Technical Memorandum No. 1

Phase I – Systems Engineering Management Plan:

A Process Review and Appraisal of the Systems Engineering Capability for the Florida Department of Transportation (FDOT)

Version 2

Prepared For:

Florida Department of Transportation
ITS Office
605 Suwannee Street, M.S. 90
Tallahassee, Florida 32399-0450
(850) 410-5600

February 20, 2003

Document Control Panel	
File Name:	W:\ITS Program\ITS GC\TWO11-SEMP\SEMP Phase 1 documents\030220 SEMP Phase I V2.pdf
Created By:	Traci Matthews
Date Created:	November 25, 2002
Version Number:	2
Reviewed By:	Gene Glotzbach
Modified By:	Pamela Hoke
Date Modified:	February 20, 2003

Table of Contents

List of Appendices

List of Tables

List of Figures

List of Acronyms

AMS .... Arterial Management System
ATIS .... Advanced Traveler Information System
ATMS .... Advanced Traffic Management System
CCTV .... Closed-Circuit Television
COTS .... Commercial Off-the-Shelf
CVO .... Commercial Vehicle Operations
DMS .... Dynamic Message Sign
DoD .... Department of Defense
EIA/IS .... Electronic Industries Alliance Interim Standard
EPIC .... Enterprise Process Improvement Collaboration
FDOT .... Florida Department of Transportation
FHWA .... Federal Highway Administration
FIHS .... Florida Intrastate Highway System
I-10 .... Interstate 10
I-4 .... Interstate 4
I-75 .... Interstate 75
I-95 .... Interstate 95
IEEE .... Institute of Electrical and Electronics Engineers
IMS .... Incident Management System
INCOSE .... International Council on Systems Engineering
ITS .... Intelligent Transportation Systems
MIL-STD .... Military Standard
MPO .... Metropolitan Planning Organization
NITSA .... *National ITS Architecture*
NTCIP .... National Transportation Communications for ITS Protocol
RTMC .... Regional Transportation Management Center
SE-CMM .... *Systems Engineering Capability Maturity Model*
SECAM .... Systems Engineering Capability Assessment Model
SECM .... *Systems Engineering Capability Model*
SEMP .... *Systems Engineering Management Plan*
SITSA .... *Statewide ITS Architecture*
Std .... Standard
TEA-21 .... Transportation Equity Act for the 21st Century
TMC .... Transportation Management Center

1. Introduction

1.1 Background

The Florida Department of Transportation (FDOT) established an Intelligent Transportation Systems (ITS) Office to coordinate the deployment of ITS projects along Florida's principal limited-access corridors. These corridors comprise the limited-access backbone of the Florida Intrastate Highway System (FIHS) and include Interstate Highways 4 (I-4), 10 (I-10), 75 (I-75), and 95 (I-95), and Florida's Turnpike. The types of ITS projects being planned or implemented include advanced traveler information systems (ATIS), advanced traffic management systems (ATMS), commercial vehicle operations (CVO), and other projects including incident management systems (IMS), surveillance using closed-circuit television (CCTV), work zone management, and ITS support projects. The ITS support projects include communications networks, software for regional transportation management centers (RTMCs), and information sharing. In addition to the FIHS, many projects involve expressways that are in the jurisdiction of local expressway authorities and other major highways that are the responsibility of metropolitan planning organizations (MPOs) and other transportation and traffic engineering divisions within local agencies. Due to this shared responsibility for Florida's major roadways, many ITS projects involve entities other than FDOT and must be coordinated with other stakeholders to ensure that new projects are fully compatible and are integrated with existing ITS infrastructures and with projects being implemented by other agencies.

Specifically, the ITS Office performs the following:

- Coordination of the deployment of statewide communications networks to support ITS;
- Coordination of the deployment of ITS along Florida's limited-access corridors and other major roadways, such as expressways;
- Coordination of the deployment of ATIS; and
- Coordination of the development of statewide information sharing for ITS.

Successful deployments of these four objectives will result in one of the largest coordinated deployments of ITS and communications infrastructure programs in the United States. In order to meet the mission, goals, and objectives of the FDOT and other stakeholders in this deployment, a comprehensive *Systems Engineering Management Plan (SEMP)* is needed to ensure that these ITS deployments result in a fully integrated, seamless, and coordinated multi-modal system that makes use of public resources in the most cost-effective and efficient manner possible. In addition, the Federal Highway Administration's (FHWA) *Rule 940, Intelligent Transportation Systems Architecture and Standards*, which implements Section 5206(e) of the Transportation Equity Act of the 21st Century (TEA-21), requires agencies implementing ITS projects utilizing federal funds not only to develop regional architectures but also to adopt a systems engineering approach for project deployments in order to qualify for ITS grants. Specifically, the *SEMP* is needed to:

- Ensure that the deployments are aligned with FDOT's overall mission, goals, and objectives;

- Ensure that the deployments result in a fully integrated, coordinated, seamless, multi-modal, and effective system;

- Ensure that public resources are being utilized with maximum cost-efficiency and effectiveness; and

- Ensure that system reliability is provided for in the maintenance and operational requirements.

As part of the development of FDOT's *ITS Corridor Master Plans* and the *ITS Plan*, a systems engineering approach was recommended for the integration of ITS deployments and the phased implementations to be used as the basis for design criteria.[1] Since regional ITS architectures are consistent with the *National ITS Architecture (NITSA)* and have been developed as part of the *ITS Corridor Master Plans*, the first requirement of *Rule 940* has been satisfied. (See *Section 2.1, Systems Engineering Defined*.)

In addition, it is a well-known and widely accepted certainty that implementing a robust *SEMP* provides agencies with an invaluable tool for maximizing the likelihood of a project's successful deployment. In fact, studies have shown that the overall success rate for projects without some form of management plan, such as systems engineering, is just over 15 percent, while the remainder of the projects were either cancelled or deemed inadequate.[2] The underlying causes for these projects' inadequacies or cancellations were invariably related to deficiencies in the management of quality, schedule, and/or budget. Because those project elements are the primary concern of systems engineering, by its very nature the implementation of a *SEMP* should improve an agency's management of them. A reasonable assumption, then, can be made that the FDOT *SEMP* will provide improved management of its ITS projects, resulting in better quality systems being implemented in shorter periods of time for less money.

1.2 Purpose

The purpose of this *Technical Memorandum* is to present the results of the systems engineering process review and appraisal, to analyze the processes and practices identified in those activities, and to make recommendations for the selection and inclusion of best practices in the comprehensive *SEMP*. As part of the appraisal, the FDOT district traffic operations personnel who oversee ITS deployments completed questionnaires designed to facilitate the assessment of systems engineering capability levels. The questions were adapted from the *Electronic Industries Alliance Interim Standard (EIA/IS) 731, Systems Engineering Capability Model (SECM)*. This model is composed of *EIA/IS 731-1*, the *SECM*, and *EIA/IS 731-2*, the *SECM Appraisal Method*.

[1] FDOT ITS General Consultant, Task Work Order No. 3, *ITS Corridor Master Plans* and *ITS Plan*, PBS&J, 2001.
[2] FHWA NHI-02-025, *Introduction to Systems Engineering for Advanced Transportation*, Course No. 137024.

The model is widely regarded as the primary standard for defining and assessing systems engineering process maturity and capability.

There are several reasons that this work is important. The most evident reason is that mentioned above – a systems engineering approach must be employed by agencies implementing ITS projects that will qualify for federal assistance. However, other reasons exist and are acknowledged and discussed herein.

This document will provide the framework for meeting the requirements of *Rule 940*, to be detailed in the comprehensive *SEMP*. In fact, the requirements of the *Rule* reflect only a portion of the total process structure to be developed in the *SEMP*. The processes required by the *Rule* focus on the Technical Category of the *EIA/IS 731, SECM,* and do not address either of the other two categories of the model, the Management or Environmental Categories. This document, as well as the actual *SEMP*, will include all three categories in the analysis and process implementation plan.

1.3 Organization of this Document

The first section provides background information and introductory material related to systems engineering in general and the FDOT ITS Program. The second section provides definitions and a review of systems engineering, systems engineering principles, information on the evolution and evaluation of systems engineering standards, and it discusses various models of systems engineering processes and their development. The third section is a summary of the results from the appraisal of FDOT process capability maturity. These results were developed from the FDOT ITS Office's district personnel responses to a questionnaire based on the *EIA/IS 731-2, SECM Appraisal Method*. The fourth section is a summary of process reviews for architecture development, systems engineering models, and ITS field element deployments. The final section provides an overall summary, conclusions, and recommendations for further systems engineering process development and improvement of systems engineering capability maturity for the FDOT *SEMP* development.

One of the major tasks of the *Phase I SEMP* development is to gather information concerning accepted standards and best practices for using the principles of systems engineering for project development in government and industry. To accomplish this task, three reports were prepared that address different areas of ITS technology. These three reports are attached to this document and are labeled *Appendices A, B*, and *C*. Another major task is to perform an assessment of the current state of systems engineering capability for the FDOT ITS Program. A questionnaire was prepared based on the EIS*/IS 731-2, SECM Appraisal Method*, and distributed to the FDOT district traffic operations personnel who oversee ITS deployments. The responses to the questionnaires were analyzed and a summary report is attached to this document as *Appendix D, A Summary Report on the Analysis of the District Responses to the Current Systems Engineering Capability Maturity Model Questionnaire*.

Appendix A, *A Process Review of the Accepted Standards and Best Practices for Developing Systems Engineering Process Models,* present six models for comparison: the Waterfall, Incremental (Iterative), Spiral, Vee, *EIA 632*, and the *Institute of Electrical and Electronics Engineers (IEEE) 1220-1998* models. As with most systems engineering process models, the processes contained in the models are very similar, but the iterations and flows are unique to each model.

Appendix B, *A Process Review of the Best Practices Using Systems Engineering to Develop ITS Architectures,* also recommends the use of the Vee Model for developing architectures. Since *Rule 940* requires agencies seeking to utilize federal funds for ITS projects to have a regional architecture and to adopt a systems engineering approach to project development, this particular process review is very important with regard to the federal *Rule*.

Appendix C*, A Process Review of the Best Practices and Process Standards for the Deployment of ITS Field Elements,* reviews stakeholder participation, requirements analysis, identifying alternatives, design, and procurement. All elements of a typical systems engineering process list are analyzed.

Appendix D*, A Summary Report on the Analysis of District Responses to the Current Systems Engineering Capability Maturity Model (SE-CMM) Questionnaire,* is a detailed summary of the appraisal of current systems engineering practices in Florida based on an appraisal conducted in accordance with the *EIA/IS 731* process tailored for application to Florida's ITS environment.

2. Systems Engineering Principles

2.1 Systems Engineering Defined

The International Council on Systems Engineering (INCOSE) defines systems engineering as follows:

> "Systems engineering is an interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem:
>
> - Operations
> - Performance
> - Test
> - Manufacturing
> - Cost & Schedule
> - Training & Support
> - Disposal
>
> Systems engineering integrates all the disciplines and specialty groups into a team effort forming a structured development process that proceeds from concept to production to operation. Systems engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs."[3]

Systems engineering is not a new discipline and has its origins in transportation systems such as traffic control and high-speed rail systems, military applications, and aerospace programs.[4] Systems engineering approaches have been used in deployments of ITS for several decades; however, no uniform or consistent processes have been adopted within the transportation community as an industry standard.

At the federal level, FHWA's *Rule 940* provides policies and procedures for implementing Section 5206(e) of TEA-21, Public Law 105-178, 112 Stat. 457, pertaining to conformance with the *NITSA* and applicable standards. As part of this rule, systems engineering is defined as follows:

---

[3] INCOSE web site, http://www.incose.org.
[4] Shinners, Stanley, *Techniques of Systems Engineering*, McGraw Hill, 1967.

