is the scheduled time period for each task sufficient to get the work done as defined? Does the team that will perform the task agree?
- ☑ Are the necessary stakeholder organizations identified? Are their roles defined and agreed to?
- ☑ Are all products of each task [documents, meetings, hardware and software] identified? Or, alternatively, is a task defined to identify those products?
- ☑ Are any supporting plans required to supplement the PMP? Is their preparation defined as a task?
- ☑ Do all stakeholders, including management, approve the PMP?

6.1.4 Template

PROJECT MANAGEMENT PLAN TEMPLATE

SECTION	CONTENTS
Title Page	The title page should follow the Transportation Agency procedures or style guide. At a minimum, it should contain the following information: ▪ PROJECT MANAGEMENT PLAN FOR THE [insert name of project] AND [insert name of transportation agency] ▪ Contract number ▪ Date that the document was formally approved ▪ The organization responsible for preparing the document ▪ Internal document control number, if available ▪ Revision version and date issued
1.0 Purpose of Document	The purpose of this document is the plan for execution of the project including defining all necessary tasks and their products.
2.0 Scope of Project	This section provides a brief description of the planned project and the objectives of the system to be built. Special emphasis is placed on the project's complexities and challenges to be addressed by the project's managers. This section defines the project's relationship to the applicable regional ITS architectures and, if necessary, to the National ITS Architecture. It also defines the relationship of the project's system to other systems with which it interfaces, either physically [with a data interface] or operationally. This section also describes the environment in which the project will operate. It identifies the organizational structures that encompass all stakeholders and gives a brief description of the role to be played by each stakeholder. This section identifies organizations within the owning agency that are stakeholders in this project. It also identifies any external agencies [especially agencies with a system that interfaces with this project's system] that are project stakeholders. A subsequent project management task is to identify individuals within those organizations and agencies who will represent their organization among the project's stakeholders. It is especially important that the Project Plan identify the system's owners who are building the system and the customer for whom the system is being built. The section also identifies any existing management work groups and multi-disciplinary technical teams to be used to support the project.

SECTION	CONTENTS
3.0 Project Tasks	This section is the heart of the PMP. It defines each task of the project in terms of its inputs, approach, and outputs. **Inputs:** Identification of the inputs to each task. Inputs can be a variety of things, including, but certainly not limited to: ▪ Documents from outside the project or from other tasks of the project, that are meant to guide the activities of this task, such as, a regional ITS architecture and other planning documents ▪ Directions from others that guide the efforts of the team performing this task, such as directions from a multi-agency steering committee established for this project ▪ Meetings with others to be conducted by the team performing this task, such as periodic status meetings with the project manager's organizational management. ▪ Products, other than documents, from other tasks that are a necessary precursor to the performance of this task. For example, a product from an integration task is a software and hardware component that is a necessary input to a verification task. **Approach:** A description of the approach to be taken by the team performing the task. This includes a description of the steps involved in developing the outputs of the task, which might include the definition of sub-tasks. This description may include identification of procurement activities that need to be taken in this task. For systems engineering and design tasks, this description may be expanded as necessary in the Systems Engineering Management Plan, which, of course, would be an activity and output of one of the tasks. **Outputs:** A description of the products of the task. As with inputs, the outputs may take many forms, including, but not limited to: ▪ Documents to be produced by the task team, such as, specifications, Verifications Plans, and the SEMP. ▪ Meetings, including management meetings and technical reviews ▪ Other products such as software code, procured hardware, and integrated or verified sub-systems ▪ Attendance at meetings conducted by others, such as periodic meetings of a multi-agency steering committee
4.0 Work Breakdown Structure and Task Budgets	This section provides a hierarchical structure of all tasks and sub-tasks of the project, identifying the name of the task or sub-task, the allocated budget, and the team or organization with the authorization and responsibility to perform the task. The budget may not be allocated to each sub-task but may be allocated to a higher-level group of sub-tasks, tasks, or group of tasks, as necessary to manage the project.

SECTION	CONTENTS
5.0 Schedule Constraints	A project's schedule is developed in two steps, and this section, at a minimum, includes information to define the initial step of schedule development. The two steps in development of a project's schedule are: ▪ **Step one:** identification of external schedule constraints. These may include a not-earlier-than start date, a not-later-than completion date, a date tied to the completion of an external system, or the date a needed resource is available. In general, these schedule constraints come from outside the project and are not within the control of the project's management. ▪ **Step Two:** development of a schedule for each task, for each sub-task, and for each output of a task. This schedule is under complete control of the project's management by a variety of means, including the assignment of more or fewer resources. This schedule takes into account the necessary precursors [inputs] to each task or sub-task. The schedule in this section of the Project Plan includes the output of step one and may either include the complete schedule of step two or identify this as an output of one of the tasks.
6.0 Deliverable Requirements List	This section is, as much as possible, a complete and precise list of the tangible deliverables of each and every task. In general, a tangible deliverable may include, from the list of outputs of a task: ▪ Documents, especially documents to be reviewed by stakeholders, and documents to be used after the system is built ▪ Meetings and reviews to be attended by project stakeholders ▪ Other products, such as deliverable hardware [by name, part number, and quantity] and deliverable software products, such as source code and executables It may not be possible to completely and precisely define each and every deliverable at the time the Project Plan is prepared. For instance, the Project Plan may state that design specifications are required but the identification of specific documents may have to wait until the sub-systems are defined in the high-level design task.
7.0 Referenced Documents	This section lists the applicable documents that are inputs to the project [that is, are needed by but not produced by the project]. Such documents may include: the regional ITS architecture description, planning documents describing the project, agency procedures to be followed, standards, specifications, and other descriptions of interfacing external systems. Other applicable documents may be required by a specific project.

6.2 Systems Engineering Management Plan (SEMP)

6.2.1 Purpose of this Document

The SEMP [Systems Engineering Management Plan], may be needed to supplement the details of the Project Plan. When used, the SEMP focuses on the technical plan of the project and the systems engineering processes to be used for the project. Its purpose is to detail out those engineering tasks; especially to provide detailed information on the processes to be used. Preparation of a SEMP is most important if the project involves development of custom software. The engineering tasks of producing custom software [from requirements, through design implementation, integration, and verification] are very complex, and are new to many transportation engineers.

Given the level of process detail needed in the SEMP, it is often written in two steps. In the first step, the framework for the document is prepared, usually by the project management staff. Enough detail is included to identify all the needed tasks [including analysis tasks] and any important constraints on the performance of a task [such as use of a specific systems engineering and design methodology]. In the second step, the various sections of the SEMP framework are completed, this time by the team that will perform each task. For instance, the requirements team provides details on the analysis and the tools used to manage requirements. The design team provides details on use of the software design methodology. The software coder provides details on configuration management of the software code. The verification team provides details on their verification methods.

6.2.2 Tailoring this Document to Your Project

Many ITS projects may not need a SEMP; the Project Plan may be sufficient. Among the project complexities that make preparation of a SEMP desirable are:

- Inexperience of the system's owner's project team in the systems engineering tasks and processes
- A larger number of stakeholders and the degree of their involvement in the various systems engineering processes and tasks
- The need to develop custom software applications
- A project where the solution is not well understood and is not generally obvious

6.2.3 Checklist: Critical Information

- ☑ Are all the technical challenges of the project addressed by the systems engineering processes described in the SEMP?
- ☑ Does the SEMP describe the processes needed for requirements analysis?
- ☑ Does the SEMP describe the design processes and the design analysis steps required for an optimum design?
- ☑ Does the SEMP clearly identify any necessary supporting technical plans, such as a Verification Plan or an Integration Plan? Does it define when and how they will be written?
- ☑ Does the SEMP spell out stakeholder involvement when it is necessary?
- ☑ Does the SEMP identify all the required technical staff and development teams? Does it identify the technical roles to be performed by the system's owner, project staff, stakeholders, and the development teams?
- ☑ Does the SEMP cover the interfaces between the various development teams?

6.2.4 Template

SYSTEMS ENGINEERING MANAGEMENT PLAN TEMPLATE

<table>
<tr><th>Section</th><th>Contents</th></tr>
<tr><td>Title Page</td><td>The title page should follow the Transportation Agency procedures or style guide. At a minimum, it should contain the following information:

• SYSTEMS ENGINEERING MANAGEMENT PLAN FOR THE [insert name of project] AND [insert name of transportation agency]
• Contract number
• Date the document was formally approved
• The organization responsible for preparing the document
• Internal document control number, if available
• Revision version and date issued</td></tr>
<tr><td>1.0 Purpose of Document</td><td>This section is a brief statement of the purpose of this document and the plan for the systems engineering activities with special emphasis on the engineering challenges of the system to be built.</td></tr>
<tr><td>2.0 Scope of Project</td><td>This section gives a brief description of the planned project and the objectives of the system to be built. Special emphasis is placed on the project’s complexities and challenges that must be addressed by the systems engineering efforts.

This section also describes the environment in which the project will operate. It identifies the organization structures that encompass all stakeholders. It gives a brief description of the role to be played by each stakeholder. This includes ad hoc and existing management work groups and multi-disciplinary technical teams that should be formed or used to support the project. Such teams are critical to reaching successful system deployment.

This section defines the general process for developing the SEMP, including the draft framework version prepared by the transportation agency or their Systems Engineer and the complete version prepared in conjunction with the Systems Engineer and Development Teams.</td></tr>
</table>

Section	Contents
3.0 Technical Planning and Control	This section lays out the plan for the systems engineering activities. It must be written in close synchronization with the project's Project Plan. Unnecessary duplication between the Project Plan and the SEMP should be avoided. The purpose of the section is to describe the activities and plans that will act as controls on the project's systems engineering activities. For instance, this section identifies the products of each systems engineering activity, such as, documentation, meetings, and reviews. Some of these plans may be completely defined in the SEMP [in the framework or the complete version]. For other plans, the SEMP may only define the requirements for a particular plan. The first set of activities/plans listed below relate primarily to the successful management of the project. These should be created for almost any project. They may be a part of the Project Plan, but if not should be part of the SEMP. • **Work Breakdown Structure** [WBS] [also included in the Project Plan] is a list of all tasks to be performed on a project, usually broken down to the level of individually budgeted items. • **Task Deliverables** is a list of the required products of each task in the WBS, including documents, software, and hardware. • **Task Decision Gates** is a list of critical activities that must be satisfactorily completed before a task is considered complete. • **Reviews and Meetings** is a list of all meetings and reviews of each task • **Systems Engineering Schedule** is a schedule of the systems engineering activities that shows the sequencing and duration of these activities. • **Risk Management Plan** addresses the processes for identifying, assessing, mitigating, and monitoring the risks expected or encountered. These next set of activities should be considered for inclusion in the SEMP if they have particular importance to the project, but will likely not be included in most SEMPs. • **Task Resources** is identification of resources needed for each task in the WBS, including for example, personnel, facilities, and support equipment. • **Task Procurement Plan** is a list of the procurement activities associated with each task of the WBS, including hardware and software procurement and, most importantly, any contracted services, such as systems engineering services or development services • **Critical Technical Objectives** is a summary of the plans for achieving any critical technical objectives that may require special systems engineering activities. It may be that a new software algorithm needs to be developed and its performance verified before it can be used. Or a prototyping effort is needed to develop a user-friendly operator interface. Or a number of real-time operating systems need to be evaluated before a procurement selection is made. This type of effort is not needed for all projects

Section	Contents
4.0 Systems Engineering Process	This section describes the intended execution of the systems engineering steps used to develop the system. These steps are generically described in Chapter 3. The SEMP describes the processes specifically needed for a project. It defines them in sufficient detail to guide the work of the systems engineering and development teams. This section will contain a description of the systems engineering procedures tailored to the specific project. There are four areas of analysis that are usually described, depending on the nature of the project: • **System Requirements Analysis** describes the methods to be used to prepare the Concept of Operations and the top-level system requirements documents. The analysis techniques that may be used include: peer reviews, working groups, scenario studies, simulation, and prototyping. The amount of analysis required increases with the risk of the specific requirement. The process for approving the resulting documents will be described, including who is involved, whether technical reviews are necessary, and how issues and comments are resolved so the baseline can be defined • **Sub-system [Functional] Analysis** describes the methods to be used to identify sub-systems and to allocate the system [top-level] requirements to the sub-systems. It is often necessary, at this step, to expand the top-level requirements into a complete description of the functions of the system, for instance, details of an operator interface. It also may be necessary, at this time, to define internal interfaces [sub-system to sub-system] to the same level of detail as the external interfaces [interfaces to other systems]. The SEMP should describe the methods for analysis and the tools required. Budget and schedule constraints, as well as completion criteria, should be included • **Design Synthesis** describes the methods to be used by the development teams to translate the functional requirements into a hardware and software design. A number of tools and methodologies exist for this. The specific ones to be used by the development team should be identified, along with the necessary resources. Describe the products to be produced as this process unfolds and the design review steps to be taken • **System Analysis** describes the methods to be used for any required technical trade-off studies, cost/benefit decisions, and risk mitigation alternative analysis. The methodologies used should provide a rigorous basis for selecting an alternative, a quantifiable basis for comparing the technical, cost, and schedule impacts of each alternative, and comprehensive description of the risks involved with each alternative.