"Systems engineering is a structured process for arriving at a final design of a system. The final design is selected from a number of alternatives that would accomplish the same objectives and considers the total life cycle of the project including not only the technical merits of potential solutions but also the costs and relative value of alternatives.

The systems engineering analysis shall include at a minimum:

- 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*);
- Identification of participating agencies' roles and responsibilities;
- Requirements definitions;
- Analysis of alternative system configurations and technology options to meet requirements;
- Procurement options;
- Identification of applicable ITS standards and testing procedures; and
- Procedures and resources necessary for operations and management of the system."

2.2 Intelligent Transportation Systems (ITS) Defined

There are many ways to define a system. The background and perspective of any person describing a system will invariably influence the description given. For example, to a manufacturer of an ITS field element, the device being built could be viewed as a system, with its internal electronic and software parts being defined as components. However, from the viewpoint of an ITS engineer, the device itself is a subsystem of the overall system being implemented. According to *IEEE Standard (Std.) 1220-1998*, *Standard for Application and Management of the Systems Engineering Process*, a system is "a set or arrangement of elements [people, products (hardware and software), and processes (facilities, equipment, material, and procedures)] that are related and whose behavior satisfies customer/operational needs, and provides for the life cycle sustainment of the products."

Another definition of a system might be the set of hardware and software components that are developed and delivered to a customer, consisting of the operational products and the enabling products. The operational products consist of one or more end products, which are the elements of the system that "end up" in the hands of the ultimate user. The associated processes are performed using enabling products that "allow" the end products to be put into service, kept in service, and retired from service. Additionally, the end products typically encompass more than just hardware and software, but also include people, facilities, data, materials, services, and techniques.

Obviously, these two definitions are quite similar and seem to use different words and phrases to describe the same basic idea. Many other definitions of a system exist and could be used to illustrate different ways to consider a system. For the purposes of this *Technical Memorandum*, however, the system being referred to will denote a collection of devices, means of information transfer, information processing equipment, and the techniques that accomplish the objectives of a particular ITS implementation project.

2.3 Review of Systems Engineering Standards and Processes

Systems engineering is a difficult concept to define concisely. Many different definitions exist. Two of these definitions were presented in the previous section of this document. However, each definition includes the same basic idea – that systems engineering is a structured process involving distinct steps that eventually lead to a preferred solution. Moreover, most references on systems engineering describe very similar process steps, although the terminologies used vary widely. Once familiarized with the “jargon” of a particular reference, one can easily see that the activities required to accomplish the intent of the systems engineering process are clearly demarcated and quite comparable.

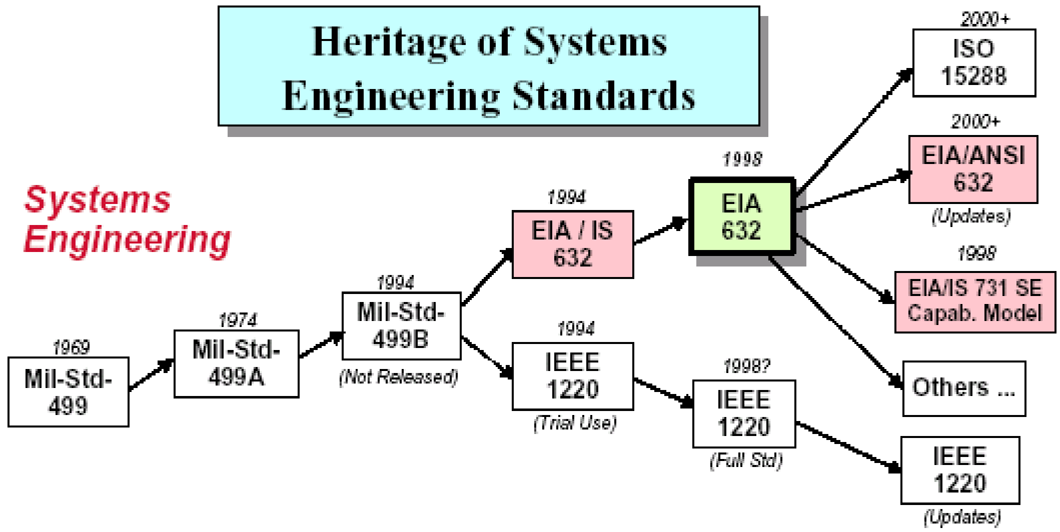
The following is a summary of the evolution of various standards in the systems engineering discipline. A more complete, detailed history can be found in the references indicated in footnotes one through four. Systems engineering standards have evolved from a federal government contract-centric approach to a commercial, voluntary-compliance approach. The focus has changed from management to process orientation.

2.3.1 Evolution of Systems Engineering Standards

Military Standard 499 (MIL-STD-499), *Engineering Management*, dated July 17, 1969, and later the A-version, dated May 1, 1974, was an early standard on the subject of systems engineering. It was produced by the United States Department of Defense (DoD) for application within the defense industry. *MIL-STD-499B*, dated in 1992, was distributed to reviewers and was an updated and significantly rewritten *MIL-STD-499A*. The DoD decreed the end of military standards other than performance specifications in June 1994 before the standard was officially released. This standard, however, has been used extensively by the Air Force. An industry working group was formed composed of representatives from the Aircraft Industry Association, DoD, the National Security Industries Association, the EIA, IEEE, and INCOSE. This working group released a "commercialized" version of *MIL-STD-499B* in December 1994 known as *EIA/IS 632*.[5] This was done with the understanding that considerably more industry input would go into a replacement version, to be called *EIA 632*. In parallel, in December 1998, the IEEE also released the commercialized standard *IEEE 1220-1998, Standard for Application and Management of the Systems Engineering Process*. Figure 2.1 illustrates the evolution of the various systems engineering standards and guidelines that dominate the systems engineering process models and capability models in practice today.

[5] *EIA 632, Interim Standard (IS): Systems Engineering*. 1994. EIA, December 1994.

Figure 2.1 – Evolution of the Systems Engineering Standards[6]/[7]



In 1992, INCOSE sponsored a working group that began to address the assessment of systems engineering capability. This group has evolved the Systems Engineering Capability Assessment Model (SECAM), which was released in July of 1996. Also, in December 1993, the Enterprise Process Improvement Collaboration (EPIC) group spun off from the INCOSE SECAM working group. This group developed a *Systems Engineering Capability Maturity Model (SE-CMM)* that was released in December 1994. This standard evolved from a software legacy.

The development of these groups meant there were now two systems engineering models in the market. INCOSE and the Director for Systems Engineering in the Office of the Secretary for Defense agreed that the two models had to come together. EPIC and INCOSE agreed to work towards a merged model, eventually called *EIA/IS 731, Systems Engineering Capability Model (SECM)*. The Frameworks Quagmire[6/8], shown in Figure 2.2, provides a detailed map of the standards evolution and adds to the potential confusion over current systems engineering standards and models.

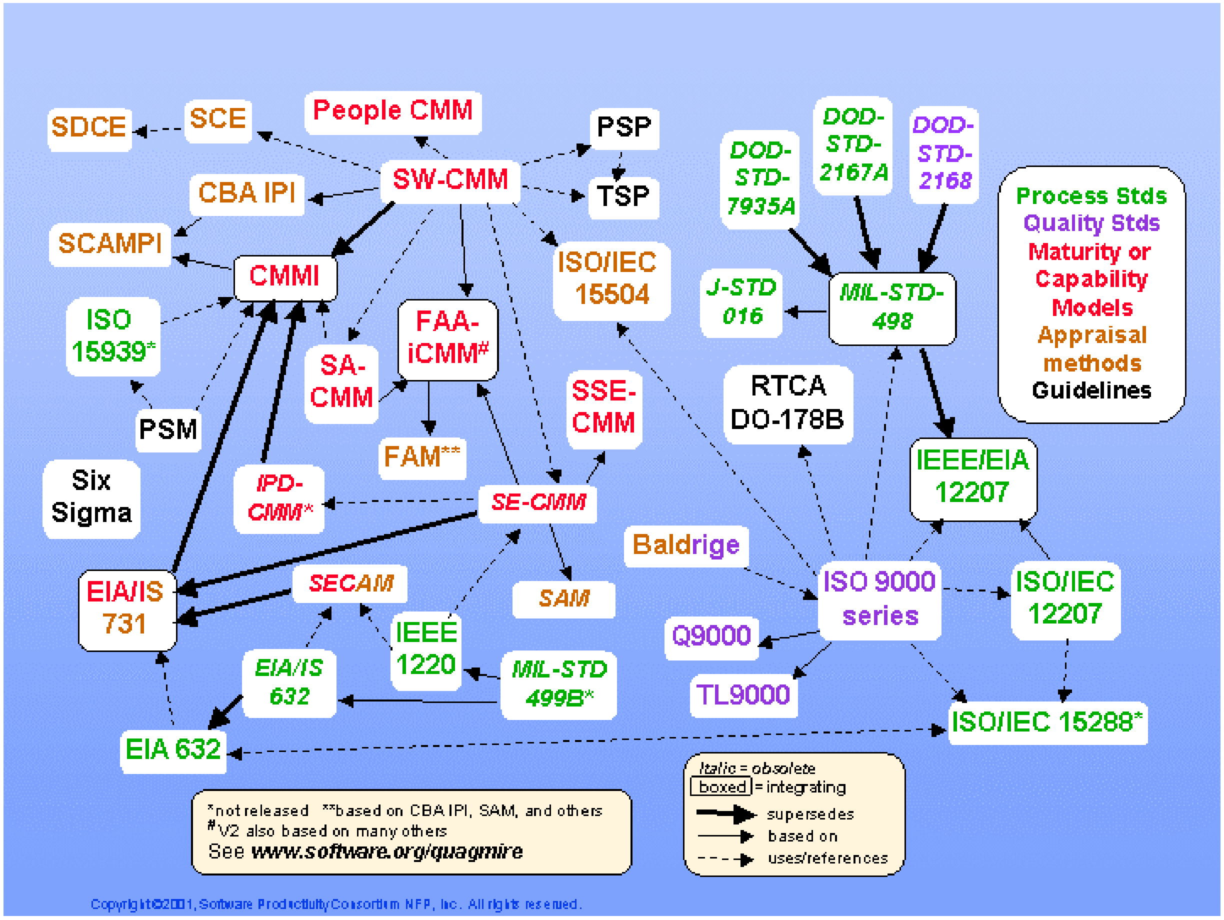
---

[6] Martin, james N., *Evolution of EIA 632 from an Interim Standard to a Full Standard*. Proceedings of INCOSE, 1998.

[7] Martin, James N., *Overview of EIA 632: Processes for Engineering a System*. Proceedings of INCOSE, 1998.

[8] Sheard, Sara A., *The Frameworks Quagmire: A Brief Look*. Proceedings of INCOSE, 1997b.

Figure 2.2 – The Frameworks Quagmire[4/8]



2.3.2 Evolution of Systems Engineering Standards

One important aspect of systems engineering, which is not a process but rather a concept that must be considered during project development, is the system life cycle. The typical system life cycle has seven phases, but it is crucial to bear in mind that the system life cycle and its description will vary based on the industry for which the system is being developed. The seven life cycle phases are:

- Defining system requirements;
- Concept exploration;
- Full-scale engineering design;
- Implementation;
- Systems integration and testing;
- Operations and maintenance; and
- Retirement, disposal, and replacement.

It is not difficult to realize that the phases of the typical system life cycle are quite similar in nature to many of the systems engineering processes. This phenomenon is not accidental, since the fundamental intent of systems engineering is to facilitate the development of systems that meet both technical and business needs throughout the system's total life cycle.