Section	Contents
5.0 Transitioning Critical Technologies	This section will describe the methods and processes to be used to identify, evaluate, select, and incorporate critical technologies into the system design. If the project includes critical technologies, then this section will represent an area of importance to the project, since it is one of the major efforts of risk management. For many projects, using existing, well-defined technologies, this section will not be relevant to the SEMP. The need for a critical technology may be based on a performance objective. It may also be based on other factors; the desire to reduce acquisition or maintenance costs; the need to introduce standard compliance; or the need to meet an operational objective. In some cases, the need may move away from a technology that is obsolete and no longer supported by industry. Identification of candidate technologies hinges on a broad knowledge of the technologies and knowledge of each technology's status and maturity. In other words, build on a thorough understanding of the pros and cons of each available technology. Obtaining the resource[s] capable of performing this step is one of the major risks encountered by project management. Sufficient analysis of the risks and benefits of a particular technology may become a major effort involving acquiring the technologies, modifying the technology to meet system requirements, and developing methods to test and evaluate the various technologies that need to be considered. Each of these steps can introduce considerable risk. Finally, incorporation of a technology into an operational system may involve considerable work, especially establishing the support and maintenance environment for the technology. All of these aspects of technology introduction, especially introduction of novel technology, need to be carefully and fully addressed in the SEMP.
6.0 Integration of the System	This section describes the methods to be used to integrate the developed components into a functional system that meets the system requirements and is operationally supportable. The systems engineering steps to be detailed here include: integration, verification, transition, and the training necessary to support operations & maintenance. Plans for validation of the system should also be covered. For each step, the resources [tools and personnel] are identified and products and criteria for each step defined.

Section	Contents
7.0 Integration of the Systems Engineering Effort	This section addresses the integration of the multi-disciplinary organizations or teams that will be performing the systems engineering activities. Obviously, the larger the number of such organizational teams, the more important the integration of their efforts is. Each team will have both primary and support tasks from the WBS. Each team will have to be aware of the activities of other teams, especially those activities that immediately precede or follow their own primary tasks. Representatives of most teams will have to be involved in critical technical reviews, and in the review of baseline documentation. Likewise, up-front teams [e.g. requirements and design] must be available to support the ending activities, such as, integration, verification, deployment, and training.
8.0 Applicable Documents	This section lists the applicable documents which are inputs to the project [i.e., needed but not produced by the project]. Such documents may include: the regional ITS architecture description, planning documents describing the project, agency procedures to be followed, standards & specifications, and other descriptions of interfacing external systems. Other applicable documents may be required by a specific project.

The SEMP can also include references to a number of different plans that are designed to address specific areas of the systems engineering activities. Most of the plans below, if they are needed, will be developed as separate documents. The SEMP will usually give guidance for their preparation. Sometimes the plans are included in the SEMP. The unique characteristics of a project will dictate their need.

- **Stakeholder Outreach Plan** describes how stakeholders will be engaged through the steps of the project. This plan would be particularly relevant if the project involves multiple stakeholder organizations.
- **Configuration Management Plan** describes the development team's approach and methods to manage the configuration of the system's products and processes. It will also describe the change control procedures and management of the system's baselines as they evolve. This plan is often included in the SEMP or project plan. On larger projects it might be a standalone document.
- **Software Development Plan** describes the organization structure, facilities, tools, and processes to be used to produce the project's software. Describes the plan to produce custom software and procure commercial software products. Relevant if there is new software to be developed.
- **Hardware Development Plan** describes the organization structure, facilities, tools, and processes to be used to produce the project's hardware. It describes the plan to produce custom hardware [if any] and to procure commercial hardware products. Since most ITS projects include procurement of commercial hardware products, some aspect of this plan is widely applicable.
- **Technology Plan** if needed, describes the technical and management process to apply new or untried technology to an ITS use. Generally, it addresses performance criteria, assessment of multiple technology solutions, and fallback options to existing technology.
- **Interface Control Plan** identifies the physical, functional, and content characteristics of external interfaces to a system and identifies the responsibilities of the organizations on both sides of the interface. This plan, if needed, will be created during the design step. The SEMP would discuss the need and timing for its development.
- **Technical Review Plan** identifies the purpose, timing, place, presenters & attendees, subject, entrance criteria, [a draft specification completed] and the exit criteria [resolution of all action

items] for each technical review to be held for the project. If needed, this could be an expansion on the Reviews and Meetings discussed above.

- **System Integration Plan** defines the sequence of activities that will integrate software components into sub-systems and sub-system into entire systems. This plan is especially important if there are many sub-systems produced by a different development team
- **Verification Plan** identifies how the requirements will be tested. This plan is almost always part of the systems engineering effort, sometimes written along with the requirements specifications, and sometimes as a standalone document.
- **Verification Procedures** are developed by the Development Team and this defines the step by step procedure to conduct verification and must be traceable to the verification plan.
- **Installation Plan** or **Deployment Plan** describes the sequence in which the parts of the system are installed [deployed]. This plan is especially important if there are multiple different installations at multiple sites. A critical part of the deployment strategy is to create and maintain a viable operational capability at each site as the deployment progresses
- **Operations & Maintenance Plan** defines the actions to be taken to ensure that the system remains operational for its expected lifetime. It defines the maintenance organization and the role of each participant. This plan must cover both hardware and software maintenance
- **Training Plan** describes the training to be provided for both maintenance and operation
- **Data Management Plan** describes how and which data will be controlled, the methods of documentation, and where the responsibilities for these processes reside
- **Other plans** that might be included are for example, a Safety Plan, a Security Plan, a Resource Management Plan, and/or a Validation

6.3 Configuration Management Plan (CMP)

6.3.1 Purpose of this Document

A Configuration Management Plan is one of the more common technical and management plans needed to supplement the Project Plan and the Systems Engineering Management Plan. Preferably, the agency has an established CM process in place. If that is the case, then the agency's CM Plan only needs to be supplemented with project specific information, such as organization, products, and schedules. If an agency CM plan does not exist, then a project specific CM plan is developed that focuses on managing the specific project.

Configuration management is as much of a concern after the project's system is deployed [because of maintenance and upgrades] as it is during development. If possible, the CM Plan should be written to handle both phases, development and operations.

Additional information on Configuration Management is found in Section 3.4.2 of this Document.

6.3.2 Tailoring this Document to Your Project

The major challenge in writing a useable CM Plan is to create a CM process that is commensurate with the size and scope of the project. Configuration Management can become very labor intensive and expensive. Too often, the expense of CM overrides the value of CM and it falls by the wayside. This problem is especially prominent in Change Control Management where the process is made so complex and difficult that it stifles the willingness of the developers to participate in it. Too many levels of change approval, or too large of a group that must approve a change, are common problems. Change approval should be focused on finding workable solutions and not insisting on the perfect solution every time.

The CM processes obviously become more complex when the project involves development of custom software. However, maintaining the configuration of the Concept of Operations and the Requirements Specification is applicable to almost any project.

6.3.3 Checklist: Critical Information

- ☑ Does the agency have an existing Configuration Management Plan that must be used by the project?
- ☑ Does the Organization section of the CM Plan identify and describe the roles of all necessary participants? Does it include stakeholders from outside the project staff?
- ☑ Have all named participants been notified of their role? Do they and their organization understand and accept this responsibility?
- ☑ Does the Configuration Item Identification section specifically name each item [documents/hardware/software] that will be placed under configuration control? Alternatively, does it identify the types of items to be placed under configuration control?
- ☑ Does the Configuration Item Identification section describe when each item is placed under configuration control [baseline]? Does it define the process steps that must occur before this happens?
- ☑ Does the Change Management section describe the process for preparing and submitting a proposed change request?
- ☑ Does the Configuration Status Accounting section describe the establishment of a configuration repository where the current versions of all items are kept? Are they made available to project personnel and other stakeholders?

6.3.4 Template

CONFIGURATION MANAGEMENT PLAN TEMPLATE

SECTION	CONTENTS
Title Page	The title page should follow the Transportation Agency procedures or style guide. At a minimum, it should contain the following information: ▪ CONFIGURATION MANAGEMENT PLAN FOR THE [insert name of project] AND [insert name of transportation agency] ▪ Contract number ▪ Date that the document was formally approved ▪ The organization responsible for preparing the document ▪ Internal document control number, if available ▪ Revision version and date issued
1.0 Purpose of Document	This section is a brief statement of the purpose of this document. It defines the processes for establishing and maintaining configuration control of the products and documentation of the project. These processes are meant to remain in place for the life of these products and documents, i.e., through development, operations, and upgrades.
2.0 Scope of Project	This section gives a brief description of the planned project and the purpose of the system to be built. This section may be lifted from earlier documents. It is important only to people [stakeholders] who will be introduced to the project for the first time by this document.
3.0 Organization, Roles and Responsibili-ties	This section identifies the organizational structure needed to manage and perform configuration management for this project. If possible, the members of the configuration management organization are identified by name. The section then defines the role of each member of the organization. Typically, the organization includes: ▪ A CM manager who supervises all CM activities ▪ CM staff, reporting to the manager and who are responsible for the performance of the CM processes ▪ A change management board who, after a configuration item is an approved baseline item, approves/rejects all proposed changes to that item This section also may identify any configuration management tools to be used by the project to support the CM processes, such as, a software configuration management tool.

SECTION	CONTENTS
4.0 Configuration Item Identification	This section defines the process to identify those items [outputs of the tasks of the project] which will be placed under configuration management. It also identifies when those items are made a baseline and placed under CM control. Such items include documents as well as hardware and software products. The process for placing an item under CM control is general in nature. The specifics of the process for each item produced by this project are defined in the plan. For instance, the process for placing the project's High Level Design specification may involve: review of the completed document by an identified set of stakeholders, an in-depth design review by those same stakeholders, and resolution and incorporation of all stakeholder comments. The review makes sure that all requirements are traced into the design. It also ensures that appropriate and sufficient trade-off studies were completed concerning alternate designs. In other words, only when the stakeholders are satisfied with a particular CM item is that item declared a baseline, placed under change management control and approved for use in subsequent steps in the development of the system.
5.0 Change Management	This section defines the formalized process for making a change to a baseline CM item. This process generally involves generation of a change request, an in-depth analysis of the impacts of the proposed change and then formal approval [or rejection] by the change management board. The plan defines how proposed changes are to be documented. How they are submitted to the CM staff. How the staff prepares them for preliminary review by the change management board. How and when the board conducts this preliminary review. How the need [as determined by the board] for further analysis is recorded. How and when this analysis is presented to the board. Finally, how the disposition of the change request is documented and distributed by the staff.
6.0 Configuration Status Accounting	This section describes the steps to be taken by the CM manager and staff that will keep the other participants in the project aware of the configuration of the various outputs and products of the project. They will follow these defined processes to make the current configuration of documents and products known, and available, in a timely manner. They will make the status of any proposed changes known as the changes are being considered by the change management board. Today, for both documents and software products, this means having procedures for keeping and making available electronic files that contain the currently approved version of the item. They will make those files available to other project participants.

SECTION	CONTENTS
7.0 Configuration Audits	This section defines the process, and the application of that process, for verifying the configuration of a hardware or software product. This process will be invoked during verification to ensure the product version being verified is known and is accurately described by its documentation. The processes describe how and by whom this audit is to be conducted.
8.0 Applicable Documents	This section lists other documents that are referenced in this Configuration Management Plan.

Based on EIA 649 National Consensus Standard on Configuration Management

6.4 Concept of Operations Template

6.4.1 Purpose of this Document

The Concept of Operations is a description of how the system will be used. It is non-technical, and presented from the viewpoints of the various stakeholders. This provides a bridge between the often-vague needs that motivated the project to begin with and the specific technical requirements. There are several reasons for developing a Concept of Operations.

- Get stakeholder agreement identifying how the system is to be operated, who is responsible for what, and what the lines of communication are
- Define the high-level system concept and justify that it is superior to the other alternatives
- Define the environment in which the system will operate
- Derive high-level requirements, especially user requirements
- Provide the criteria to be used for validation of the completed system

6.4.2 Tailoring this Document to Your Project

The greater the expected impact on operations, the more detailed the Concept of Operations needs to be. For example, automating operations that were formerly manual or integrating activities that were formerly independent will require the involvement of the various operators, clear and detailed description of their new procedures, and possibly examination of alternative approaches. This is especially true when building a regional system by integrating existing local systems. Local operations will usually change after integration, for compatibility and to take advantage of newly available regional resources.

For a simple system that requires little operator involvement and no coordination, this document may only be a couple of pages long. The key is to describe all possible system modes, both normal and failure, as seen by each stakeholder.

Figure 29 shows two standard outlines for the Concept of Operations. As shown, the ANSI outline lends itself to new systems while the IEEE standard is well suited to system upgrades with its initial focus on the current system.

6.4.3 Checklist: Critical Information

- ☑ Is the reason for developing the system clearly stated?
- ☑ Are the objectives of the project clearly stated?
- ☑ Are all the stakeholders identified and their anticipated roles described? This should include anyone who will operate, maintain, build, manage, use, or otherwise be affected by the system.
- ☑ Are alternative operational approaches [such as centralized vs. distributed] described and the selected approach justified?
- ☑ Is the external environment described? Does it include required interfaces to existing systems?
- ☑ Is the support environment described? Does it include maintenance?
- ☑ Is the operational environment described?
- ☑ Are there clear and complete descriptions of normal operational scenarios?
- ☑ Are there clear and complete descriptions of maintenance and failure scenarios?
- ☑ Do the scenarios include the viewpoints of all involved stakeholders? Do they make it clear who is doing what?
- ☑ Are all constraints on the system development identified?

Two different industry standards provide suggested outlines for Concepts of Operations: ANSI/AIAA-G-043-1992 and ISO/IEC/IEEE 29148. Both outlines include similar content, although the structure of the 29148 outline lends itself more to incremental projects that are upgrading an existing system or capability. The ANSI/AIAA outline is focused on the system to be developed, so it may lend itself more to new system developments where there is no predecessor system. Successful Concepts of Operation have been developed using both outlines.

ANSI/AIAA-G-043 Outline	**ISO/IEC/IEEE 29148 Outline**
1. Scope	1. Scope
2. Referenced Documents	2. Referenced Documents
3. User-Oriented Operational Description	3. The Current System or Situation
4. Operational Needs	4. Justification for and Nature of Changes
5. System Overview	5. Concepts for the Proposed System
6. Operational Environment	6. Operational Scenarios
7. Support Environment	7. Summary of Impacts
8. Operational Scenarios	8. Analysis of the Proposed System

Figure 29: Alternative Concept of Operations Document Outlines

Note that this guide uses the term Concept of Operations (ConOps) where other sites like the International Council of Systems Engineering (INCOSE) uses the term Operational Concept when discussing a project level document that captures the needs the stakeholders have and how those needs will be met in the system. In the ITS industry, an Operational Concept is defined in 940.9 (c) (3) as "An operational concept that identifies the roles and responsibilities of participating agencies and stakeholders in the operation and implementation of the systems included in the regional ITS architecture;". Thus, an Operational Concept tends to be associated with a broader regional view of ITS within the ITS industry, while the Concept of Operations is a more specific project or system-level document. When you consult references outside the ITS industry, these terms may be used in precisely the opposite way.

6.4.4 Template

CONCEPT OF OPERATIONS TEMPLATE

SECTION	CONTENTS
Title Page	The title page should follow the Transportation Agency procedures or style guide. At a minimum, it should contain the following information: ▪ CONCEPT OF OPERATIONS FOR THE [insert name of project] AND [insert name of transportation agency] ▪ Contract number ▪ Date that the document was formally approved ▪ The organization responsible for preparing the document ▪ Internal document control number, if available ▪ Revision version and date issued
1.0 Purpose of Document	This section is a brief statement of the purpose of this document. It is a description and rationale of the expected operations of the system under development. It is a vehicle for stakeholder discussion and consensus to ensure that the system that is built is operationally feasible. This will briefly describe contents, intention, and audience. One or two paragraphs will suffice.
2.0 Scope of Project	This short section gives a brief overview of the system to be built. It includes its objectives and a high-level description. It describes what area will be covered and which agencies will be involved, either directly or through interfaces. One or two paragraphs will suffice.
3.0 Referenced Documents	This optional section is a place to list any supporting documentation used and other resources that are useful in understanding the operations of the system. This could include any documentation of current operations and any strategic plans that drive the goals of the system under development.
4.0 Background	Here is a brief description of the current system or situation, how it is used currently, and its drawbacks and limitations. This leads into the reasons for the proposed development and the general approach to improving the system. This is followed by a discussion of the nature of the planned changes and a justification for them.

SECTION	CONTENTS
5.0 Concept for the Proposed System	This section describes the concept exploration. It starts with a list and description of the alternative concepts examined. The evaluation and assessment of each alternative follows. This leads into the justification for the selected approach. The operational concept for that selected approach is described here. This is not a design, but a high-level, conceptual, operational description. It uses only as much detail as needed to be able to develop meaningful scenarios. In particular, if alternative approaches differ in terms of which agency does what, that will need to be resolved and described. An example would be the question of whether or not a regional signal system will have centralized control.
6.0 User-Oriented Operational Description	This section focuses on how the goals and objectives are accomplished currently. Specifically, it describes strategies, tactics, policies, and constraints. This is where the stakeholders are described. It includes who users are and what the users do. Specifically, it covers when, and in what order, operations take place, personnel capabilities, organizational structures, personnel & inter-agency interactions, and types of activities. This may also include operational process models in terms of sequence and interrelationships.
7.0 Operational Needs	Here is a description of the vision, goals & objectives, and personnel needs that drive the requirements for the system. Specifically, this describes what the system needs to do that it is not currently doing.
8.0 System Overview	This is an overview of the system to be developed. This describes its scope, the users of the system, what it interfaces with, its states and modes, the planned capabilities, its goals & objectives, and the system architecture. Note that the system architecture is not a design [that will be done later]. It provides a structure for describing the operations, in terms of where the operations will be carried out, and what the lines of communication will be.
9.0 Operational Environment	This section describes the physical operational environment in terms of facilities, equipment, computing hardware, software, personnel, operational procedures and support necessary to operate the deployed system. For example, it will describe the personnel in terms of their expected experience, skills and training, typical work hours, and other activities [e.g., driving] that must be or may be performed concurrently.
10.0 Support Environment	This describes the current and planned physical support environment. This includes facilities, utilities, equipment, computing hardware, software, personnel, operational procedures, maintenance, and disposal. This includes expected support from outside agencies.

SECTION	CONTENTS
11.0 Operational Scenarios	This is the heart of the document. Each scenario describes a sequence of events, activities carried out by the user, the system, and the environment. It specifies what triggers the sequence, who or what performs each step, when communications occur and to whom or what [e.g., a log file], and what information is being communicated. The scenarios will need to cover all normal conditions, stress conditions, failure events, maintenance, and anomalies and exceptions. There are many ways for presenting scenarios, but the important thing is that each stakeholder can clearly see what his expected role is to be.
12.0 Summary of Impacts	This is an analysis of the proposed system and the impacts on each of the stakeholders. It is presented from the viewpoint of each, so that they can readily understand and validate how the proposed system will impact their operations. Here is where any constraints on system development are documented. Metrics for assessing system performance are also included here.
13.0 Appendices	This is a place to put a glossary, notes, and backup or background material for any of the sections. For example, it might include analysis results in support of the concept exploration.

6.5 Requirements Template

6.5.1 Purpose of This Document

This document describes what the system is to do [functional requirements], how well it is to perform [performance requirements], and under what conditions [non-functional and performance requirements]. This document does not define how the system is to be built. It pulls together requirements from a number of sources including but not limited to:

- Concept of Operations and Scenarios
- Elicitation process – previous studies, "Day in the Life" studies, interviews, and workshops
- Constraints that are put onto a project, such as policies that will drive constraints on the system. [Example, the Agency policy is to use Oracle in ITS]

ITS projects have a Requirements Specification at the system and sub-system levels.

This document sets the technical scope of the system to be built. It is the basis for verifying the system and sub-systems when delivered [via the Verification Plan].

6.5.2 Tailoring this Document to Your Project

Any ITS projects will need a set of requirements defining what is needed. The tailoring is in how extensive to document these requirements. One way to gauge how many requirements to write and/or how much detail to have in the requirements document is to start at the finish line. The following should be asked when starting at the top level of the system:

- What are all the functions needed in order to satisfy for the agency that the system is doing what it is expected to do?
- How well does the system need to perform the required functions?
- Under what conditions does the system need to operate?
- What tests are needed to show the system operates as intended

Each of these tests will need a requirement. This is done for the system and the sub-systems. For simple systems there may only be 1 or 2 pages of requirements that can fully define what the system is to do. In more complex systems this could be 10 to 20 pages or more.

Other factors that drive the extent to which requirements need to be written are the amount of commercial products that are used. These products have their own specifications. So, it may be sufficient to reference them after they have been reviewed to determine if the product will meet the agency's intended need. For example, the traffic control systems that are on the market have sufficient documentation to cover the majority of functions that are required. The additional requirements would be for any modifications or enhancements needed.

6.5.3 Checklist: Critical Information

- ☑ Is there a definition of all the major system functions?
- ☑ For each function, do requirements describe: what the function does, what performance is needed, and under what conditions [e.g. environmental, reliability, and availability] the system performs the function?
- ☑ Are all terms, definitions, and acronyms defined?
- ☑ Are all supporting documents such as standards, concept of operations, and others referenced?
- ☑ Does each requirement have a link [traceability] to a higher-level requirement or a user-specified need?

- ☑ Is each requirement concise, verifiable, clear, feasible, necessary, unambiguous, and technology independent?
- ☑ Are all technology dependent requirements identified as constraints?
- ☑ Does each requirement have a method of verification defined?

6.5.4 Template

SYSTEM AND SUB-SYSTEM REQUIREMENTS TEMPLATE

IEEE Std 1233 Guide for developing System Requirements Specifications

SECTION	CONTENTS
Title Page	The title page should follow the Transportation Agency procedures or style guide. At a minimum, it should contain the following information: ▪ SYSTEM REQUIREMENTS/SUB-SYSTEM REQUIREMENTS [insert name of project] AND [insert name of transportation agency] ▪ Contract number ▪ Date that the document was formally approved ▪ The organization responsible for preparing the document ▪ Internal document control number, if available ▪ Revision version and date issued
1.0 Scope of System or Sub-system	▪ Contains a full identification of the system ▪ Provides a system overview and briefly states the objectives of the system ▪ Describes the general nature of the system ▪ Summarizes the history of system development, operation, and maintenance ▪ Identifies the project stakeholders, acquirer, users, and support agencies ▪ Identifies current and planned operating sites
2.0 Reference	Identifies all needed standards, policies, laws, concept of operations, concept exploration documents and other reference material that supports the requirements.
3.0 Requirements	Functional requirements [What the system shall do] Performance requirements [How well the requirements should perform] Interface requirements [Definition of the interfaces] Data requirements [Data elements and definitions of the system] Non-Functional requirements, such as reliability, safety, environmental [temperature] Enabling requirements [Production, development, testing, training, support, deployment, and disposal]. This can be done through references to other documents or embedded in the requirements Constraints – [e.g. Technology, design, tools, and/or standards]

SECTION	CONTENTS
4.0 Verification Methods	For each requirement, identify one of the following methods of verification: **Demonstration is** a requirement that the system can demonstrate without external test equipment. **Test** is a requirement that requires some external piece of test equipment. E.g. logic analyzer, and/or volt meter. **Analyze** is a requirement that is met indirectly through a logical conclusion or mathematical analysis of a result. E.g. Algorithms for congestion: the designer may need to show that the requirement is met through the analysis of count and occupancy calculations in software or firmware. **Inspection** is verification through a visual comparison. For example, quality of welding may be done through a visual comparison against an in-house standard.
5.0 Supporting Documentation	Catch-all for anything that may add to the understanding of the Requirements without going elsewhere [Reference section] Examples: diagrams, analysis, key notes, memos, rationale, stakeholders contact list
6.0 Traceability Matrix	This is a table that traces the requirements in this document to the higher-level requirements or if this is a top-level requirements document, it should trace to the User Needs (which are defined in the Concept of Operations)
7.0 Glossary	Terms, acronyms, definitions

6.6 Design Specification Template

6.6.1 Purpose of these Documents

These documents describe how the system is to be built. They take the requirements [what the system will do] and translate them into a hardware and software design that can be built. Collectively, the purpose of these documents is to:

- Provide a documented description of the design of the system that can be reviewed and approved by the stakeholders
- Provide a description of the system in enough detail that its component parts can be procured and built
- Provide a description of the hardware and software system components in sufficient detail for them to be maintained and upgraded
- For most projects, two levels of design specification are developed. The High-Level Design Specification Document supports the project architecture, interfaces, and sub-system requirements. The Detailed Design Specification Documents provide the build-to specification for software and hardware construction

For some systems, it is advisable to create separate documents, called Interface Design Documents, to describe the internal and external interfaces of the system being built.

6.6.2 Tailoring these Documents to Your Project

Any ITS projects that are structured to produce a physical hardware / software system require some level of design description of the system to be procured or built. Study projects with only paper products don't need them. For simple systems or for systems that use products already deployed by the agency to address similar requirements, only one minimal document is sufficient [perhaps just a list of the items to be procured].

If a project involves the fabrication of hardware components, the information contained in the design specifications are supplemented with drawings from which the parts are built. Construction and installation drawings may also be required.

If a project involves the development of custom software, even relatively simple software, then both documents are strongly recommended.

A software design is documented by these specifications and by the source code itself. It is vital that the Detailed Design Specification exists along with the source code. Further, the specification must track to this code.

Interfaces that are not shared with others may be completely contained in the Detailed Design Specifications otherwise they are specified in the Interface Specification. Some modern programming techniques make processor-to-processor interfaces completely transparent to the code. However, some interface methods, especially interfaces to existing external systems, are very specialized and unique. In these cases, a separate document that can be easily reviewed by engineers on both sides of the interface is very useful.

6.6.3 Checklist: Critical Information

- ☑ Does the High Level Design Document include definition of requirements unique to the chosen architecture [interfaces between sub-systems, for instance]?
- ☑ Is the definition of each requirement from the Requirements Document complete enough for implementation? Or, does it need to be expanded in the High Level Design Document?
- ☑ Are system requirements traced to the sub-systems in the High Level Design Document?