2.3.3 Evaluation of Systems Engineering Standards

An extensive body of experience collected from the industry indicates that nearly two-thirds of software development projects in the United States fail, due to either cancellation, overrunning of budgets, or delivery of software that is never put into production.[9] The reasons for these failures in descending order are as follows:

- Lack of user (i.e., customer and stakeholder) involvement;
- No clear statement of requirements;
- No project ownership;
- No clear vision and objectives; and
- Lack of planning.

A good systems engineering development model should impose discipline on developers to produce a consistent set of requirements, functional arrangements, and design solutions. The end goal is to improve productivity while at the same time providing deliverables that satisfy the product's end purpose. Almost every systems development effort varies in its specifics, so project managers and engineers need to have structured freedom to customize their work processes.

Processes – A process is a sequence of activities executed by a human or machine, often with the goal of transforming a set of inputs into outputs. A complete description of a process includes naming of the steps within the process and using models of the system in various textual/graphical abstractions. Systems engineering processes are concerned with the step-by-step development of complex engineering systems, from the identification of user needs through the specification of all components and subsystems to be designed.

Methods – A methodology is simply the implementation of a specific process. Methodologies for systems engineering development should contain an underlying model. The underlying model refers to the ensemble of objects (i.e., data types or data structures) represented, manipulated, and analyzed by the method. Modeling is a key element of systems engineering that helps to close the gap between "what is needed" and "how the system will work."

[9] Grapham, I., *Object-Oriented Methods: Principles and Practice*, Third Edition, Addison-Wesley, 2001.

2.3.4 The SIMILAR Model

One description of systems engineering processes uses the acronym SIMILAR to summarize the tasks necessary for accomplishing basic systems engineering project management.[10] The process is comprised of the following seven tasks:

- ***S****tate the Problem* – Stating the problem is the most important systems engineering task. It entails identifying stakeholders, understanding stakeholder and user needs, establishing the need for a new system, describing requirements, and defining system functions.

- ***I****nvestigate Alternatives* – Alternatives are investigated and evaluated based on performance, costs, and risks.

- ***M****odel the System* – Developing a model clarifies and validates requirements, reveals bottlenecks and fragmented activities, reduces costs, and exposes duplication of effort.

- ***I****ntegrate System Components* – Integration involves designing interfaces with legacy systems and bringing components, elements, and subsystems together to function as an amalgamated system. This activity requires extensive communications and coordination between legacy system owners and/or operators, stakeholders, and system implementers.

- ***L****aunch the System* – Launching the system means operating the system and producing outputs; i.e., making the system do what it was intended to do.

- ***A****ssess Performance* – Performance is assessed using figures of merit, technical performance measures, and metrics. Measurement is the key. <u>If it cannot be measured, it cannot be controlled. If it cannot be controlled, it cannot be improved.</u>

- ***R****e-Evaluate* – Re-evaluation should be a continual and iterative process with many parallel loops.

It is easily seen from the bulleted list above that the names of the process steps begin with the letters that spell the word “similar.” However, an important difference exists between the last two steps and the others that should be pointed out. The last two bullets, *Assess Performance* and *Re-Evaluate*, are not truly process steps on their own, but are intended to be applied to each of the other steps iteratively and concurrently. In other words, the overall process should include assessment and evaluation during and after each of the process steps in order to identify and manage potential changes to the needs, design, or environment of the system being developed.

[10] *What Is Systems Engineering? A Consesus of Senior Systems Engineers*, Bahill and Dean, 1994-2000.

2.3.5 EIA/IS 731, Systems Engineering Capability Model (SECM)

Because the *EIA/IS 731*, *SECM*, was used as the primary model for the development of this *Technical Memorandum* and because it will be the basis for the comprehensive *SEMP*, it is important to refer to the processes and process descriptions contained in that document. The SIMILAR model introduced above was included for comparison purposes and because it provides a good, concise description of the basic systems engineering processes. However, as previously mentioned regarding *Rule 940*, the SIMILAR model does not address either the Management or Environmental Categories of the *EIA/IS 731, SECM*.

The *EIA/IS 731, SECM,* architecture consists of 19 focus areas grouped into three categories. Within each focus area are themes and specific practices. A specific practice is an explicit activity that is performed to accomplish the goals and meet the intent of the focus area to which it is assigned. The specific practices are listed in ascending order of increasing systems engineering capability (e.g., a program that has only implemented the first one or two specific practices has a minimum level of systems engineering capability, whereas a program that performs most or all of the specific practices in a theme has achieved a very high level of systems engineering capability).

The *EIA/IS 731, SECM,* architecture is structured as indicated below. The numbering of each focus area indicates the category that it belongs to.

1.0 Systems Engineering Technical Category

Focus Area 1.1	Define Stakeholder and System Level Requirements
Focus Area 1.2	Define Technical Problem(s)
Focus Area 1.3	Define Solution(s)
Focus Area 1.4	Assess and Select
Focus Area 1.5	Integrate System
Focus Area 1.6	Verify System
Focus Area 1.7	Validate System

2.0 Systems Engineering Management Category

Focus Area 2.1	Plan and Organize
Focus Area 2.2	Monitor and Control
Focus Area 2.3	Integrate Disciplines
Focus Area 2.4	Coordinate with Suppliers
Focus Area 2.5	Manage Risk
Focus Area 2.6	Manage Data
Focus Area 2.7	Manage Configurations
Focus Area 2.8	Ensure Quality

3.0 Systems Engineering Environmental Category

Focus Area 3.1	Define and Improve the Systems Engineering Process
Focus Area 3.2	Manage Competency
Focus Area 3.3	Manage Technology
Focus Area 3.4	Manage the Systems Engineering Support Environment

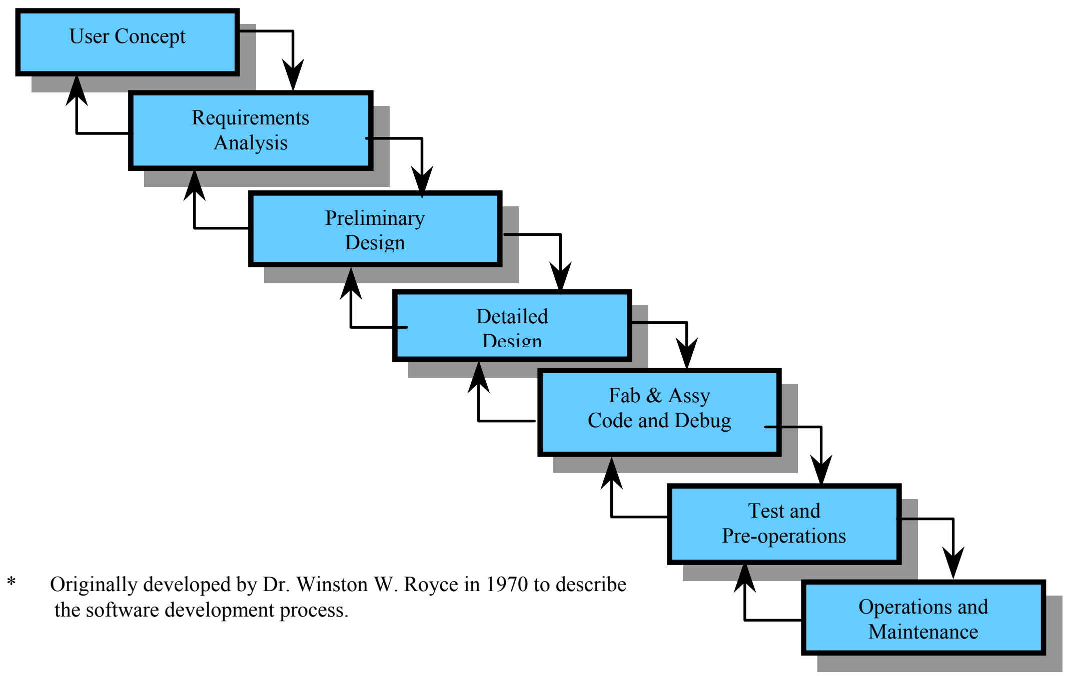
Since each theme contains a large number of specific practices and there are many themes in each focus area, the themes and specific practices will not be delineated in this document.

The most important systems engineering processes from a project-oriented perspective are those in the Technical Category. They have traditionally been the focus of most systems engineering process models and capability maturity models. As can easily be determined by perusing the focus areas in the Management and Environmental Categories, those processes are primarily aimed towards improving efficiency, managing technology, managing risk, controlling change, and improving the knowledge and skill bases of the people responsible for the design, implementation, operation, and maintenance of the system. These activities are also known as "cross-cutting" activities, since they are intended to be performed concurrently with the planning, design, implementation, and integration phases of a systems engineering compatible project.

2.3.6 The Waterfall Model

Figure 2.3 illustrates a classic Waterfall Model. In a simplified view, system life cycle development begins with the gathering of requirements and domain knowledge and ends with system deployment, maintenance, and, eventually, retirement.

Figure 2.3 – The Waterfall Process Model



The Waterfall Model was accepted as the primary systems engineering process model through most of the 1980s. Each phase of sequential development is completed via formal review before the next phase begins. The Waterfall Model is most useful when the problem and solution methods are well understood.

Variations on the Waterfall Model include those listed below.

2.3.6.1 Single-Pass Waterfall Model

The Single-Pass is the simplest view of the Waterfall Models where there is no iteration in the process activities. The three main phases of development are analysis, design, and build (i.e., construction or implementation). The key components of each phase are:

- *Analysis Phase* – The analysis phase begins with the project's inception and continues through the requirements definition. The latter may include a user needs study (i.e., what does the customer really want?) and a feasibility study (i.e., from a technical standpoint, is the project feasible?).

- *Design Phase* – The design phase covers all aspects of system design, logical design, physical design, and so forth. Often, this phase is loosely defined, with design evolving as a series of progressive decompositions towards increased technical detail.

- *Build Phase* – Now, the system is actually built, integrated into neighboring systems, and tested for proper functionality. In other words, does the system do what it is meant to?

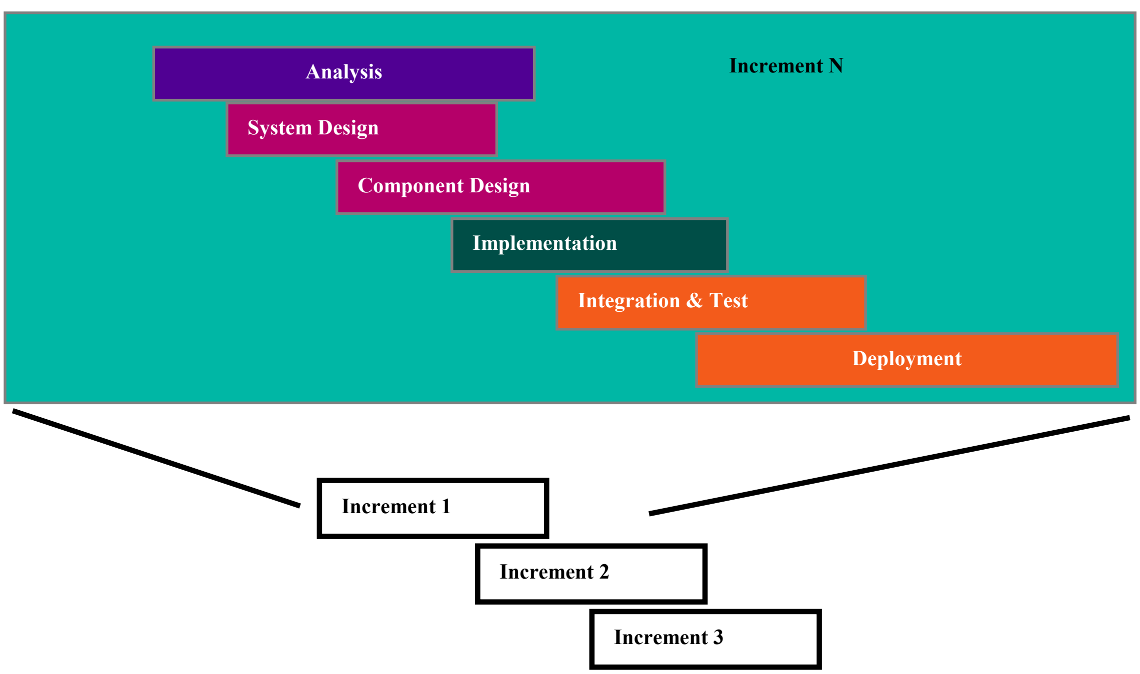
2.3.6.2 Incremental (Iterative) Waterfall Model

The Incremental Waterfall Model, sometimes known as the Iterative Waterfall Model, is illustrated in Figure 2.4.