- ☑ Are commercial off-the-shelf products identified in the High Level or Detailed Design Document?
- ☑ Is the design approach for common software methods defined, as appropriate, in both the High Level and Detailed Design Documents?
- ☑ Is the architecture, both hardware and software, of the sub-systems [components and interconnections] defined in the high level design specification?
- ☑ Are any necessary database schema and structures defined in the High Level and Detailed Design Documents?
- ☑ Are the hardware components defined in enough detail in the design documents to support procurement or fabrication?
- ☑ Has the trace from requirements to hardware and software components been checked and verified?
- ☑ Is the Detailed Design Document linked to the source code components, that is, do they use the same object names, file names, attribute names, and method names?

6.6.4 Templates

HIGH LEVEL DESIGN SPECIFICATION TEMPLATE

IEEE Std 1233 Guide for developing System Requirements
IEEE 1471-2000 Recommended Practice for Architectural Description of Software Intensive Systems

SECTION	CONTENTS
Title Page	The title page should follow the Transportation Agency procedures or style guide. At a minimum, it should contain the following information: ▪ HIGH LEVEL DESIGN SPECIFICATION FOR THE [insert name of project] AND [insert name of transportation agency] ▪ Contract number ▪ Date that the document was formally approved ▪ The organization responsible for preparing the document ▪ Internal document control number, if available ▪ Revision version and date issued
1.0 Purpose of Document	This section is a brief statement of the purpose of this document. It is a high level description of the architecture [hardware and software] of the system. It summarizes the contents of the document. Sometimes the High Level Design specification is used to document some requirements not covered elsewhere, such as an operator interface or interfaces to external systems. It also may be necessary to include functional requirements arising from the internal interfaces created between the sub-systems.
2.0 Scope of Project	This section gives a brief description of the planned project and the objectives of the system to be built. This section can be copied from a previous document, and is included for completeness. This may be the only document which some project participants and stakeholders may see.

SECTION	CONTENTS
3.0 Sub-systems	This section describes the architecture of the system and how it is divided into sub-systems, when that is found to be necessary. Simpler systems may not need to be subdivided, and if so, this section is void. When sub-systems are needed, each is described in terms of its purpose, its functionality, its interfaces with other sub-systems, and its component parts [hardware and software]. If the requirements call for different capabilities at multiple sites, then the allocation of the sub-systems to these sites is shown. In order to describe the functionality of a sub-system, it is necessary to allocate system requirements to each sub-system. All requirements must be covered by at least one sub-system. However, some requirements [and especially performance requirements] may be applicable to several sub-systems. An explicit trace of all requirements from the Requirements Document into the sub-systems is a part of this document. In addition to the system requirements, additional requirements may be necessary to show how the sub-systems work together. Those types of requirements are analyzed and documented here.
4.0 Hardware Components	This section identifies the hardware components of each sub-system. It identifies them by name, function, capabilities, source [manufacturer], and quantity. It shows the interconnections between the components [e.g. point-to-point, or local area network]. If a hardware component needs optional components or features, they are listed and defined at this time. This section also includes a trace of requirements, where applicable, into the hardware components.

SECTION	CONTENTS
5.0 Software Components	This section describes the preliminary design of the software application. It shows the allocation of the software to sub-systems and to hardware elements. It shows and identifies the software packages to be used; and their allocation to sub-systems and to hardware components. It also shows/identifies all custom designed software packages and their allocation to sub-systems and hardware components. It shows the architectural relationship between the various software packages, both custom and SW that is already used by the agency. The high level design of each custom software package is described. The method used for this description depends on the methodology being used for software design. That methodology may be object-oriented design, data flow design, structured design, or any other method chosen by the project and the software development team. For example, if an object oriented software design methodology is to be used, the description of the custom software components for the High Level Design specification would include: Preliminary class description for significant internal and external classes necessary to implement the functional requirements ▪ Preliminary description of the attributes, methods, and relationships of each class of objects ▪ Class diagrams and other diagramming methods as appropriate, such as, sequence, package, activity concurrency, and state diagrams ▪ Component diagrams to describe the physical partitioning of the software into code components ▪ Descriptions of common patterns to be used in the software design, such as, the pattern to be used for inter-process communication, or for implementation of an operator interface ▪ Trace requirements into each software package

SECTION	CONTENTS
6.0 Sub-system Requirements	This document may be used to describe additional requirements that were not covered in the requirements specifications. These may include, but are not limited to: ▪ Showing greater detail of previously defined functional requirements based on additional functional analysis; for instance, defining the details of a complex algorithm ▪ Providing complete details of complex requirements, such as a detailed description of a complex operator interface where considerable work with operations personnel is necessary before a definitive statement of the requirement can be made ▪ Providing complete details of an interface with an external system ▪ Stating requirements which result from the separation of the system into sub-systems. That is, identifying functional requirements for the way these sub-systems work together Of course, these types of requirements [with the exception of the last type] also may be included in the Requirements Document or documented in separate documents, as deemed appropriate.
7.0 Applicable Documents	This section lists the applicable documents that constrain the design process. Such documents may include standards and external system specifications.

DETAILED DESIGN SPECIFICATION TEMPLATE

SECTION	CONTENTS
Title Page	The title page should follow the Transportation Agency procedures or style guide. At a minimum, it should contain the following information: ▪ DETAILED DESIGN SPECIFICATION FOR THE [insert name of project] AND [insert name of transportation agency] ▪ Contract number ▪ Date that the document was formally approved ▪ The organization responsible for preparing the document ▪ Internal document control number, if available ▪ Revision version and date issued
1.0 Purpose of Document	This section is a brief statement of the purpose of this document. The purpose is, to expand and complete the preliminary design descriptions included in the High Level Design Document.
2.0 Scope of Project	This section describes the project and may be lifted from the High Level Design Document.
3.0 Sub-systems	This section completes the description of the system architecture and the sub-systems, as necessary.

SECTION	CONTENTS
4.0 Hardware Components	This section completes the description of the hardware components. It contains a detailed list of the exact hardware items to be procured by name, part number, manufacturer, and quantity. If necessary, it lists any hardware component specifications or drawings which have been prepared by the design team.
5.0 Software Components	This section completes the description of the software components. It contains a detailed list of the software products to be procured, by vendor, name, part number, and options. If the project involves custom software applications, this section becomes the dominant and largest part of the Detailed Design Document. Its purpose is to provide enough information so the code can be developed. Subsequently, so the code can be understood for maintenance and system upgrades. As a result, the overriding requirement is that the descriptions of the software components are complete and the link between these descriptions and the actual source code is clear and explicit. The Detailed Design Specification is primarily a completion of the preliminary information in the High Level Design Specification. Any corrections to the information in the previous document should be made at this time. Again, if a software design tool is used, it may produce most of the Detailed Design Specification. For example, if an object oriented software design methodology is to be used, the description of the custom software components for the Detailed Design Specification would include expansion of the following from the High Level Design Specification: ▪ Class description for significant internal and external classes necessary to implement the functional requirements ▪ Description of each class attributes, methods, and relationships ▪ Class diagrams and other diagramming methods as appropriate, such as: sequence, package, activity concurrency, and state diagrams ▪ Component diagrams to describe the physical partitioning of the software into code components ▪ Descriptions of common patterns to be used in the software design, such as, the pattern to be used for inter-process communication, or for implementation of an operator interface

INTERFACE DESIGN DOCUMENT TEMPLATE

SECTION	CONTENTS
Title Page	The title page should follow the Transportation Agency procedures or style guide. At a minimum, it should contain the following information: ▪ INTERFACE DESIGN DOCUMENT FOR THE [insert name of interface] FOR THE [insert name of transportation agency] ▪ Contract number ▪ Date that the document was formally approved ▪ The organization responsible for preparing the document ▪ Internal document control number, if available ▪ Revision version and date issued
1.0 Purpose of Document	This section is a brief statement of the purpose of this document. It defines the function and design of an interface between two parts of the system or between the system and an external system.
2.0 Scope of Project	This section describes the project and may be lifted from the High Level Design Document.
3.0 Interface Purpose and General Description	This section is used to describe, in operational terms, the purpose of this interface. It shows how that purpose relates to the overall operation of the system being designed. It describes the information flow, in both directions if that is applicable, and the actions or conditions that cause information to be transferred across the interface. It describes where that information comes from and where it is used.
4.0 Communications Method	This section describes the communications protocols associated with information flow across the interface. Especially, protocols that the programmer has to use in order to make the transfer occur. This form and content of this section, and the next, are very dependent on the type of communication method used. For instance, the description of a database replication method is different from a File Transfer Protocol [FTP] method or from a remote procedure call method. There are many other communications methods that can be used. For internal interfaces, selection of a process-to-process communications method is part of the software design effort. However, when communicating with an external system, the usual case is that system already exists, and has a defined communication protocol. In this case, the software designer must build a compatible interface. That work is facilitated by this document.

SECTION	CONTENTS
5.0 Specific Interface Design	Along with the previous section, the form and content of this is completely dependent on the method used to transfer information, or data, from process to process and from system to system. This section focuses on the form and content of the data elements themselves instead of the communications protocols described before. For instance, if database replication is used, this section describes the logical data structure and the specific database information contained in the fields of the database. If a message method is used, this section describes the content of each field of the message and its allowable values. If a remote procedure call type of interface is implemented, this sections describes the function of the call, the parameters passed with it, the parameters returned by the call, and the actions taken by the remote procedure. These are just three examples of a variety of methods that may be used. This section must contain enough information to allow the software developer to design and write code to implement the interface.

6.7 Integration Plan Template

6.7.1 Purpose of this Document

A project's integration and verification strategy is closely tied to the design of the system and its decomposition into sub-systems. The factors that are considered when developing the sub-system design are covered elsewhere in this document. Whatever the goals were [and they vary from project to project], the Integration Plan needs to be structured to bring the components together to create each sub-system and to bring the various sub-systems together to make the whole system. Further, this needs to be done in a way that supports the deployment strategy. That is the first purpose of an Integration Plan.

The second purpose is to describe to the participants in each integration step what has to be done. The integration team has to assemble various resources for each integration step. The Integration Plan identifies the needed resources. In addition, it identifies when and where the resources will be needed.

6.7.2 Tailoring this Document to Your Project

An Integration Plan, at least as a separate written document, is not always needed. The complexity of the system, the complexity of the eventual deployment of the system, and the complexity of the development effort influence the decision to prepare an Integration Plan. For instance, a deployment strategy that calls for multiple installations at multiple locations can require a complex sequence of integration activities. Another common complexity of integration arises when different teams are developing the sub-systems. This is especially true when the different development teams are comprised of different contractors, each with their own contract. In this case, they need to know more about their required work to support integration than would be the case if the same development team were working both sides of the integration effort. The same type of complexity comes into play when an integration step involves external systems owned by other agencies, or at least other organizations within the agency.

If a separate Integration Plan is not warranted, the necessary planning information can be included in: the Project Plan, the SEMP, the Verification Plan and the software development plans of the development team.

6.7.3 Checklist: Critical Information

- ☑ Does the Integration Plan include and cover integration of all of the components and sub-systems, either developed or purchased, of the project?
- ☑ Does the Integration Plan account for all external systems to be integrated with the system [for example, communications networks, field equipment, other complete systems owned by the agency or owned by other agencies]?
- ☑ Does the Integration Plan fully support the deployment strategy. For example, *when* and *where* the sub-systems and system is to be deployed?
- ☑ Are the integration steps defined in the Integration Plan consistent with the verification activities defined in the Verification Plan?
- ☑ For each integration step, does the Integration Plan define what components and sub-systems are to be integrated?
- ☑ For each integration step, does the Integration Plan identify all the needed participants and define what their roles and responsibilities are?
- ☑ Does the Integration Plan establish the sequence and schedule for every integration step?
- ☑ Does the Integration Plan spell out how integration problems are to be documented and resolved?

6.7.4 Template

INTEGRATION PLAN TEMPLATE

SECTION	CONTENTS
Title Page	The title page should follow the Transportation Agency procedures or style guide. At a minimum, it should contain the following information: ▪ INTEGRATION PLAN FOR THE [insert name of project] AND [insert name of transportation agency] ▪ Contract number ▪ Date that the document was formally approved ▪ The organization responsible for preparing the document ▪ Internal document control number, if available ▪ Revision version and date issued
1.0 Purpose of Document	A brief statement of the purpose of this document. It is, the plan for integrating the components and sub-systems of the project prior to verification.
2.0 Scope of Project	This section gives a brief description of the planned project and the objectives of the system to be built. Special emphasis is placed on the project's deployment complexities and challenges. This section may be lifted from earlier documents. It is important only to people [stakeholders] who will be introduced to the project for the first time by this document.
3.0 Integration Strategy	This section informs the reader what the high level plan is for integration and, most importantly, why the integration plan is structured the way it is. As mentioned before, the Integration Plan is subject to several constraints, sometimes conflicting constraints. Also, it is one part of the larger process of build, integrate, verify, and deploy. All of which must be synchronized to support the same project strategy. So, for even a moderately complex project, the integration strategy, based on a clear and concise statement of the project's goals and objectives, is described here at a high, but all-inclusive, level. It may also be necessary to describe the analysis of alternative strategies to make it clear why this particular strategy was selected. The same strategy is the basis for the Build Plan, the Verification Plan, and the Deployment Plan. So, it may only be necessary to justify this strategy once, perhaps in the Project Plan, or in the SEMP. This section covers and describes each step in the integration process. It describes what components are integrated at each step and gives a general idea of what threads of the operational capabilities [requirements] are covered. It ties the plan to the previously identified goals and objectives so the stakeholders can understand the rationale for each integration step. This summary level description also defines the schedule for all the integration efforts.