Incremental development is defined as the development of a system in a series of versions or increments. At each increment, a subset of functionality is selected, designed, developed, and implemented. Additional increments extend the system functionality.

Often, the rework of system functionality can proceed with a reasonable amount of certainty that the desired result will be achieved (e.g., enhancements to software functionality). In these cases, this process can be modeled with a series of Waterfall Models.

Figure 2.4 – The Incremental (Iterative) Process Model



2.3.7 The Spiral Model

The Spiral Model of systems development corresponds to a sequence of Waterfall Models and is displayed in Figure 2.5. This model corresponds to risk-oriented, iterative enhancement and it recognizes that implementation options are not always clear at the beginning of a project. An implementation option may be uncertain, for example, because it is critically dependent on a technology still under development.

The radial direction of Figure 2.5 corresponds to cumulative costs incurred and the angular direction corresponds to the progress made in completing each cycle of the spiral. Each cycle of development has the following phases:

- Identification of the design and development objectives for the cycle and the alternatives that are possible to achieve the goals;

- Evaluation of the different alternatives based on objectives and constraints and, where appropriate, identification of the uncertainties and risks;

- Development of strategies such as simulation, prototyping, and benchmarking for resolving uncertainties and risks; and

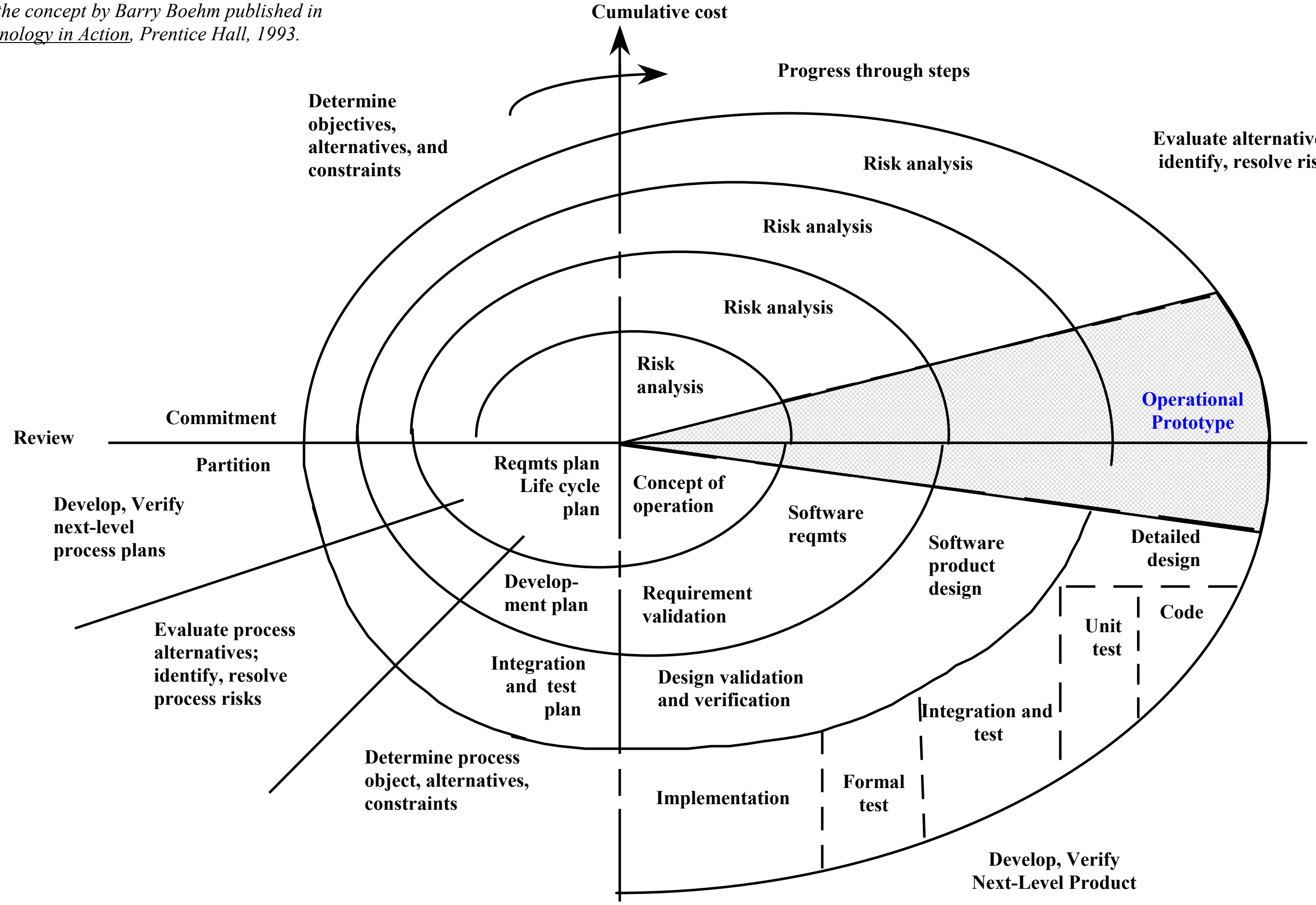
- Planning the next stage, allowing for any of the possible life cycle models to be used.

The initial Spiral Model release is a small subset of the anticipated system. Subsequent releases add capability to previous releases and each release is developed using the Waterfall Model. In some application domains, early releases have been called rapid prototypes.

A key characteristic of the Spiral Model is the assessment of management risks at regular stages in the project and the initiation of actions to counter these risks. Before each cycle, risk analysis is initiated and, at the end of each cycle, a review procedure assesses whether or not to proceed to the next loop in the spiral.

Figure 2.5 – The Spiral Process Model

Figure based on the concept by Barry Boehm published in Information Technology in Action, Prentice Hall, 1993.



2.3.8 The Vee Model

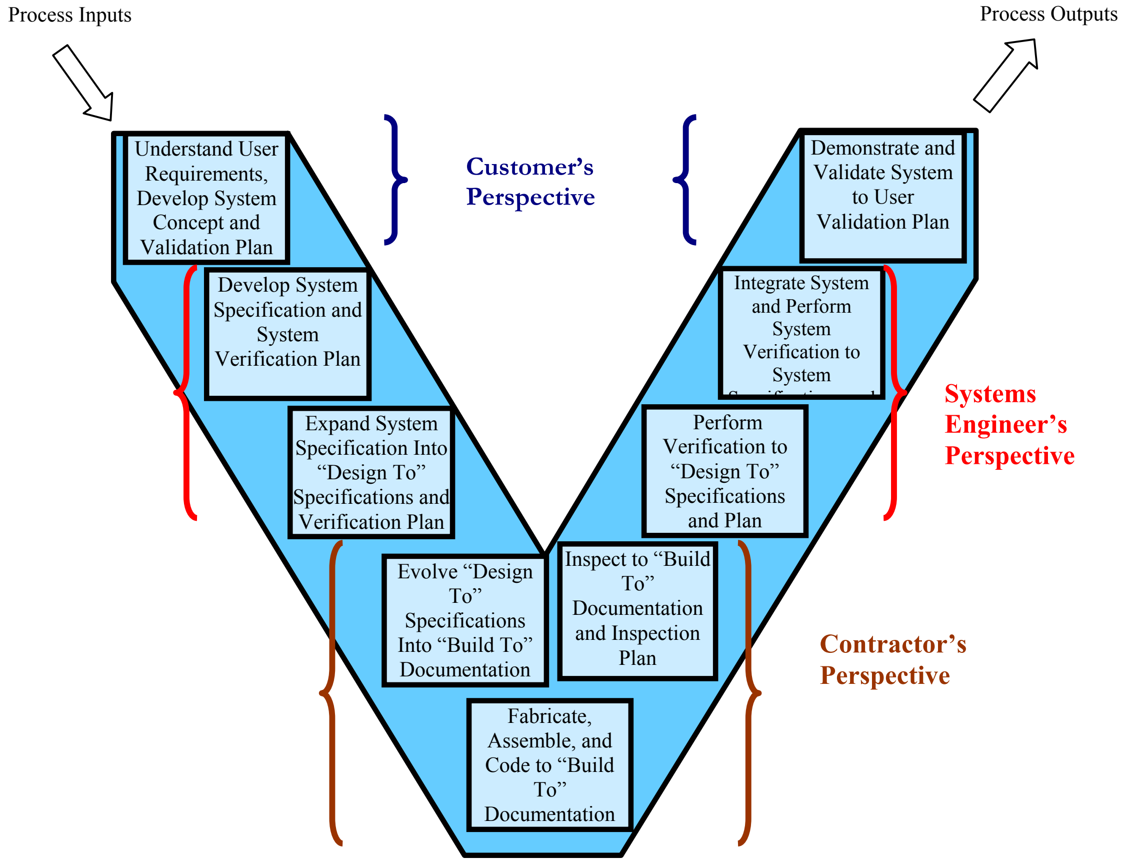
The Vee Model depicts a "top-down" development and "bottom-up" implementation approach. On the left side of Figure 2.6, decomposition and definition descends as in a traditional Waterfall Model. On the right side of Figure 2.6, integration and verification ascends as successfully higher levels of units, assemblies, and subsystems are integrated and verified, culminating at the system level. The Vee Model is a composition of three layers or perspectives of the system in increasing engineering detail:

- **User's Perspective –** This is the view of the customer or stakeholder who is interested in presenting a list of requirements and receiving a finished product that meets the requirements.

- **Systems Engineer's Perspective –** This perspective encompasses the architectural details that address the decomposition of the system-level specification into system design and the subsystems' specifications and designs. It is paired with built and tested subsystems and, finally, the tested system.

- **Contractor's Perspective –** This perspective covers the implementation process that is normally performed by contractors and/or subcontractors. In practice, the contractor's perspective is associated with component specifications and designs with fully tested components.

Figure 2.6 – The Vee Process Model



2.3.9 *The EIA 632 and IEEE 1220 Models*

The following information on the *EIA 632* model (Figure 2.7) and the *IEEE 1220* model (Figure 2.8) describes generic, problem-solving systems engineering processes that produce the specifications, baselines, and related products for project development. The processes provide the mechanism for identifying and evolving the product and process definitions of a system and applying the mechanism throughout the system life cycle to all process activities. Both the *EIA 632* and *IEEE 1220* models resemble the Incremental (Iterative) Waterfall Model discussed earlier in this report. In general, these models include not only the "operational product," which is delivered to the customer and used by the user, but also the enabling products associated with that operational product. The operational product consists of one or more end products (so called since these are the elements of the system that "end up" in the hands of the ultimate user). The associated processes are performed using products that enable the end products to be put into service, kept in service, and retired from service.

The *EIA 632* and *IEEE 1220* standards utilize the generic models to illustrate how each element of the process iterates to produce a consistent set of requirements, functional arrangements, and design solutions. The standards describe the detailed requirements of the process in Figures 2.7 and 2.8.

Figure 2.7 – The *EIA 632* Process Model

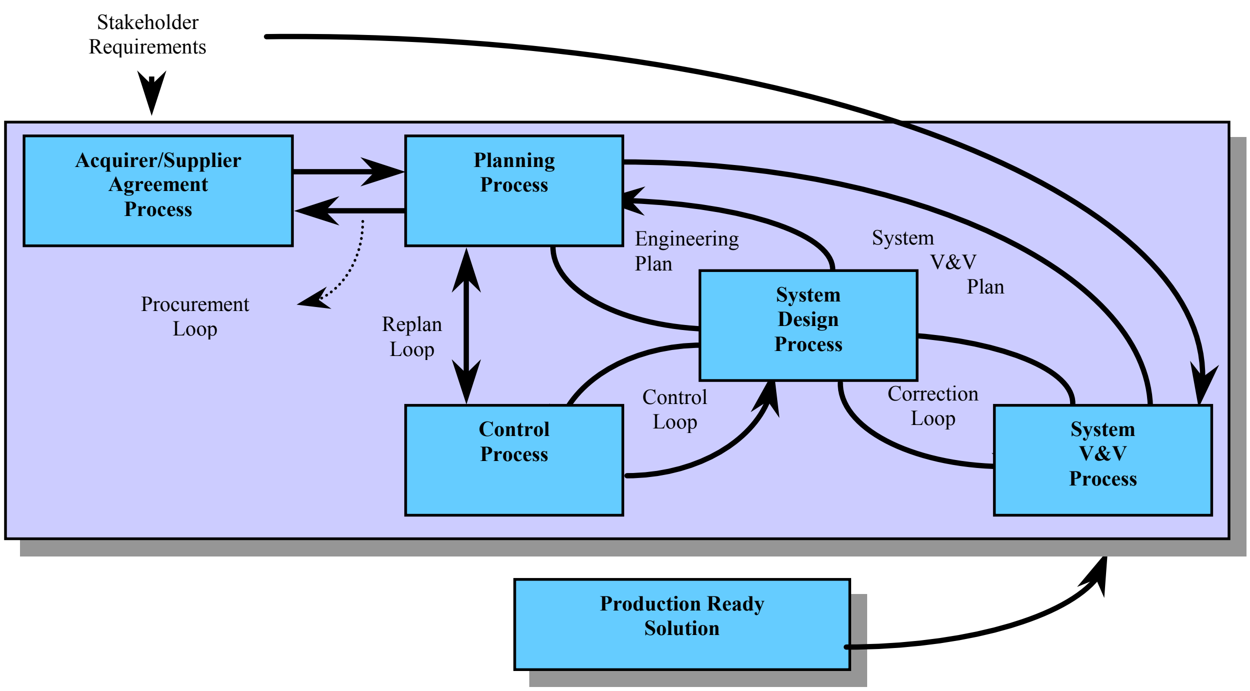
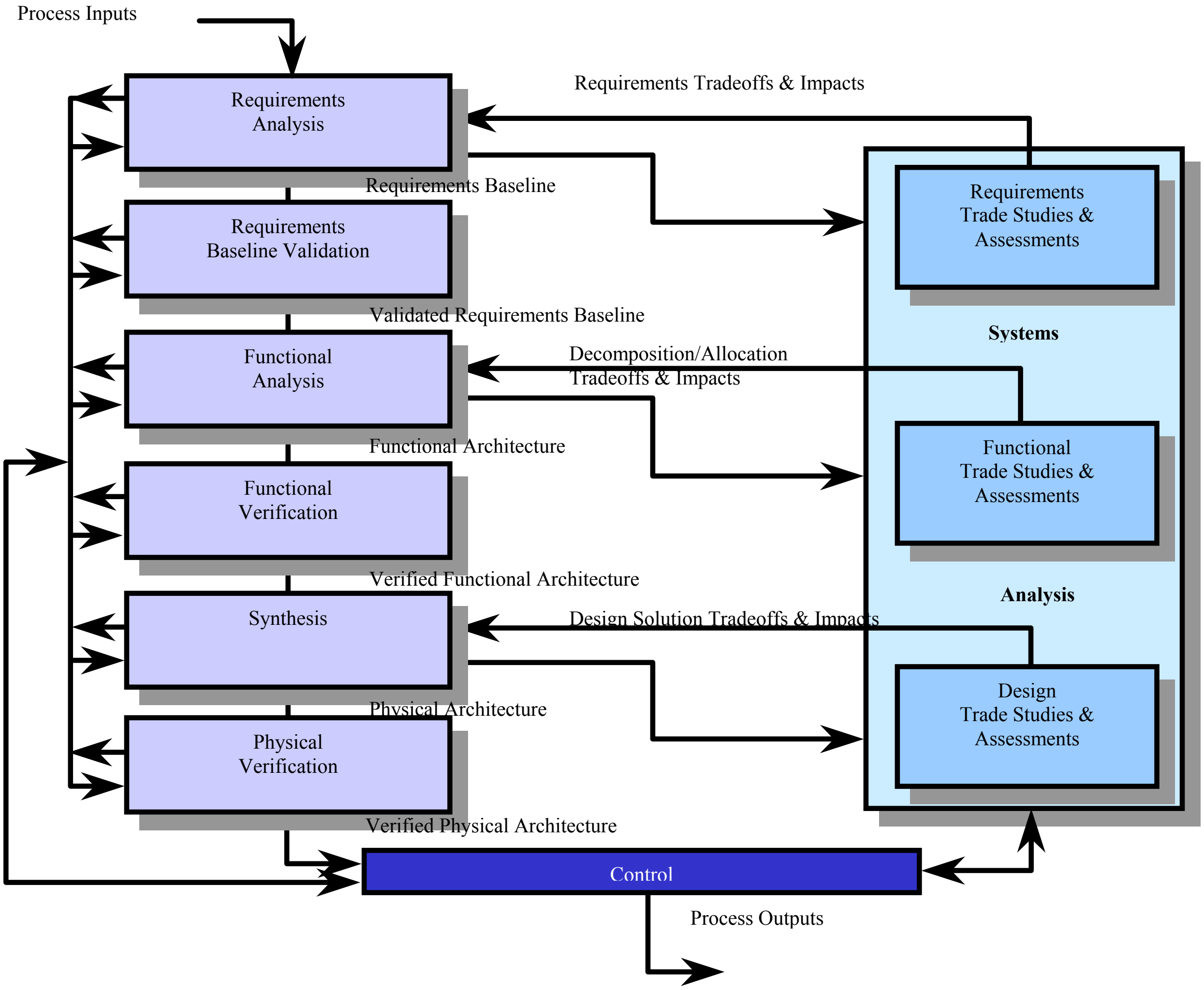


Figure 2.8 – The *IEEE 1220* Process Model



2.4 Summary and Evaluation of Process Models

Six process models were presented above – the Waterfall, Incremental (Iterative), Spiral, Vee, *EIA 632*, and *IEEE 1220* models. It is obvious in the discussion that, for the most part, all of the models are derivatives of the Waterfall Model with different variations in how the process flows and iterates. Each of these models has advantages and disadvantages associated with their use. Table 2.1 compares the models that have distinctive characteristics, namely, the Waterfall, Incremental (Iterative), Spiral, and Vee Models.

The Waterfall and Spiral Models of development lend themselves to a functional decomposition approach that follows a top-down systems design. The top-down approach has the following advantages:

- It is an orderly, systematic model for managing the size and complexity of system development.
- It can be customized to a specific system. If a portion of the system is not well understood, prototyping can be used in the analysis phase. If the project is large, risks can be reduced by scheduling incremental deliveries.

The limitations of a top-down systems design include:

- Top-down design does not take into account evolutionary changes.
- In top-down design, the system is characterized by a single function. This is a questionable concept.
- Top-down design is based on a functional mind-set. The underlying data types (or data structures) are often ignored.
- Top-down design by itself does not encourage reusability. Reusability is handled by a bottom-up synthesis of previously developed components and concepts.

The first bullet indicates the most serious shortcoming of the Waterfall and Spiral Models. With engineering and computing applications rapidly becoming more complex and businesses being forced to reorganize in order to remain competitive in global markets, the ability of an engineering process to adapt to change has recently become of paramount importance. For the Single-Pass Waterfall Model, changing requirements are the biggest cause of cost overruns and schedule slips. Users have been unable to define the requirements of a complex system without having had previous hands-on experience with the system. Applying an iteration loop to each of the process activities can alleviate this disadvantage. On the other hand, the Spiral Model, which is a Waterfall Model with a great number of iterations, can easily become corrupted unless each release is developed with discipline and standards.

The Vee Model incorporates both the top-down and bottom-up approaches in the process, where each phase is distinctively allocated to appropriate personnel resources according to their role in the system development. The Vee Model leverages the advantage of the Waterfall Model in illustrating the evolution of user requirements into preliminary and detailed designs in the top-down manner. It also accommodates integration and verification of system components through system and subsystem testing using a bottom-up path that tends to allow design reuse. The Vee Model further emphasizes the hierarchy of the decomposition process and its culmination in the build-up process that facilitates a typical development of software subsystems.

Table 2.1 – Evaluation Summary of Systems Engineering Process Models

Models	Advantages	Limitations
The Waterfall Model	• This model is the oldest and most widely used life cycle model. • It is well documented and supported. • It is accepted and well understood by customers. • The model is a logical sequence of processes and includes: o Direct mapping to phase specific processes; and o Clear boundaries of tasks.	• Real projects rarely follow a sequential flow of processes. • The model requires the customer to state all requirements up-front with high fidelity. • It is inflexible to changes in the program scope. • The customer must have patience because the product is only delivered once, at the end of the process. • Early major problems may be undetected until later stages with disastrous results.
The Incremental Model	• The model contains multiple deliveries – one for each increment. • Development phases are executed in each increment. • Planning is performed on an incremental basis. • The focus is on building the system in increments because: o Increments are cohesive system elements; and o System functionality is provided by "horizontal" slices of the system. • Requirements in an increment should be "frozen." • Standard development phases are executed in each increment. • Strong emphasis is placed on the early production of an initial capability. • Parallel development efforts are supported. • Deliveries are made to the customer from each increment. • The Incremental Model is nicely compatible with integrated product team concepts and advantages include: o Early program functionality, decreased risks, and increased customer satisfaction; o Adjustment to scope and requirements changes; o Suppression of detail on future increments; o Development cycles within an increment are efficient due to the size of the increment; and o Incremental integration is facilitated.	• The comprehensive planning and management of increment sequence can satisfy stable requirements at first but future deliveries are undefined. • Some rework is usually required of early increments. • Baseline management can be awkward. • The model requires a new mind set for integrated product teams and program management. • It requires a "fluid" relationship with the customer through: o Increment definition; and o Frozen requirements for an increment.