<table>
<tr><th>SECTION</th><th>CONTENTS</th></tr>
<tr><td>4.0 Phase 1 Integration</td><td>This, and the following sections, define and explain each step in the integration process. The intent here is to identify all the needed participants and to describe to them what they have to do.

In general, the description of each integration step should identify:
▪ The location of the activities
▪ The project-developed equipment and software products to be integrated Initially this is just a high level list but eventually the list must be exact and complete, showing part numbers and quantity
▪ Any support equipment [special software, test hardware, software stubs, and drivers to simulate yet-to-be-integrated software components, external systems] needed for this integration step. The same support equipment is most likely needed for the subsequent verification step
▪ All integration activities that need to be performed after installation, including integration with on-site systems and external systems at other sites
▪ A description of the verification activities [as defined in the applicable Verification Plan] that occur after this integration step
▪ The responsible parties for each activity in the integration step
▪ The schedule for each activity</td></tr>
<tr><td>5.0 Multiple Phase Integration steps [1 or N steps]</td><td>This, and any needed additional sections, follow the format for section 3. Each covers each step in a multiple step integration effort.</td></tr>
</table>

6.8 Verification Documents Template

6.8.1 Purpose of these Documents

These documents plan, describe, and record the activity of verifying that the system being built meets the specified requirements. Since a complex system may involve a series of verification activities, several sets of these verification documents may be needed. All of these verification documents follow the master plan for verification defined in the Systems Engineering Management Plan.

Usually, for even moderately complex systems, the following three levels of verification documents are prepared:

- a plan to initially lay out the specific verification effort
- a procedure that is the specific and detailed steps to be followed to perform the test
- a report on the results of the testing activity

These three documents are described in this section.

A critical issue is assuring that all requirements are verified by the testing activity. This is best done by first tracing each requirement into a test case then, into a step in the Verification Procedure.

Additional Information is found in IEEE 1012-1998, Software Verification and Validation.

6.8.2 Tailoring these Documents to Your Project

A separate Verification Plan and procedure may not be required for the simplest projects, especially where the system is essentially using products already deployed by the agency to address similar requirements and does not involve any custom software development, and where the project office personnel have a very clear understanding of the purpose of the system. In some cases, it is possible to take a copy of the Requirements Document, improvise procedures, and annotate the Requirements Document with the results of each test step. This can be a perfectly acceptable way to verify the operations of a system.

However, preparation of these verification documents is strongly advised if:

- the system is more complex
- there are a number of separate verification activities
- multiple deployment sites are involved
- more than one or two stakeholders have to be satisfied

There is also the question of how comprehensive to make the verification effort. It is impossible to test everything, that is, all possible combinations of actions under all possible operational situations. A good rule of thumb is: if it was important enough to write down as a requirement, then it should be tested, at least once, as part of a reasonable operational scenario. This may not, for example, test all possible failure mode conditions. If a good job was done in writing the requirements, then the most important and most likely are verified.

6.8.3 Checklist: Critical Information

- ☑ Is there a documented Verification Plan for the Project?
- ☑ Does the Verification Plan answer all the questions of *who*, *what*, *where*, and *when* concerning test conduct?
- ☑ Does the Verification Plan make clear what needs to happen if a test failure is encountered?
- ☑ Does the Verification Plan define the configuration of the hardware, software, and external system needed for each test case?

- ☑ Are all applicable requirements traced to a test case in the Verification Plan? Does each test case define a realistic and doable test?
- ☑ Are detailed verification procedures documented for the project?
- ☑ Is each step in the Verification Procedure traced to a test case and a requirement?
- ☑ Are all of the necessary initial conditions and set-up defined for each procedure?
- ☑ Has each verification procedure been dry run prior to the formal test? Have the procedures been updated as a result?
- ☑ *Is there a Verification Report that documents the project verification results?*
- ☑ Does the Verification Report describe, in detail, the resolution of every test anomaly encountered during testing?

6.8.4 Templates

VERIFICATION PLAN TEMPLATE

IEEE 1012-1998 Independent Verification and Validation

SECTION	CONTENTS
Title Page	The title page should follow the Transportation Agency procedures or style guide. At a minimum, it should contain the following information: ▪ VERIFICATION PLAN FOR THE [insert name of project] AND [insert name of transportation agency] ▪ Contract number ▪ Date that the document was formally approved ▪ The organization responsible for preparing the document ▪ Internal document control number, if available ▪ Revision version and date issued
1.0 Purpose of Document	This section identifies the type of verification activity to be performed within this Verification Plan. For instance, this activity may verify the entire system, a sub-system, the deployment at a site, a burn-in test, or any other verification activity called for in the Program Plan or in the SEMP.
2.0 Scope of Project	This section gives a brief description of the planned project and the objectives of the system to be built. Special emphasis is placed on the project's complexities and challenges that must be addressed and verified by the systems engineering efforts. This section also describes the environment in which the project operates. It identifies the organization structures that encompass all stakeholders. It also gives a brief description of the role to be played by each stakeholder. This includes ad hoc and existing management work groups and multi-disciplinary technical teams that should be formed for supporting the project. Such teams are critical to reaching successful system deployment.

<table>
<tr><th>SECTION</th><th>CONTENTS</th></tr>
<tr><td>3.0 Referenced Documents</td><td>This is a list of all documents used in the preparation of this Verification Plan. This almost always includes the Project Plan, the SEMP [if one was written], and the applicable Requirements Documents. However, reference of other documents, such as descriptions of external systems, standards, a Concept of Operations, and manuals may need to be included.</td></tr>
<tr><td>4.0 Test Conduct</td><td>This section provides details on how the testing is accomplished. It defines: who does the testing; when and where it is to be done; the responsibilities of each participant before, during, and after each test; the hardware and software to be used [and other systems as well]; and the documents to be prepared as a record of the testing activity. Another very important part of this section defines how testing anomalies are to be handled [that is, what to do when a test fails].

In general, the following information should be included in this section:
A description of the participating organizations and personnel and identification of their roles and responsibilities. This may include for example, a test conductor, test recorder, operators, and/or engineering support.
Identification of the location of the testing effort, that is, the place, or places, where the testing progress must be observed.
The hardware and software configuration for all of the test cases, including hardware and software under test and any supporting test equipment, software, or external systems. Several configurations may be necessary.
Identification of the documents to be prepared to support the testing, including Verification Procedures, a Verification Report and descriptions of special test equipment and software.
Details on the actual conduct of the testing, including:
Notification of participants
Emphasis on the management role of the test conductor
Procedures for approving last minute changes to the procedures
The processes for handling a test failure, including recording of critical information, determination of whether to stop the testing, restart, or skip a procedure, resolution of the cause of a failure [e.g. fix the software, reset the system, and/or change the requirements], and determination of the retesting activities necessary as a result of the failure.</td></tr>
</table>

SECTION	CONTENTS
5.0 Test Identification	This section is the heart, and largest, section of the Verification Plan. Here we identify the specific test cases to be performed. A test case is a logical grouping of functions and performance criteria [all from the Requirements Documents] that is to be tested together. For instance, a specific test case may cover all the control capabilities to be provided for control of a changeable message sign. There may be several individual requirements that define this capability, and they all are verified in one test case. The actual grouping of requirements into a test case is arbitrary. They should be related and easily combined into a reasonable set of test procedure actions. Each test case should contain at least the following information: ▪ A description name and a reference number ▪ A complete list of the requirements to be verified. For ease of tracing of requirements into the Verification Plan and other documents, the requirements are given numbers. They can be accurately and conveniently referenced without repeating all the words of the requirement ▪ A description of the objective of the test case, usually taken from the wording of the requirements, to aid the reader understanding the scope of the test case ▪ Any data to be recorded or noted during the test, such as expected results of a test step. Other data, such as a recording of a digital message sent to an external system, may be required to verify the performance of the system. ▪ A statement of the pass/fail criteria. Often, this is just a statement that the system operates per the requirements ▪ A description of the test configuration. That is a list of the hardware and software items needed for the test and how they should be connected. Often, the same configuration is used for several tests ▪ A list of any other important assumptions and constraints necessary for conduct of the test case

VERIFICATION PROCEDURE TEMPLATE

SECTION	CONTENTS
Title Page	The title page should follow the Transportation Agency procedures or style guide. At a minimum, it should contain the following information: ▪ VERIFICATION PROCEDURE FOR THE [insert name of project] AND [insert name of transportation agency] ▪ Contract number ▪ Date that the document was formally approved ▪ The organization responsible for preparing the document ▪ Internal document control number, if available ▪ Revision version and date issued

SECTION	CONTENTS
1.0 Purpose of Document	This section identifies the type of verification to be performed. For instance, this activity may verify the entire system, a sub-system, the deployment at a site, a burn-in test, or any other verification activity called for in the Program Plan or in the SEMP.
2.0 Verification Configuration and Software Under Test	This section identifies the equipment and software to be verified. It also identifies all equipment and software necessary for this verification activity that is external to the system / sub-system configuration under test. This may include special test equipment and any external systems with an interface to the configuration under test. For the hardware / software configuration under test, this section identifies: ▪ Each hardware item by part number and serial number ▪ Each item of software, by part number and version number ▪ Each source code file of custom developed software, by file name and version number ▪ For all special test equipment / software, this section identifies: – Each hardware item by part, serial, and version number – Each item of software, by part number and version number – Each source code file of custom developed software by file name and version number For each external system interface, this section identifies: ▪ The name and location of the external system
3.0 Verification Setup	This section describes the steps to be taken to set up each verification configuration, including, but not limited to, tuning of the hardware, configuring and starting the software, starting the special test software, and set-up steps at each external system to be used.
4.0 Verification Procedures	This section describes the step-by-step actions to be taken by the verification operator for each verification case. Each step includes: ▪ Operator action to be taken. This operator action may be, for example, an entry at a workstation, initiation of a routine in the special test software, or an action at an external system. ▪ Expected result to be observed. This too may take several forms, for example, display of certain information at a workstation, a response at an external system, recording of data for subsequent analysis, or an action by a field device. ▪ Pass / fail entry space. Here the verification conductor records whether or not the expected result occurred. If the expected results are not observed, then the procedures for dealing with failures contained in the Verification Plan are invoked. ▪ A trace of each verification step from a verification case in the applicable Verification Plan and a trace from a requirement in the applicable Requirements Document.

VERIFICATION REPORT TEMPLATE

SECTION	CONTENTS
Title Page	The title page should follow the Transportation Agency procedures or style guide. At a minimum, it should contain the following information: ▪ VERIFICATION REPORT FOR THE [insert name of project] AND [insert name of transportation agency] ▪ Contract number ▪ Date that the document was formally approved ▪ The organization responsible for preparing the document ▪ Internal document control number, if available ▪ Revision version and date issued
1.0 Purpose of Document	This section identifies the type of verification performed. For instance, the activity may verify the entire system, a sub-system, the deployment at a site, a burn-in test, or any other verification activity called for in the Program Plan or in the SEMP. This section can be taken from the applicable Verification Procedure.
2.0 Identification of the Configuration under test	This section identifies the equipment and software verified. It also identifies all equipment and software necessary for this verification activity that is external to the system / sub-system configuration under test. This may include special test equipment and any external systems with an interface to the configuration under test. This section can be taken from the applicable Verification Procedure.
3.0 Individual Test Case Report	This section summarizes the purpose and results of each test case performed in the applicable Verification Procedure. Special attention is paid to any test case where a failure occurred and how the failure was resolved. This section covers: ▪ Test case overview and results ▪ Completed Verification Procedure pages annotated with pass / fail results ▪ Description of each failure, if any, from the expected result called for in the Verification Procedure ▪ Any back-up data or records related to the field procedure ▪ Details of the resolution of each test failure, including procedure modification, software fix, re-testing and results, regression testing and results, and required document changes [including changes to the requirements].

6.9 Deployment Plan Template

6.9.1 Purpose of this Document

Deployment is the final step in the development of a system. A Deployment Plan is developed based on a thorough analysis of the steps necessary to achieve the deployment goals of the project. It both serves to justify the strategy for deployment and to inform all deployment participants [and other stakeholders] of what will happen and what they will be required to do.

These two parts of the plan serve different purposes and should be written at different times. The strategy section shows management [and the operations people who will get the system] what the selected strategy is and how it best meets the constraints placed on the project [for instance, a multi-year funding profile and viable operational capabilities at each step].

The plan section is just that, a detailed plan for each deployment step, answering what, when, where, how, and by whom. This part is best written when the design is fairly complete and the exact system components, as well as their characteristics, are known in great detail.

6.9.2 Tailoring this Document to Your Project

There are a number reasons to have a Deployment Plan. Sometimes the deployment of a system is very simple and may not need a very extensive plan. For example, if all deployment takes place at one location and at one time. On the other hand, if there are multiple locations, multiple deployments at each location, many external interfaces [other systems], or there are multiple agencies involved a Deployment Plan can be very helpful.

It is also possible that only one of the two parts of the Deployment Plan [as mentioned above] is needed. Specifically, the time spent in preparing the strategy section very much depends on how much “selling” of the plan is needed.