Table 2.1 (Continued)

<table>
<tr><th>Models</th><th>Advantages</th><th>Limitations</th></tr>
<tr><td>The Spiral Model</td><td>• This model has the potential for multiple deliveries.
• It has the potential for multiple executions of phase specific processes.
• It is better for software intensive systems.
• Planning is conducted on a spiral basis.
• The model is focused on reducing risks.
• It allows for an evaluation of risks before proceeding to a subsequent phase.
• High-risk requirements are identified, implemented, and evaluated using prototyping.
• The model provides a formal opportunity to determine completion or redirection of the development effort.
• It may be applied to individual system components independently.
• Each spiral requires customer approval to proceed.
• Standard life cycle phases are executed only in the development phase; precisely what is done is a program decision.</td><td>• The rework of prototypes is usually required.
• At its core, a Spiral may still be a Waterfall development.
• The completion opportunity is not likely to be exercised on multi-year contracts.
• The model may encounter some difficulty in overall planning and costing.
• The deliverables may not be well defined.
• Prototypes are used primarily to assess risk, not as a basis for product development.
• There is a perception of low value added quadrants.
• Each spiral requires customer approval to proceed.
• Standard life cycle phases are executed only in the development phase; precisely what is done is a program decision.</td></tr>
<tr><td>The Vee Model</td><td>• The model reflects both the top-down and bottom-up approaches with:
o Evolution of user requirements into preliminary and detailed designs as shown on the left side; and
o Integration and verification of system components through subsystem and system testing.
• The model emphasizes the hierarchy of the decomposition process and its culmination in the build-up process.
• Software subsystems can be easily included in the Vee Model.</td><td>• One must ensure that feedback is included in the process.
• There should be integration planning during the design requirements phase.
• There should be requirements verification during the integration phase.</td></tr>
</table>

3. Systems Engineering Appraisal of the FDOT Processes

3.1 Method

An appraisal of the inclusion of systems engineering principles in the total life cycle of ITS deployments in Florida was conducted as part of this study. The *EIA/IS 731-2*, *SECM Appraisal Method*, was adopted for this appraisal.

The *SECM Appraisal Method* is structured to support a wide variety of improvement activities including appraisals, process improvements, and process designs. This model is intended for internal process improvements. The process improvement efforts, using the *SECM Appraisal Method* reference model, are constructed to support the business goals of FDOT.

The components of this model are categories, focus areas, themes, and specific practices. The primary elements of the model are the focus areas, each of which is defined by a set of unique specific practices.

There are six capability levels. Each capability level has practices and attributes associated with process and non-process characteristics. These capability levels are:

- 0 – Initial;
- 1 – Performed;
- 2 – Managed;
- 3 – Dedicated;
- 4 – Measured; and
- 5 – Optimized.

Capability levels are assessed based on the performance achievements while conducting practices of the focus areas for a given category, thus indicating the capability level of that category at FDOT. The capability level assessments are documented in *Appendix A, District Responses to the Systems Engineering Appraisal Questionnaire Results*.

A primary result of the appraisal process is a rating profile covering the appraised focus areas. The rating profile correlates with the appraisal findings and the two are developed in a closely coupled process. The rating profile is developed and refined at specific points in the appraisal process. Ratings are the judgment of the appraisal team and are based on the degree to which FDOT performs all of the requirements (practices) at a given level.

The data gathered from the questionnaire is synthesized into a rating profile. This is accomplished by the following:

- A review of Section 5 of the *EIA/IS 731-1* for typical work products and practices that are found in the categories;

- An assessment of the capability level of FDOT responses based on the previous step;

- Documentation of the assessed capability level as provided in the FDOT district responses to the systems engineering questionnaire;

- Averaging the numerical equivalent scores for each capability level assessed on each district's response. The average is rounded off since there are no provisions for decimal capability levels (e.g., 1.5 becomes 2.0; 1.4 becomes 1.0; etc.);

- Evaluating the individual theme scores to assess focus area placement on the scoring template for each category; and

- Completing the scoring templates for all three categories. The scoring templates are contained in *Section 3.3, Appraisal Results*, of this document.

As a result of the above activities, the findings represent an assessment of the level of implementation of systems engineering processes within FDOT as they are related to the *SECM Appraisal Method*. These findings become the basis for the next step, which is to develop a work plan for the tasks required for developing the *SEMP* for FDOT.

A detailed summary of the responses and analysis are contained in *Appendix D*. A summary of the results and some key examples are also included.

3.2 Appraisal Questionnaire

An appraisal questionnaire was used as the basis for the assessment of current systems engineering practices. The questionnaire was developed and tailored for ITS deployments in the FDOT environment based on the *EIA/IS 731* standard and is included in *Appendix D*.

3.3 Appraisal Results

3.3.1 Technical Category

The *SECM* Technical Category's focus areas are performed at 96 percent of Capability Level 1. This indicates that there is a solid foundation of practices in the Technical Category performed at Capability Level 1. The focus areas assessed at 100 percent of Capability Level 1 are:

- Define Stakeholder and System Level Requirements;
- Define Technical Problem(s);
- Define Solution(s);
- Assess and Select;
- Verify System; and
- Validate System.

The remaining focus area in the Technical Category, Integrate System, was assessed at three-quarters of Capability Level 1, indicating the need for additional specific practice development and compliance to bring this area up to Capability Level 1.

In addition, 11 percent of the Technical Category's focus areas are performed at some part of Capability Level 2, representing an initial capability beyond Level 1. These Technical Category focus areas are:

- Define Stakeholder and System Level Requirements;
- Define Technical Problem(s); and
- Define Solution(s).

The observations for the Technical Category focus areas and their effects are summarized in Table 3.1.

Table 3.1 – Technical Category Focus Areas – Observations and Effects

Observations	Effects
• The Technical Category's specific practices are mostly performed. • Project activities are performed informally. • Non-rigorous plans and tracking are in evidence. • There is a dependence on individuals with historical project knowledge. • Work products are in evidence. • There is a general recognition of the need for activity.	• The Technical Category's activities are marginally effective and work products are of marginal utility. • Information is ad hoc. • The Technical Category's activities are driven only by immediate contractual or customer requirements and the systems engineering focus is limited.

Table 3.2 is a graphic representation of the FDOT district responses for the *SECM* Technical Category focus areas. The shaded components of Table 3.2 represent the capability level assessed for the focus areas in the Technical Category of the model.

Table 3.2 – Rating for Technical Category Focus Areas

Technical Category	Level 1 – Specific practices are performed.	Results are at least of marginal utility.	Level 2 – Specific practices are performed.	Level 2 – Generic practices are performed.	Results are at least of adequate utility.	Level 3 – Specific practices are performed.	Level 3 – Generic practices are performed.	Results are at least of significant utility.	Level 4 – Specific practices are performed.	Level 4 – Generic practices are performed.	Results are at least of a measurably significant utility.	Level 5 – Specific practices are performed.	Level 5 – Generic practices are performed.	Results are of optimum utility.
1.1 Define Stakeholder and System Level Requirements														
1.2 Define Technical Problem(s)														
1.3 Define Solution(s)														
1.4 Assess and Select														
1.5 Integrate System														
1.6 Verify System														
1.7 Validate System														
	Level 1		**Level 2**			**Level 3**			**Level 4**			**Level 5**		

3.3.2 Management Category

The *SECM* Management Category's focus areas are performed at 47 percent of Capability Level 1. This indicates that the practices in the Management Category are performed at less than half of Capability Level 1. The focus areas that have a foundation for a Level 1 Capability are:

- Plan and Organize;
- Monitor and Control;
- Integrate Disciplines, and;
- Coordinate with Suppliers.

The Management Category focus areas with little or no capability level are:

- Manage Risk;
- Manage Data;
- Manage Configurations; and
- Ensure Quality.

The observations for the Management Category focus areas and their effects are summarized in Table 3.3.

Table 3.3 – Management Category Focus Areas – Observations and Effects

Observations	Effects
• The Management Category's specific practices are not regularly performed. • There is a general failure to perform Management Category activities. • There are no easily identifiable work products. • In most cases, there is no proof that tasks are accomplished.	• The Management Category's activities and work products have little effectiveness or value. • There is no assurance of success. • Information is difficult to identify. • The driving force for activities is indeterminate. • There is no assurance of successfully completing complex management activities. • There is no focus on the principles of systems engineering.

Table 3.4 is a graphic representation of the FDOT district responses for the *SECM* Management Category focus areas. The shaded components of Table 3.4 represent the capability level assessed for the focus areas in the Management Category of the model.

Table 3.4 – Rating for Management Category Focus Areas

Management Category	Level 1 – Specific practices are performed.	Results are at least of marginal utility.	Level 2 – Specific practices are performed.	Level 2 – Generic practices are performed.	Results are at least of adequate utility.	Level 3 – Specific practices are performed.	Level 3 – Generic practices are performed.	Results are at least of significant utility.	Level 4 – Specific practices are performed.	Level 4 – Generic practices are performed.	Results are at least of a measurably significant utility.	Level 5 – Specific practices are performed.	Level 5 – Generic practices are performed.	Results are of optimum utility.
2.1 Plan and Organize														
2.2 Monitor and Control														
2.3 Integrate Disciplines														
2.4 Coordinate with Suppliers														
2.5 Manage Risk														
2.6 Manage Data														
2.7 Manage Configurations														
2.8 Ensure Quality														
	Level 1		Level 2			Level 3			Level 4			Level 5		

3.3.3 Environmental Category

The *SECM* Environmental Category's focus areas are performed at 31 percent of Capability Level 1. This indicates that the practices in the Environmental Category performed at a little over a quarter of Capability Level 1. The focus areas that have at least a 50 percent foundation for Capability Level 1 are:

- Manage Competency; and
- Manage Technology.

The Environmental Category's focus area, Manage the Systems Engineering Support Environment, was assessed at one-quarter of Capability Level 1, indicating a significant need for additional specific practice development and compliance to bring this area up to Capability Level 1.

Another Environmental Category focus area, Define and Improve the Systems Engineering Process, was assessed with no capability level of performance.

The observations for the Environmental Category focus areas and their effects are summarized in Table 3.5.

Table 3.5 – Environmental Category Focus Areas – Observations and Effects

Observations	Effects
• The Environmental Category's specific practices are not regularly performed. • There is a general failure to perform Environmental Category activities. • There are no easily identifiable work products. • In most cases, there is no proof that tasks are accomplished.	• The Environmental Category's activities and work products have little effectiveness or value. • There is no assurance of success. • Information is difficult to identify. • The driving force for activities is indeterminate. • There is no assurance of successfully completing complex management activities. • There is no focus on the principles of systems engineering.

Table 3.6 is a graphic representation of the FDOT ITS Office's district responses for the *SECM* Environmental Category focus areas. The shaded components of Table 3.6 represent the capability level assessed for the model's Environmental Category's focus areas.

Table 3.6 – Rating for Environmental Category Focus Areas

Environment Category	Level 1 – Specific practices are performed.	Results are at least of marginal utility.	Level 2 – Specific practices are performed.	Level 2 – Generic practices are performed.	Results are at least of adequate utility.	Level 3 – Specific practices are performed.	Level 3 – Generic practices are performed.	Results are at least of significant utility.	Level 4 – Specific practices are performed.	Level 4 – Generic practices are performed.	Results are at least measurably significant utility.	Level 5 – Specific practices are performed.	Level 5 – Generic practices are performed.	Results are of optimum utility.
3.1 Define and Improve the Systems Engineering Process														
3.2 Manage Competency	■													
3.3 Manage Technology	■													
3.4 Manage the Systems Engineering Support Environment	▌													
	Level 1		**Level 2**			**Level 3**			**Level 4**			**Level 5**		

3.4 Recommendations

Based on an analysis of the results, the FDOT should develop a detailed plan to identify the specific practices that are needed to improve all three categories' capability level performances. The plan should include the efforts needed to achieve a consensus on the content, format, processes, and performance criteria. The plan should be designed with interim, verifiable goals of performance at a specific capability level before advancing to a higher capability level.

3.5 Conclusions

The appraisal report in *Appendix D* gives a brief explanation of the process employed to administer the questionnaires and the capability levels and ratings of the *EIA/IS 731-2, SECM Appraisal Method*. An explanation of how the ratings for the FDOT ITS Program were developed is also included. The ratings themselves are shown in a series of three charts, one each for the Technical, Management, and Environmental Categories. A summary and recommendation section follows.

Overall, the appraisal determined that even without the introduction of any formal systems engineering approach into the ITS Program, 96 percent of the Technical Category's focus areas are performed, with six of the seven focus areas assessed at 100 percent of Capability Level 1. In addition, 11 percent of the Technical Category's focus areas are performed at some part of Capability Level 2. These results are quite good news and represent an initial capability beyond Level 1 in a program that has barely begun to implement systems engineering. However, the fact that FDOT has had standard manuals, procedures, and guidelines for project development in place for many years may contribute to this relatively high initial capability.

In the Management and Environmental Categories, the capability levels were rated at 47 and 31 percent, respectively. Although these results are significantly below the results for the Technical Category, they are not surprising for a program that has not yet formally implemented systems engineering.

4. Best Practices for Systems Engineering Capability

4.1 Intelligent Transportation Systems (ITS) Architecture Development

In performing a review of best practices for systems engineering capability involving the use and application of ITS architectures, the following results were developed:

- Previous efforts relating ITS architectures and systems engineering were reviewed and no previous efforts were found with regards to statewide or regional ITS architecture documentation. The most applicable previous efforts were an FHWA course (NHI Course No. 137024, *Introduction to Systems Engineering)* and an FHWA guidance document on developing regional ITS architectures.

- The relationship between the development of an ITS architecture and the systems engineering process was reviewed.

- Statewide and regional ITS architectures have been developed in Florida. The process used to develop them and the outputs created by them indicate that many of the steps of the systems engineering process were in fact followed in the architecture development.

- The use of statewide and regional ITS architectures to support the systems engineering analysis that will be required for the development of ITS projects was explored.

4.1.1 Systems Engineering and Architecture Development

This section addresses the application of the systems engineering process in the creation of the statewide and regional ITS architectures or, more simply, the nature of the relationship of the development of the architecture to the systems engineering process.

Regional ITS architectures have been developed for many Florida regions (e.g., statewide, district, and corridor architectures have been developed). A regional ITS architecture is defined as a “regional framework for ensuring institutional agreement and technical integration for the implementation of ITS projects in a particular region.” The regional ITS architectures will be used as a tool to support regional transportation planning and ITS project development. As part of this project development effort, a project-level ITS architecture may be created. This is defined as a “framework that identifies the institutional agreement and technical integration necessary to interface a major ITS project with other ITS projects and systems.”

Normally, the systems engineering process is applied to the development of the ITS services composed of hardware, software, and communications links. However, the systems engineering process can be applied to the development of virtually any system, including cases where the system is a regional or project ITS architecture. The following discussion uses this approach to illustrate the similarities between the process that was used to create regional ITS architectures and the systems engineering process.

In the development of regional or project ITS architectures, the concept of operations can be viewed as the description of how the architecture will be utilized. In the case of a regional ITS architecture, it will be used as a resource for transportation planning and as a source of input for the application of a systems engineering analysis for the development of projects. In the case of a project ITS architecture, it will be used to provide definition to a project and as a source of information for the performance of the systems engineering analysis required as part of the project development. This description of how the architecture will be utilized, which equates roughly to a systems engineering concept of operations, may be explicitly expressed in the systems engineering management plan or in the actual architecture documentation, where it might exist as an implementation plan.

When developing an architecture, what aspect of this development relates to the requirements portion of the systems engineering process? The user needs represent a description of the high-level requirements that must be met and a detailed list of services that must be provided by ITS projects in the region, or by the ITS project itself, can represent the detailed requirements for the architecture.

What aspect of architecture relates to the design aspect of the systems engineering process? The high-level design could be equated to the architecture inventory, which is a set of elements that represent the ITS services. The inventory also includes non-ITS elements that interface with ITS services. An example of this latter type might be an element representing the media. The detailed design could be equated to the customized market packages defined in the statewide, regional, and district ITS architectures. These provide detailed definitions of how the elements interact.

The implementation aspect of the systems engineering process could be equated to the detailed definition of the set of interfaces and information flows defined by the architecture. Validation of the architecture is usually obtained through stakeholder review and comparison to the requirements (e.g., the services and elements).

Finally, the architecture is maintained (i.e., regional ITS architectures are required by *Rule 940* to develop a maintenance plan that defines the process for updating the architecture and organizing the way that changes to the architecture are managed).

This set of connections between the development of ITS architectures and the systems engineering process is meant to indicate areas of comparison between the two, rather than provide a precise association.

4.1.2 Using Architectures to Support the Systems Engineering Process

This section addresses the use of statewide and regional ITS architectures to support the systems engineering analysis that will be required for the development of ITS projects. Specifically, this section will consider the definition and decomposition phase of the systems engineering process (i.e., the left side of the Vee in Figure 2.6) as applied to the development of ITS projects. Particularly, it will discuss:

- The relationship between the process steps that are part of the definition and decomposition phase and the requirements of *Rule 940* for systems engineering analysis in the development of ITS projects; and

- How statewide or regional ITS architectures can be used to support the definition and decomposition phase of the systems engineering process as it is related to the development of ITS projects.

Each ITS project that uses federal funds is required by FHWA's *Rule 940* and the companion Federal Transit Administration (FTA) policy to meet certain systems engineering analysis requirements. These requirements are closely related to the systems engineering process steps that are described in the definition and decomposition phase of the Vee diagram (Figure 2.6). Table 4.1 provides a mapping from the systems engineering process steps to the systems engineering analysis requirements of *Rule 940*.

Table 4.1 – Mapping of the Systems Engineering Process to *Rule 940* Requirements

Systems Engineering Process Step(s)	Corresponding *Rule 940* Requirements
Concept of Operations	• Identification of participating agencies' roles and responsibilities • Procedures (and resources) necessary for operations and management of the system
Requirements: High-Level and Detailed	• Requirements definition
Design: High-Level and Detailed	• Identification of portions of the regional ITS architecture being implemented • Analysis of alternative system configurations and technology options to meet requirements • Procurement options • Identification of applicable ITS standards and testing procedures

Is additional definition of the *Rule 940* requirements provided? The answer is no. The information in the second column of Table 4.1 is the full text of the requirements and there is no supporting guidance or other documentation to give further definition of the form or level of detail that will be expected in meeting the *Rule 940* requirements.

The following discussion identifies how statewide or regional ITS architectures can support the application of the above systems engineering process steps in the development of ITS projects. Specific instances of where the regional ITS architectures can address aspects of the process steps that represent requirements called out in *Rule 940* are also identified.

The connections made between the systems engineering process steps and the statewide or regional ITS architecture outputs draw on the two sources mentioned in *Section 4.1, Intelligent Transportation Systems (ITS) Architecture Development*, specifically NHI Course No. 137024, *Introduction to Systems Engineering*, and FHWA's *Regional ITS Architecture Guidance Document*, dated October 12, 2001. The suggestions here are in line with these documents, but carry the connections to a greater level of detail. They constitute a suggested approach, rather than best practices, since no previous statewide or regional documentation could be found on this subject. Where applicable, alternate approaches for using regional ITS architectures in support of ITS project developments have been highlighted. The approach best suited for any given project will depend on the scope of the ITS project and the details of the statewide or regional architectures available to support the project's development.

4.1.3 Developing a Concept of Operations

As described by the systems engineering process, the initial step in the development of a project is the creation of a concept of operations. When compared to the requirements of *Rule 940*, this aspect of the process relates to the following systems engineering requirements:

- Identification of participating agencies' roles and responsibilities; and
- Identification of procedures and resources necessary for the operations and management of the system.

The first aspect of the concept of operations is the identification of the stakeholders involved in the project and the roles and responsibilities of the stakeholders. To use statewide or regional ITS architectures as an input to this part of the process, the architecture or possible architectures that would apply to the project must first be identified. To do this, some idea of the geographic or service scope of the project is needed. Once the proper architecture(s) is selected, the operational concept contained within the architectures can serve as a useful starting point for the project level definition of roles and responsibilities. The best way to use the regional ITS architecture outputs depends on the level of detail in the operational concept.

If the statewide or regional ITS architecture contains a high-level description of stakeholder roles and responsibilities, then, based on the scope of the project (i.e., what aspects of the regional or statewide role or responsibility are relevant to the project), this description should be edited. If the statewide or regional ITS architecture has defined its operational concept in greater detail (e.g., role and responsibility by service or using customized market packages to define the service role and responsibility), then the appropriate portions of the detailed descriptions are selected to create an initial draft of the roles and responsibilities of the stakeholders in the context of the project.

A step beyond roles and responsibilities is the definition of the procedures necessary for the operations and management of the system. This aspect of the concept of operations addresses a portion of the *Rule 940* systems engineering requirements. A definition of the procedures needed for the stakeholders to use the system(s) that the project is creating and upgrading might include:

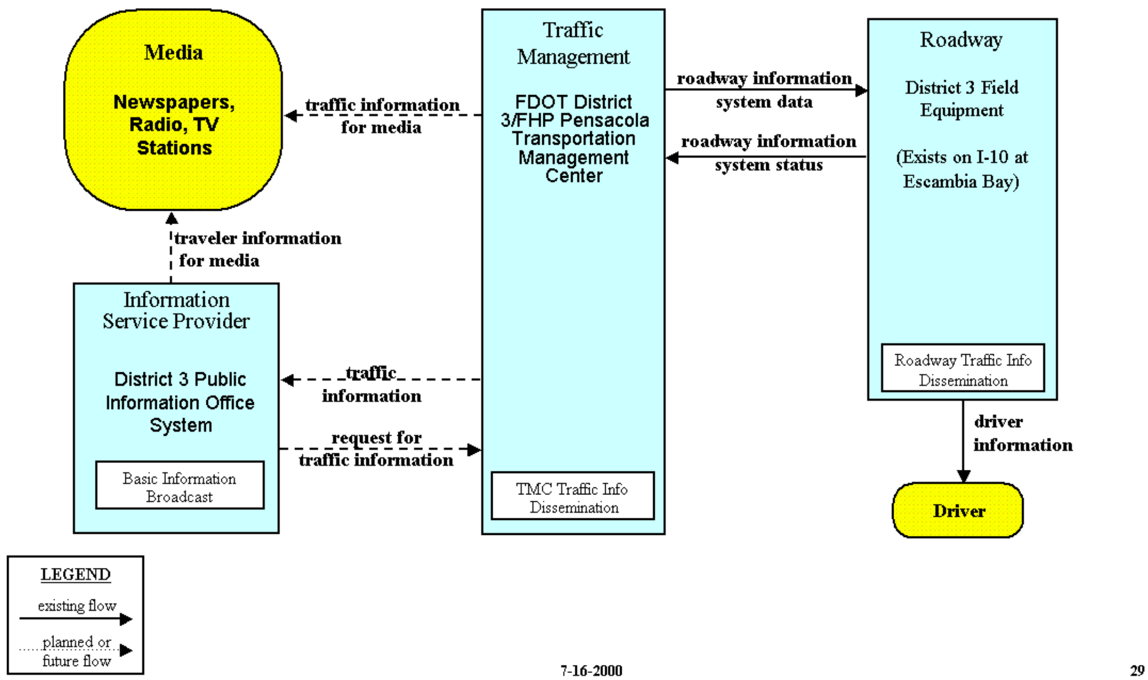
- Activities to be performed;
- Organizational relationships and responsibilities;
- Information flows;
- Message priorities;
- Archiving needs; and
- Administration, including security.