Project management may also decide that the subject of deployment is covered well enough in other documents [especially the Project Plan, the SEMP and the Verification Plan, as well as installation and construction drawings] that a separate Deployment Plan Document is not necessary. There are many factors to be considered, but the most important is, can the deployment be successful without the expense of developing a Deployment Plan?

6.9.3 Checklist: Critical Information

- ☑ Are all the important, and significant, deployment goals and objectives captured?
- ☑ Have as many as possible of the viable deployment strategies been analyzed and compared?
- ☑ Are the strengths of the recommended deployment strategy fully explained?
- ☑ Does the recommended deployment strategy include a clear description of the operational capabilities that exist after each deployment step?
- ☑ Has the recommended deployment strategy been presented to the appropriate stakeholder decision makers?
- ☑ Has the recommended deployment strategy been accepted by the stakeholder decision makers?
- ☑ Are all of the deployment phases included in the Deployment Plan?
- ☑ Are all of the prerequisites to starting each deployment step included and is the responsible party for each identified?
- ☑ Are the installation plans needed for each deployment step identified?
- ☑ Is the list of hardware and software products needed for each deployment step identified?
- ☑ For each deployment, are all participants identified?

6.9.4 Template

DEPLOYMENT PLAN TEMPLATE

SECTION	CONTENTS
Title Page	The title page should follow the Transportation Agency procedures or style guide. At a minimum, it should contain the following information: ▪ DEPLOYMENT PLAN FOR THE [insert name of project] AND [insert name of transportation agency] ▪ Contract number ▪ Date that the document was formally approved ▪ The organization responsible for preparing the document ▪ Internal document control number, if available ▪ Revision version and date issued
1.0 Purpose of Document	A brief statement of the purpose of this document. It is the plan for deploying the systems of the project over one or more phases and at one or more physical locations [sites].
2.0 Scope of Project	This section gives a brief description of the planned project and the objectives of the system to be built. Special emphasis is placed on the project's deployment complexities and challenges. This section may be lifted from earlier documents. It is important only to people [stakeholders] who will be introduced to the project for the first time by this document.

SECTION	CONTENTS
3.0 Deployment Strategy	A complex deployment, involving multiple deployment steps at multiple sites, is based on certain goals and objectives. This section lists those goals and objectives and is used to “sell” the Deployment Plan to the stakeholders. It is also important that the deployment participants understand why the deployment is proceeding as it is so they can work with and support the plan. The significant goals and objectives guiding the deployment strategy should be relatively few [no more than a dozen] and need to be clearly stated in this section. Some typical examples of goals and objectives include: ▪ The funding profile for a multi-year project which limits the scope of deployment in a single year ▪ Development and installation prerequisites. An analysis of the system may show that feature A must be deployed first before features B, C or D, all of which need A to function ▪ Construction activities that must precede deployment ▪ Deployment of interfacing systems [especially by other agencies] that must precede deployment of a system feature ▪ The need to create a viable operational capability at each stage of the deployment. This influences how much of the system must be deployed at each step Following the statement of the goals and objectives, a high level view of the deployment strategy is presented. This covers and describes each phase of deployment at each of the sites involved. It describes: what is deployed, where it is deployed, and what operational capabilities are the results of this phase of the deployment. It ties the plan to the previously identified goals and objectives so the stakeholders can understand the rationale for each phase. This summary should include an estimate of the cost of each phase to show the plan satisfies the funding profile. It should also show the overall deployment schedule.

SECTION	CONTENTS
4.0 Phase 1 Deployment	This, and the following sections, define and explain each phase of the deployment. The intent here is to identify all the needed participants and to describe to them what they have to do. As will be seen in the following list of section contents, not only are the deliverable products identified, but so is any site work that must be done prior to installation, as well as all activities necessary to show that the deployment was successful and the system is ready for operations, or whatever comes next. In general, each phase description should identify: ▪ The location of the deployment activities ▪ The project-developed equipment and software products to be deployed. Initially this is just a high level list but eventually the list must be exact and complete, showing part numbers and quantity. If detailed hardware installation drawings have been prepared, they are referenced here ▪ All site work [including construction and facilities] that is needed before installation can begin. Again, reference to drawings may be required. Also, any necessary inspection and testing of this work is defined ▪ All integration activities which need to be performed after installation, including integration with on-site systems and with external systems at other sites ▪ All verification activities [as defined in the applicable Verification Plan] that must occur prior to acceptance of the site ▪ All supporting activities that must be completed before site acceptance, such as training and manuals ▪ The responsible parties for each activity ▪ The schedule for each activity
5.0 Multiple Phase Deployment steps [1 or N steps]	This, and any needed additional sections, follows the format for section 3. Each covers each step in a multiple step deployment effort.

6.10 Validation Documents Template

6.10.1 Purpose of these Documents

These documents plan, describe, and record the activity of validating that the system meets the intended purpose and needs of the systems' owner and stakeholders. Since a complex system may involve a series of validation related activities, several sets of these documents may be needed. All of these validation documents follow the master plan for validation defined in the Systems Engineering Management Plan.

Usually, for even moderately complex systems, the following two levels of validation documents are prepared:

- a plan to lay out the specific validation efforts

- a report on the results of the validation activity

These documents are described in this section. Note that for this phase there is no set of detailed procedures. Validation as described in Section 3.3.11 involves stakeholders and the actual users of the system. The Validation Plan will lay out the overall expectations for the assessment of the completed system but will let the users work the system as part of their every day job rather than imposing a strict step-by-step procedure. It is their system after all and they need to be comfortable with it and provide the final answer on how well it satisfies their needs. Reporting the results and any corrective actions needed will be part of this effort.

A critical issue is assuring that all user needs are included in the validation activity. This is best done by first tracing each of the needs documented in the Concept of Operations into a validation activity.

Additional Information is found in IEEE 1012-1998, Software Verification and Validation.

6.10.2 Tailoring these Documents to Your Project

A separate Validation Plan may not be required for the simplest projects, especially where the system is currently deployed by the agency to address similar requirements and does not involve any custom software development, and where the project office personnel have a very clear understanding of the purpose of the system. In some cases, it is possible to take a copy of the ConOps Document, improvise an outline or scenario to check-off the validated system, and annotate the ConOps with the results of the assessment. This can be a perfectly acceptable way to validate the performance of a simple system.

However, preparation of these validation documents is strongly advised if:

- the system is more complex
- there are a number of separate validation activities
- multiple deployment sites are involved
- more than one or two stakeholders have to be satisfied

There is also the question of how comprehensive to make the validation effort. It is impossible to cover everything, that is, all possible combinations of actions under all possible operational situations. A good rule of thumb is: if it was important enough to write down as a need, then it should be validated, at least once, as part of using the system in a real-world environment. This may not, for example, test all possible failure mode conditions. If a good job was done in earlier phases of testing and verification then the most important and most likely scenarios are covered.

6.10.3 Checklist: Critical Information

- ☑ Is there a documented Validation Plan for the system of interest?
- ☑ Does the Validation Plan answer all the questions of who, what, where, and when concerning validation?
- ☑ Does the Validation Plan make clear what needs to happen if a problem is encountered?
- ☑ Does the Validation Plan define the environmental conditions and systems configuration needed for each scenario?
- ☑ Are all applicable user needs traced to an event in the Validation Plan? Does each validation event define a realistic and doable scenario?
- ☑ Was a Validation Report developed that documents the validation results?
- ☑ Does the Validation Report describe, in detail, the resolution of every anomaly encountered during testing?
- ☑ Does the Validation Report include recommendations from the users and stakeholders to address the situation in future system evolutions?

6.10.4 Template

VALIDATION PLAN TEMPLATE

IEEE 1012-1998 Independent Verification and Validation

SECTION	CONTENTS
Title Page	The title page should follow the Transportation Agency procedures or style guide. At a minimum, it should contain the following information: ▪ VALIDATION PLAN FOR THE [insert name of project] AND [insert name of transportation agency] ▪ Contract number ▪ Date that the document was formally approved ▪ The organization responsible for preparing the document ▪ Internal document control number, if available ▪ Revision version and date issued
1.0 Purpose of Document	This section identifies the type of validation activity to be performed within this Plan. For instance, this activity may validate the entire system, a sub-system, the deployment at a site, or any other validation activity called for in the Program Plan or in the SEMP.
2.0 Scope of Project	This section gives a brief description of the planned project and the purpose of the system to be built. Special emphasis is placed on the project's complexities and challenges that must be addressed by the systems engineering efforts. This section also describes the environment in which the project operates. It identifies the organization structures that encompass all stakeholders. It also gives a brief description of the role to be played by each stakeholder. This includes ad hoc and existing management work groups and multi-disciplinary technical teams that should be formed for supporting the project. Such teams are critical to reaching successful system deployment.
3.0 Referenced Documents	This is a list of all documents used in the preparation of this Validation Plan. This almost always includes the Project Plan, the SEMP [if one was written], and the Concept of Operations. However, reference of other documents, such as descriptions of external systems, standards, and manuals may need to be included.

SECTION	CONTENTS
4.0 Validation Conduct	This section provides details on how the validation is accomplished. It defines: who does it; when and where it is to be done; the responsibilities of each participant before, during, and after each event/activity; the hardware and software to be used [and other systems as well]; and the documents to be prepared as a record of the activity. Another very important part of this section defines how anomalies are to be handled [that is, what to do when something fails or, in the case of Validation, does not match the documented needs or does not satisfactorily address the original problem]. In general, the following information should be included in this section: ▪ A description of the participating organizations and personnel and identification of their roles and responsibilities. This may include for example, the operators, an event recorder, witnesses, and/or engineering support. Some agencies prefer to have contractors not around during validation, others want access to them in case questions or problems arise. ▪ Identification of the location of the activity, that is, the place, or places, where the progress must be observed. ▪ The schedule of when Validation will occur including a sequencing of the events that make up the Validation activity. ▪ The system configuration for all of the activities, including the main system hardware and software and any supporting equipment, software, or external systems. Several configurations may be used depending on the type of system and type of development that was just completed. For instance, a signal upgrade may have a smaller configuration to validate than a new TMC. ▪ Identification of the documents to be prepared to support the validation, including any special scenarios, a Validation Report and descriptions of special test equipment and software. ▪ Details on the actual conduct of the activity, including: − Notification of participants − Emphasis on the management role of the operators − Procedures for approving last minute changes to the scenarios ▪ The processes for handling anomalies, including recording of critical information, resolution of the cause of a failure [e.g. fix the software, reset the system, change the ConOps, record potential future changes], and determination of any retesting activities necessary.

SECTION	CONTENTS
5.0 Validation Event Identification	This section is where we identify the specific scenarios and other events to be performed. For Validation, scenarios can be clustered around a typical operator's use of the system. It may also be structured around the operational needs defined in the baseline ConOps. There may also be events setup to exercise the final system during failure modes or even situations such as loss of power to the building or a flood near the field equipment. The actual grouping of Needs into a validation event is arbitrary. They should be related and easily combined into a reasonable set of repeatable actions. Each event should contain at least the following information: ▪ A description name and a reference number ▪ A complete list of the needs to be validated. For ease of tracing into the Validation Plan and other documents, the Needs are given numbers. They can be accurately and conveniently referenced without repeating all the words from the ConOps. ▪ A description of the objective of the event, usually taken from the wording of the Needs ▪ Any data to be recorded or noted during the event. ▪ A statement of the pass/fail criteria. Often, this is just a statement that the system satisfies the needs. ▪ A description of the system configuration. That is a list of the hardware and software items needed and how they should be connected. Often, the same configuration is used for several events/scenarios ▪ A list of any other important assumptions and constraints necessary for conduct of the event

VALIDATION REPORT TEMPLATE

SECTION	CONTENTS
Title Page	The title page should follow the Transportation Agency procedures or style guide. At a minimum, it should contain the following information: ▪ VALIDATION REPORT FOR THE [insert name of project] AND [insert name of transportation agency] ▪ Contract number ▪ Date that the document was formally approved ▪ The organization responsible for preparing the document ▪ Internal document control number, if available ▪ Revision version and date issued

SECTION	CONTENTS
1.0 Purpose of Document	This section identifies the type of validation performed. For instance, the activity may validate the entire system, a sub-system, the deployment at a site, or any other validation activity called for in the Program Plan or in the SEMP. This section can be taken from the applicable Validation Plan.
2.0 Identification of the Configuration under test	This section identifies the equipment and software validated. It also identifies all equipment and software necessary for this validation activity that is external to the system / sub-system configuration. This may include special test equipment and any external systems with an interface to the system. This section can be taken from the applicable Validation Plan and updated to reflect the actual system as delivered.
3.0 Individual Validation Reports	This section summarizes the purpose and results of each event performed in the applicable Validation Plan. Special attention is paid to any situation where a failure (or deviation from the expected System performance) occurred and how the failure was resolved. This section covers: ▪ Event overview and results ▪ Completed Validation Plan pages annotated with results ▪ Description of each anomaly, if any, from the expected result called for in the Validation Plan ▪ Any back-up data or records related to the experience ▪ Details of the resolution of each anomaly, including procedure modifications, software fix, re-testing and results, regression testing and results, and required document changes [including changes to the ConOps, new requirements for next version].

6.11 Operations & Maintenance Plan Template

6.11.1 Purpose of this Document

This document describes how the finished system will be operated and maintained. Operation and maintenance activities were described in Section 3.3.12. These templates describe the scope and content of the Operation & Maintenance Plan, which covers both hardware and software.

The Operation & Maintenance Plan is prepared incrementally during system implementation, and revised as needed during on-going system operation. The first version should be produced as early in the project as possible, to ensure that operation and maintenance needs are understood and planned for. This initial version may be quite limited in content, focusing on issues such as staffing, funding, and documentation that need to be worked on well in advance of system startup. Details of specific operation and maintenance activities can be added as needed, and after the system is developed and its specific characteristics are known.

The Operation & Maintenance Plan is separate from operating manuals and maintenance manuals provided by system or component developers or suppliers. Those documents describe detailed procedures, whereas the O&M Plan describes resource organization, responsibilities, policies, and general procedures. For example, the O&M Plan may say that the system administrator will ensure that databases are backed up daily. An operation or maintenance manual will describe how to do a backup.

6.11.2 Tailoring this Document to Your Project

Operation and maintenance activities can usually be described in a single plan. However, for large or complex systems it may be appropriate to prepare a maintenance plan separately from the operation plan. Similarly, large or complex systems may warrant separate plans for specific aspects of operation or maintenance, including configuration management, staff training, data management, safety, and security.

Some sections of the document described below may not be needed for a particular system. Other systems may need additional sections not mentioned here. The plan should provide sufficient information for the system to be effectively operated and maintained, even in the event of a complete turn-over of the personnel originally involved.

The project Concept of Operations, System Requirements, and Design Documents will provide initial guidance as to the extent and nature of operation and maintenance activities. As specific components are procured and implemented, the plan can be updated and expanded to include more specific information.

For small or simple systems, configuration management may be covered within the Operation and Maintenance Plan. Otherwise it will be the subject of a separate plan [see 6.3 Configuration Management Plan]. The two are closely related.

Since the Operation and Maintenance Plan needs to be used and updated throughout the life of the system, it is not appropriate to merely make it a section within the Project Plan.

6.11.3 Checklist: Critical Information

- ☑ Does the Operation and Maintenance Plan answer all the questions of who, what, where, and when concerning operation and maintenance?
- ☑ Does the Plan identify the personnel responsible for operation and maintenance?
- ☑ Does the Plan identify the human resources and facilities, including tools, needed for operation and maintenance?
- ☑ Does the Plan identify funding sources for on-going operation and maintenance?
- ☑ Does the Plan describe the operation and maintenance activities to be performed?
- ☑ Does the Plan describe the checks to be made, and the data to be collected, for health and performance monitoring?
- ☑ Does the Plan cover periodic reporting of system health and performance to provide feedback to management on the effectiveness of operations & maintenance?
- ☑ Does the Plan address the training of operators and maintenance personnel?
- ☑ Does the Plan address safety and security?
- ☑ Does the Plan identify other documents used in operations & maintenance, such as relevant policy directives, system configuration documentation, and operating & maintenance manuals?
- ☑ Does the Plan address system testing and configuration documentation updates [may be dealt with in a separate Configuration Management Plan], following configuration changes, repairs, and upgrades?
- ☑ Does the Plan address preventive maintenance as well as reactive maintenance?
- ☑ Does the Plan address expected life and end-of-life replacement or upgrade?

6.11.4 Template

OPERATION & MAINTENANCE PLAN TEMPLATE

The following format is one example of many alternatives. If the new system is one of multiple systems operated and maintained by the same personnel, the material described here may be incorporated in an existing Operations & Maintenance Plan covering multiple systems.

SECTION	CONTENTS
Title Page	The title page should follow the Transportation Agency procedures or style guide. At a minimum, it should contain the following information: ▪ OPERATION & MAINTENANCE PLAN FOR THE [insert name of system] ▪ The organization responsible for preparing the document ▪ Internal document control number, if available ▪ Revision version and date issued
1.0 Purpose of Document	This section identifies the scope and purpose of the O & M Plan. It explains how it fits in with related documents such as the Configuration Management Plan, operating manuals, and maintenance manuals. Included is a brief description of the system being operated and maintained. Also covered are its stakeholders, such as agencies and departments within agencies that rely on its successful operation. The system description should list all the system elements that are the subject of this document, including auxiliary equipment and facilities such as any special air conditioning, communications links, special lighting, and/or special furniture.
2.0 Facilities and Resources	This section identifies the facilities and resources to be used for system operation and maintenance. It should cover at least the following elements: ▪ Personnel, including positions, general qualifications, and specialty skills needed and a percentage of time dedicated to system operation or maintenance, if not full time. ▪ Building space, including for example, rooms and space within rooms, also specialty areas such as: workshops, raised floors, additional air conditioning, additional power, and communications trunks. ▪ Furniture, equipment, and tools. ▪ Training needed for operations & maintenance personnel, including off-site courses, on-site courses, and hands-on training on the system itself. ▪ Funding, including the amount needed each year and sources. Attempt to predict future costs, including unusual items such as end-of-life replacement.

SECTION	CONTENTS
3.0 Operations	This section describes policies and high-level procedures governing operation of the system. Minimally, it should address the activities described in the project's Concept of Operations and any other activities needed to achieve the project's objectives. In general, the following information should be included in this section: ▪ A clear statement of system operation goals and expectations ▪ Hours of operation [if not continuous] or the conditions that trigger the commencement and termination of intermittent system operation ▪ Operation activities [including monitoring of automated processes] needing human involvement and the personnel responsible for each ▪ Backup facilities, personnel, and procedures for invoking use of backups ▪ Interaction and coordination needed with other systems and personnel, including policies for decision making, overrides, and notification in the event of competing interests ▪ Special procedures and interactions which apply in the event of major emergencies ▪ Parameters used to monitor the effectiveness of system operation. Also, how those data are to be collected and reported ▪ Policies on security, covering access to the system [e.g., log in/out, password management, remote access, and firewalls.], and fire and safety. ▪ Procedures related to system health monitoring and reporting, initiation of maintenance actions, and hand-off between operation and maintenance personnel at both the start and end of maintenance actions ▪ Policies regarding data collection and archiving, including what data are to be stored for how long ▪ Deployment of interfacing systems [especially by other agencies] that must precede deployment of a system feature

SECTION	CONTENTS
4.0 Maintenance	This section describes policies and high-level procedures governing maintenance of the system. It should address both proactive [preventive] and reactive [corrective] activities needed to keep the system fully operational. In general, the following information should be included in this section: ▪ Preventive maintenance activities and the time schedule or other triggers for each activity ▪ Corrective maintenance activities, the relative urgency of each, and the maximum target response and correction times for each type of fault ▪ Policies with regard to purchase of spare equipment, manufacturer or vendor maintenance agreements or extended warranties, and third party maintenance contracts ▪ Parameters used to monitor the effectiveness of system maintenance, and how those data are to be collected and reported ▪ Procedures for coordination with operations personnel and activities ▪ Demarcation of responsibilities relative to maintenance by other parties and procedures for coordination with personnel responsible for interconnected systems or components that are not part of this system
Appendix	A list of the names and contact information of personnel currently assigned to system operation and maintenance. Include the names and contact information of personnel in other parts of the organization or in other organizations, including emergency response services, with which system operations & maintenance personnel must interact.

7 Glossary and Acronyms

This glossary and acronym list includes all the key terms and acronyms used in this document; as well as others that often appear in systems engineering. While these are many of the definitions that can be used, each project will have its own set of terms that need to be defined and adopted as part of the initial tasks.

7.1 Glossary

Acceptance: An action by an authorized representative of the acquirer by which the acquirer assumes ownership of products as a partial or complete performance of contract.

Acceptance criteria: The criteria a product must meet to successfully complete a test phase or meet delivery requirements.

Acceptance test: Formal testing conducted to determine whether or not a system satisfies its acceptance criteria and to enable the acquirer to determine whether or not to accept the system.

Acquirer: An organization that procures products for itself or another organization.

Appraisal: In CMMI, an examination of one or more processes by a trained team of professionals using an appraisal reference model as the basis for determining, at a minimum, strengths and weaknesses. (See also assessment.)

Approval: Written notification by an authorized representative of the acquirer that the developer's plans, design, or other aspects of the project appear to be sound and can be used as the basis for further work. Such approval does not shift responsibility from the developer to meet contractual requirements.

Architecture: The organizational structure of a system, identifying its components, their interfaces, and a concept of execution among them.

Assembly: A number of parts or sub-assemblies, or any combination thereof joined together, to perform a specific function and capable of disassembly.

Assessment: In CMMI, an appraisal that an organization does internally for the purposes of process improvement.

Audit: An independent examination of a work product/process or set of work products/processes to assess compliance with specifications, standards, contractual agreements, or other criteria.

Authentication: The procedure [essentially approval] used by the approval authority in verifying that specification content is acceptable. Authentication does not imply acceptance or responsibility for the specified item to perform successfully.

Baseline: An approved product at a point in time. Any changes made to this product must go through a formal change process.

Components: Components are the named "pieces" of design and/or actual entities [sub-systems, hardware units, software units] of the system/sub-system. In system/sub-system architectures, components consist of sub-systems [or other variations], hardware units, software units, and manual operations.

Concept [project concept]: A high-level conceptual project description, including services provided and the operational structure.

Concept exploration: The process of developing and comparing alternative conceptual approaches to meeting the needs that drive the project.

***Concept of Operations*:** A document that defines the way the system is envisioned to work from multiple stakeholder viewpoints [Users including operators, maintenance, management].

***Configuration item [CI]*:** A product such as a document or a unit of software or hardware that performs a complete function and has been chosen to be placed under change control. That means that any changes that are to be made must go through a change management process. A baseline is a configuration item.

***Configuration management*:** A discipline applying technical and administrative direction and surveillance to identify and document the functional and physical characteristics of Configuration Items [CI's]; audit the CI's to verify conformance to specifications, manage interface control documents and other contract requirements control changes to CI's and their related documentation; and record and report information needed to manage CI's effectively, including the status of proposed changes and the implementation status of approved changes.

***Configuration Management Plan*:** A plan defining the implementation [including policies and methods] of configuration management on a particular program/project.

***Contract*:** A mutually binding legal relationship obligating a seller to furnish the supplies or services [including construction] and a buyer to accept and pay for them. It includes all types of commitments, in writing, that obligate the buyer to an expenditure of appropriate funds. In addition to bilateral agreements, contracts include, but are not limited to, awards and notices of awards; job orders or task letters issued under purchase orders under which the contract becomes effective by written acceptance or performance; and bilateral modifications.

***Contractor*:** An individual, partnership, company, corporation, association or other service, having a contract with a buyer for the design, development, manufacture, maintenance, modification, or supply of items under the terms of a contract.

***Control gates*:** Formal decision points along the life cycle that are used by the system's owner and stakeholders to determine if the current phase of work has been completed and that the team is ready to move into the next phase of the life cycle.

***Cross-cutting activities*:** Enabling activities used to support one or more of the life cycle process steps.

***Data*:** Recorded information, regardless of medium or characteristics, of any nature, including administrative, managerial, financial, and technical.

***Data product*:** Information that is inherently generated as the result of work tasks cited in a Statement of Work [SOW] or in a source document invoked in the contract. Such information is produced as a separate entity [for example, drawing, specification, manual, report, records, and parts list].

***Database*:** A collection of related data stored in one or more computerized files in a manner that can be accessed by users or computer programs via a database management system.

***Database management system*:** An integrated set of computer programs that provide the capabilities needed to establish, modify, make available, and maintain the integrity of a database.

***Decomposition*:** The process of successively breaking down the system into components that can be built or procured. Functional and physical decomposition are the key activities that are used. Functional decomposition is breaking a function down into its smallest parts. For example, the function ramp metering decomposes into a number of sub-functions, e.g. detection, meter rate control, main line metering, ramp queuing, time of day, and communications. Physical decomposition defines the physical elements needed to carry out the function. For example, the ramp metering physical decomposition includes loops or video detection, WWV time [worldwide standard clock for accurate time], fiber or twisted pair for communications, 2070 or 170 controllers, and host computer.

***Design*:** Those characteristics of a system or components that are selected by the developer in response to the requirements.

***Detailed Design Document*:** The product baseline used to develop the hardware and software components of the system.

***Developer*:** An organization that develops products ["develops" may include new development, modification, reuse, reengineering, maintenance, or any other activity that results in products] for itself or another organization.

***Development model*:** A specific portion of the life cycle model that relates to the definition, decomposition, development, and implementation of a system or a part of a system.

***Development strategy*:** The way the development and deployment of the overall system will be carried out. For example, an evolutionary development strategy means that the system will be developed and deployed in multiple segments over time. These pieces are complete functional units that will perform independently from other functional pieces. Incremental development is the development of pieces that are done concurrently or nearly concurrently by the same or different development teams.

***Elicitation*:** The process to draw out, to discover and to make known so to gain knowledge and information, often used in defining needs.

***Enabling products*:** Products that enable the end product to be developed, supported, and maintained. For example, these products typically are the software compilers, prototypes, development workstations, plans, specifications, requirements, and training materials.

***End products*:** Products that perform the desired capability e.g. the hardware, software, communications, and databases.

***End-item*:** A deliverable item that is formally accepted by the acquirer in accordance with requirements of a detail specification.

***Evaluation*:** The process of determining whether an item or activity meets specified criteria.

***Evolutionary development*:** Breaking a project down into parts and developing them in serial fashion.

***Feasibility assessment*:** A pre-development activity to evaluate alternative system concepts, selects the best one, and verifies that it is feasible within all of the project and system constraints.

***Firmware*:** The combination of a hardware device and computer instructions and/or computer data that resides as read-only software on the hardware device.

***Gap analysis*:** A technique to assess how far current [legacy] capabilities are from meeting the identified needs, to be used to prioritize development activities. This is based both on how far the current capabilities are from meeting the needs [because of insufficient functionality, capabilities, performance or capacity] and whether the need is met in some places and not others.

***Hardware*:** Articles made of material, such as aircraft, ships, tools, computers, vehicles, fittings, and their components [mechanical, electrical, electronic, hydraulic, and pneumatic]. Computer software and technical documentation are excluded.

***Integrated product team*:** A team consisting of agency and contractor representatives working together.

***Integrity*:** A system characteristic that means that the system's functional, performance, physical, and enabling products are accurately documented by its requirements, design, and support specifications.

***Intelligent Transportation Systems*:** A broad range of diverse technologies which, when applied to our current transportation system, can help improve safety, reduce congestion, enhance mobility, minimize environmental impacts, save energy, and promote economic productivity. ITS technologies are varied and include information processing, communications, control, and electronics.

***Interface*:** The functional and physical characteristics required to exist at a common boundary - in development, a relationship among two or more entities [such as software-software, hardware-hardware, hardware-software, hardware-user, or software-user].

***Interface control*:** Interface control comprises the delineation of the procedures and documentation, both administrative and technical, contractually necessary for identification of functional and physical characteristics between two or more configuration items that are provided by different contractors/acquiring agencies, and the resolution of the problems thereto.

***Item*:** A non-specific term used to denote any product, including systems, sub-systems, assemblies, subassemblies, units, sets, accessories, computer programs, computer software, or parts.

***Legacy system*:** The existing system to which the upgrade or change will be applied.

***Life cycle*:** The end-to-end process from conception of a system to its retirement or disposal.

***Life cycle model*:** A representation of the steps involved in the development and other phases of an ITS project.

***Metrics*:** Measures used to indicate progress or achievement.

***Model*:** An abstraction of reality. Examples: A road map is an abstraction of the real road network. A globe is a model of the world. A simulation is a dynamic model of a time sequence of events.

***Module*:** A self-contained part of a hardware item designed as a single replaceable unit, with a specific integral electronic function. It should require no installation other than mechanical mounting and completion of electrical connection.

***National ITS Architecture*:** A general framework for planning, defining, and integrating ITS. It was developed to support ITS implementations over a 20-year time period in urban, interurban, and rural environments across the country. The National ITS Architecture is available as a resource for any region and is maintained by the USDOT independently of any specific system design or region in the nation.

***Needs assessment*:** An activity accomplished early in system development to ensure that the system will meet the most important needs of the project's stakeholders, specifically that the needs are well understood, de-conflicted, and prioritized.

***Non-conformance*:** The failure of a unit or product to conform to specified requirements.

***Operational baseline*:** The system that is currently in use, including all of the design, development, test, support, and requirements documentation.

***Operational concept*:** In the ITS industry, the operational concept defines the roles and responsibilities of the primary stakeholders and the systems they operate.

***Part*:** One piece, or two or more pieces joined together which are not normally subjected to disassembly without destruction or impairment of designed use [examples: gear, screws, transistors, capacitors, integrated circuits].

***Performance*:** A quantitative measure characterizing a physical or functional attribute relating to the execution of a mission/operation or function.

***Policy*:** A guiding principle, typically established by senior management, which is adopted by an organization or project to influence and determine decisions.

***Process*:** An organized set of activities

***Process Area*:** A cluster of related practices in an area that, when implemented collectively, satisfies a set of goals considered important for making improvement in that area. All CMMI process areas are common to both continuous and staged representations.

***Product*:** A product is a given set of items. The set could consist of system, sub-system, hardware or software items, and their documentation.

***Project*:** An undertaking requiring concerted effort, which is focused on developing and/or maintaining a specific product. The product may include hardware, software, and other components. Typically, a project has its own funding, cost accounting, and delivery schedule with the acquirer [customer].

***Project architecture*:** High-level design

***Project life cycle*:** See Life cycle

***Project Plan*:** A description [what is to be done, what funds are available, when it will be done and by whom] of the entire set of tasks that the project requires.

***Qualification testing*:** Testing performed to demonstrate to the acquirer that an item, system, or sub-system meets its specified requirements.

***Quality assurance*:** A planned and systematic pattern of all actions necessary to provide adequate confidence that management, technical planning, and controls are adequate to establish correct technical requirements for design and manufacturing. And to manage design activity standards, drawings, specifications, or other documents referenced on drawings, lists or technical documents.

***Reengineering*:** The process of examining and altering an existing system to reconstitute it in a new form. This may include reverse engineering [analyzing a system and producing a representation at a higher level of abstraction, such as design from code], restructuring [transforming a system from one representation to another at the same level of abstraction], recommendation [analyzing a system and producing user and support documentation], forward engineering [using software products derived from an existing system, together with new requirements, to produce a new system], and translation [transforming source code from one language to another or from one version of a language to another].

***Regional ITS Architecture*:** A specific regional framework for ensuring institutional agreement and technical integration for the implementation of ITS projects in a particular region.

***Regression Testing*:** Is a process that tests not only the area of change but also tests those areas that were not changed but are affected by the change.

***Requirements*:** The total consideration as to WHAT is to be done [functional], HOW well it is to perform [performance], and under WHAT CONDITIONS it is to operate. [Environmental and non-functional].

***Reverse engineering*:** The process of documenting an existing Intelligent Transportation Systems functional [what it does – requirements], physical [how it does it – design], and support [the way it was built and maintained – enabling products] characteristics.

***Risk management*:** An organized process to identify what can go wrong, to quantify and access associated risks, and to implement/control the appropriate approach for preventing or handling each risk.

***Service packages*:** Service packages provide an accessible, service-oriented perspective to ARC-IT. They are tailored to fit, separately or in combination, real world transportation problems and needs. They identify the pieces of the ARC-IT physical view that are required to implement a particular ITS service. Service packages are implemented through projects (or groups of projects, aka programs) and in transportation planning, are directly related to ITS strategies used to meet regional goals and objectives.

***Software development*:** A set of activities that result in software products. Software development may include new development, modification, reuse, re-engineering, maintenance, or any other activities that result in software products.

***Specification*:** A document that describes the essential technical requirements for items, materials or services including the procedures for determining whether or not the requirements have been met.

***Stakeholders*:** The people for whom the system is being built, as well as anyone who will manage, develop, operate, maintain, use, benefit from, or otherwise be affected by the system.

***Statement of Work*:** A document primarily for use in procurement, which specifies the work requirements for a project or program. It is used in conjunction with specifications and standards as a basis for a contract. The SOW will be used to determine whether the contractor meets stated performance requirements.

***Subcontractor*:** An individual, partnership, corporation, or an association that contracts with an organization [i.e., the prime contractor] to design, develop, and/or manufacture one or more products.

***Suppliers*:** The term 'suppliers' includes contractors, sub-contractors, vendors, developers, sellers or any other term used to identify the source from which products or services are obtained.

***Synthesis*:** The translation of input requirements [including performance, function, and interface] into possible solutions [resources and techniques] satisfying those inputs. This defines a physical architecture of people, product, and process solutions for logical groupings of requirements [performance, functions, and interface] and their designs for those solutions.

***System elements*:** A system element is a balanced solution to a functional requirement or a set of functional requirements and must satisfy the performance requirements of the associated item. A system element is part of the system [hardware, software, facilities, personnel, data, material, services, and techniques] that, individually or in combination, satisfies a function [task] the system must perform.

***System*:** An integrated composite of people, products, and processes, which provide a capability to satisfy a stated need or objective.

***System of systems*:** A system whose elements are themselves systems. A system of systems (SoS) brings together a group of systems to accomplish a mission that none of the systems can accomplish on their own. A regional transportation system is actually an SoS that is made up of signal control system(s), a freeway management system, public transportation system(s), freight distribution and logistics systems, and many others.

Systems engineering: An inter-disciplinary approach and a means to enable the realization of successful systems. Systems engineering requires a broad knowledge, a mindset that keeps the big picture in mind, a facilitator, and a skilled conductor of a team.

System specification: A top level set of requirements for a system. A system specification may be a system/sub-system specification, Prime Item Development Specification, or a Critical Item Development Specification.

Tailoring: Planning systems engineering activities that are appropriate and cost-effective for the size and complexity of the project. It may be based on cost, size, the number of stakeholders, the supporting relationships between them, complexity of systems [large number of interfaces to other systems, a large number of functions to perform, or the degree of coupling between systems.], level of ownership of system products [custom development of software owned by the agency or commercial off the shelf products], existing software products, resources, risks.

Technical reviews: A series of systems engineering activities by which the technical progress on a project is assessed relative to its technical or contractual requirements. The formal reviews are conducted at logical transition points in the development effort to identify and correct problems resulting from the work completed thus far before the problem can disrupt or delay the technical progress. The reviews provide a method for the contractor and procuring activity to determine that the identification and development of a CI have met contract requirements.

Testable: A requirement or set of requirements is considered to be testable if an objective and feasible test can be designed to determine whether each requirement has been met.

Trade-off Study: An objective evaluation of alternative requirements, architectures, design approaches, or solutions using identical ground rules and criteria.

User: The organization[s] or persons within those organizations who will operate and/or use the system for its intended purpose.

User services: User services documented what ITS should do from the user's perspective. A broad range of users were considered, including the traveling public as well as many different types of system operators. User services, including the corresponding user service requirements, formed the basis for the original National ITS Architecture development effort. The initial user services were jointly defined by USDOT and ITS America with significant stakeholder input and documented in the National Program Plan. The concept of user services allows system or project definition to begin by establishing the high level services that will be provided to address identified problems and needs.

Validation: The process of determining that the requirements are the correct requirements and that they form a complete set of requirements this is done at the early stages of the development process. Validation of the end product or system determines if the system meets the user’s needs.

Vendor: A manufacturer or supplier of an item.

Verification: The process of determining whether or not the products of a given phase of the system/software life cycle fulfill the requirements established during the preceding phase.

Work breakdown structure: A product-oriented listing, in family tree order, of the hardware, software, services, and other work tasks, which completely defines a product or program. The listing results from project engineering during the development and production of a materiel item. A WBS relates the elements of work to be accomplished to each other and to the end product.

7.2 Acronyms

AASHTO American Association of State Highway and Transportation Officials

AIAA American Institute of Aeronautics and Astronautics

ANSI American National Standards Institute

ARC-IT Architecture Reference for Cooperative and Intelligent Transportation

ASTM American Society for Testing and Materials

C2C Center to Center

C2F Center To Field

CCTV Closed-Circuit Television

CDR Critical Design Review

CE Concept Exploration

CEA Consumer Electronics Association

CFR Code of Federal Regulations

CG Control Gate

CI Configuration Item

CM Configuration Management

CMM Capability Maturity Model

CMMI Capabilities Maturity Model Integrated

CMS Changeable Message Sign

ConOps Concept of Operations

CR Change Request

DDR Detail Design Review

DOT Department of Transportation

ECP Engineering Change Proposal

EDI Electronic Data Interchange

EIA Electronic Industries Association

FAR Federal Acquisition Regulation

FCA Functional Configuration Audit

FHWA Federal Highway Administration

FSR Feasibility Study Report

FTP File Transfer Protocol

GUI Graphical User Interface

IAW In Accordance With

HLD High-Level Design

ICD Interface Control Documentation

ICWG Interface Control Working Group

IEEE Institute of Electrical and Electronics Engineers

INCOSE International Council of Systems Engineering [circa 1994]

IPT Integrated Product Teams

ISO International Organization for Standardization

IT Information Technology

ITE Institute of Transportation Engineers

ITS Intelligent Transportation System[s]

IV&V Independent Verification and Validation

MOP Measure of Progress

MPO Metropolitan Planning Organization

NA Not Applicable

NASA National Aeronautics and Space Administration

NDI Non-Developmental Item

NEMA National Electrical Manufacturers Association

NIST National Institute of Standards and Technology

NTCIP National Transportation Communications for ITS Protocol

O&M Operations & Maintenance

PD Product Development

PDR Preliminary Design Review

PM Program Manager

PMI Project Management Institute

PPP Point-to-Point Protocol

QFD Quality Function Deployment

RAD-IT Regional Architecture Development for Intelligent Transportation

RFI Request for Information

RFP Request for Proposal

RFQ Request for Quotation

ROW Right Of Way

SAE Society of Automotive Engineers

SDR System Design Review

SE Systems Engineering

SEI Software Engineering Institute [Carnegie Mellon University]

SEMP Systems Engineering Management Plan

SERF Systems Engineering Review Form

SET-IT Systems Engineering Tool for Intelligent Transportation

SI Software Item

SOW Statement of Work

SRR System Requirements Review

STIP Statewide Transportation Improvement Plan

SW Software

T & E Test & Evaluation

TIP Transportation Improvement Program

TMC Traffic Management Center

TR Technical Review

TRR Test Readiness Review

UML Unified Modeling Language

WAN Wide Area Network

WBS Work Breakdown Structure

XML Extensible Mark-up Language