Here again, the statewide or regional ITS architecture can provide information to support development of these procedures. The customized market packages contain information that can be further customized to create a definition of the procedures associated with the project. In identifying the market packages that describe the intended transportation service(s) of the project, an indication of the activities to be performed (e.g., the equipment packages that show up on the diagrams), the information flows, and the organizations/elements involved is made. The suggested approach would then be to further customize these diagrams to reflect the scope of the project as well as the current versus planned procedures, create a textual description of the procedures to go along with the diagrams, and obtain input from the affected stakeholders regarding the completeness and accuracy of the procedures.

Consider the following example from the FDOT District 3 regional ITS architecture. Suppose the project in question has as one of its aspects the dissemination of traffic information from the FDOT District 3 Tallahassee RTMC. The market package from the web-based version of the architecture is shown in Figure 4.1. This figure identifies the elements and interfaces for this service. As shown in the market package, there are two paths for traffic information to be sent to the media – directly from the transportation management center (TMC) or through the District 3 Public Information Office. The decision must be made whether the project will implement only one or both interfaces, and whether procedures will be put into place regarding the dissemination of information to the media. Editing the diagram and describing the procedures necessary provides a good concept of operations for this aspect of the project.

Figure 4.1 – Example of Regional ITS Architecture Market Package

ATMS06 – Traffic Information Dissemination
FDOT District 3



4.1.4 Requirements

The next step in the systems engineering process is the development of requirements. These requirements focus on "what" the system must do, not "how" the system does it, and include:

- Functions;
- Expected outcomes;
- Definition of expected interfaces; and
- Performance objectives.

The requirements defined should be based on the concept of operations developed previously. When considered against the requirements of *Rule 940*, this step of the process relates to the requirements definition aspect of the systems engineering analysis.

There are many types of requirements that should be developed for the project, including:

- Functional requirements;
- Interface requirements;
- Data requirements (i.e., what type of information should be stored);
- System life cycle cost requirements;
- Performance requirements; and
- Testing requirements.

As indicated in the Vee diagram (Figure 2.6), the requirements definition entails both high-level and detailed requirements.

As part of the regional ITS architecture requirements in *Rule 940*, statewide or regional ITS architectures contain a set of functional requirements that can serve as a starting point for the definition of project requirements. These functional requirements identify the existing and planned functions of the key elements in the architecture. The current versions of Florida's district regional ITS architectures and Florida's *Statewide ITS Architecture (SITSA)* were created prior to the publishing of *Rule 940* and an explicit description of functional requirements was not done for these architectures. However, the architectures do contain pertinent functional information that can be accessed in several ways.

Consider the following example from the FDOT District 3 regional ITS architecture. Suppose the objective of the project is to add incident management capability to the planned FDOT District 3 Tallahassee RTMC. The element in question is defined in the web-based architecture by the web page that is partially shown in Figure 4.2.

Selecting the Functionality Details link shown in Figure 4.2 will lead to a definition of the equipment packages that may be applicable to the element. A portion of the web page is shown in Figure 4.3. Note that the names and descriptions of the equipment packages represent an initial set of functional requirements for the TMC element. The suggested approach is to copy the list of equipment packages and then edit them as needed to create the functional requirements for the project element.

Figure 4.2 – Example of Florida's District ITS Architecture Elements Web Page

Figure 4.3 – Example of Equipment Package Web Page

4.1.5 Design

The final steps in the definition and decomposition portion of the systems engineering process are the high-level design and the detailed design. And what is design? As defined in the NHI's *Introduction to Systems Engineering* course referenced earlier, design is the:

- Appropriate selection of system components and their interconnection so as to meet the system requirements; and
- Preparation of specifications that describe the design.

When considered against the requirements of *Rule 940*, this step of the process relates to the following aspects of the systems engineering analysis:

- Identification of portions of the regional ITS architecture being implemented;
- Analysis of alternative system configurations and technology options to meet the requirements;
- Procurement options; and
- Identification of applicable ITS standards and testing procedures.

Using the process defined in the referenced course, the following information contains the basic aspects of the design step(s) and how statewide or regional ITS architectures can support these step(s).

The first step in developing the design is to identify the systems and interconnections needed to meet the project requirements (i.e., identify the portions of the statewide or regional ITS architectures that the project will implement). The Florida district regional ITS architectures and the *SITSA* have all been developed using a software product called Turbo Architecture. This software tool is used to create a database representation of a regional or project ITS architecture. (Note: Several of the other regional ITS architectures developed within the state have also made use of this tool to define the elements and information flows in a region.)

The best way for a project development effort to identify the systems and interconnections needed is by opening the relevant statewide or regional ITS architecture in Turbo Architecture and creating a project architecture that identifies the elements, market packages, and information flows that define the scope of the project from the regional ITS architecture. The statewide or regional ITS architecture can serve as an excellent start to the description of the project architecture, but additions, subtractions, or changes will probably be needed to create an architecture that completely meets the defined requirements. Once these changes are entered into the Turbo Architecture file, then outputs describing the project can be created for distribution to affected stakeholders. In addition, the changes from the statewide or regional ITS architecture should be fed back into the maintenance process so that the project will be accurately reflected as it is developed. Completion of this step clearly satisfies the *Rule 940* requirement for identification of the portions of the regional ITS architecture being implemented.

The next step in the design process is to analyze alternative configurations and technology options to meet the requirements. Alternate configurations may entail alternate architectures (i.e., identifying options for the connections of the project's elements). The recommendation is for the group performing the initial design (whether a private contractor or a public agency) to identify several alternative architectures, identify the strengths and weaknesses of the alternatives, and select the best alternative based upon selection criteria that has been created by or discussed with the project management team. Typical issues that must be investigated are the identification of alternative architecture choices and, if the project involves the integration of interagency systems, whether the connections will be made as a series of point-to-point connections or through some central communications hub. The pieces of these alternative architectures may in fact already exist within the regional ITS architecture (which often reflects more than one way to architect an ITS service), so that the selection of alternatives becomes nothing more than creating several alternate project architectures using the Turbo Architecture tool.

Technology choices can play a key role in ITS project design. It is recommended that technology options for key elements in the project be considered and that the selection of those technologies be based upon selection criteria that has been created by or discussed with the project management team. This selection is done within the context of the project's requirements (i.e., the chosen technologies meet the requirements based on the selection criteria). Another set of alternatives that could be examined at this step is procurement options. For example, the selection of commercial off-the-shelf (COTS) products that meet the requirements of the project may be made as a result of an investigation into their availability.

Completion of the analysis of alternate architectures, technologies, and procurement options satisfies the *Rule 940* requirements for:

- Analysis of alternative system configurations and technology options to meet requirements; and
- Procurement options.

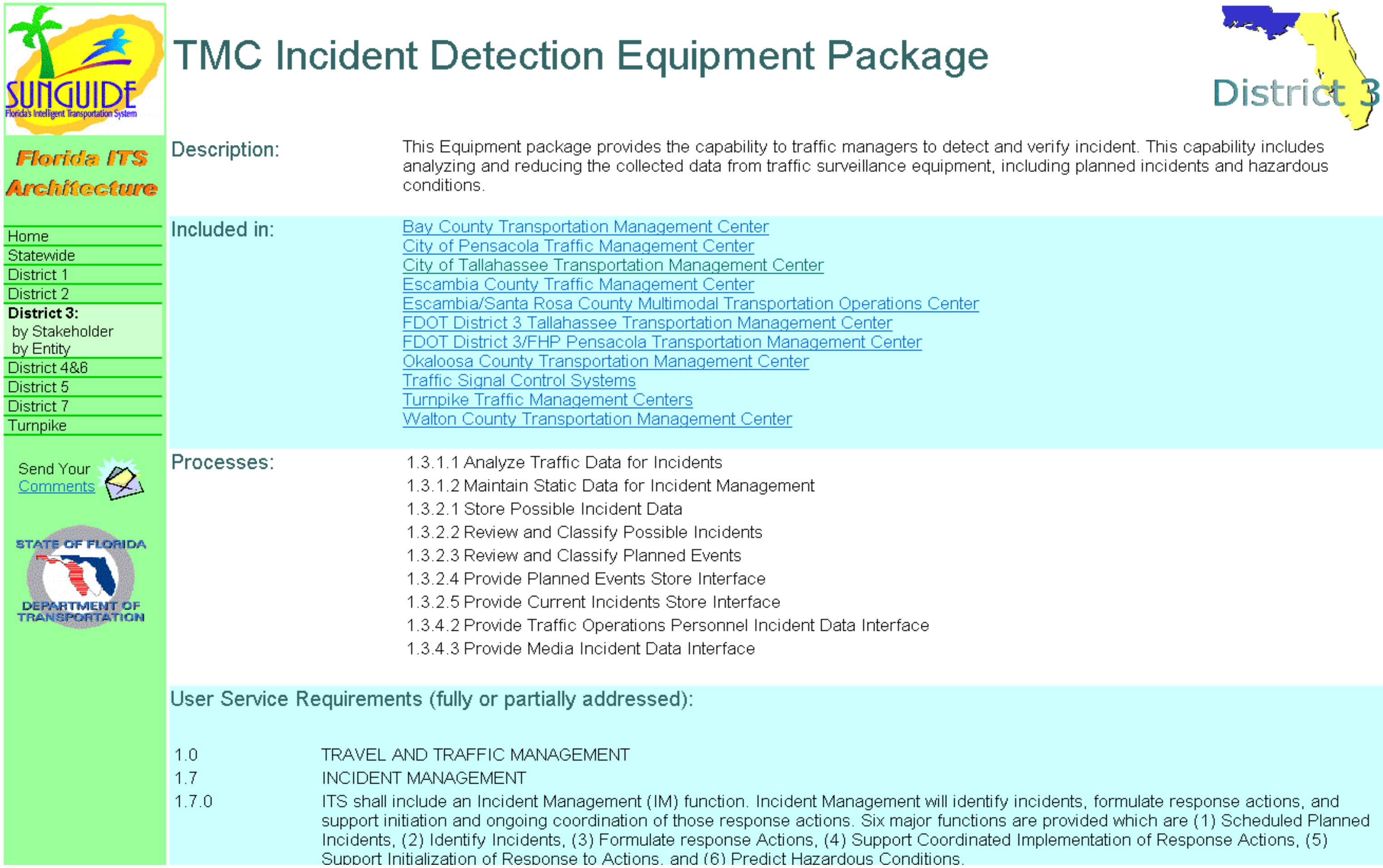
The final step in the design process is to create specifications for the project design. This step might entail a project procurement specification or a series of specifications for the key systems of the project, along with an overall project specification to cover the integration of the systems. The procurement or overall project specification can draw heavily from the requirements definition and the contribution to these requirements made by the statewide or regional ITS architecture. Specifications of individual systems, or even the overall project specification, may provide more detail than the requirements definition.

Florida's district ITS architectures and the *SITSA* have an additional level of detailed functional specification that can be accessed from the database or from the web-based version of the architectures. The following is an example of how to access the additional detail on the web-based version of the architecture.

Consider the following example from FDOT's District 3 regional ITS architecture. Suppose the objective of the project in question is to add incident management capabilities to the planned FDOT District 3 Tallahassee RTMC. As indicated under the section on requirements, selecting the Functionality Details link on the FDOT District 3 Tallahassee RTMC page will lead to a list of equipment packages that provide high-level definitions of the possible functions of the TMC. This was shown previously in Figure 4.2, as representing a level of detail appropriate for functional requirements. There is, however, an additional level of requirements detail contained in the web-based architecture view. Selecting the Details link on the TMC Incident Detection Equipment Package leads to the equipment package details page shown in Figure 4.4. In addition to the equipment package definition, the details page contains a listing of the set of process specifications (P-Specs) from the *NITSA* that may be applicable. These P-Specs represent the definition of ITS functions in the *NITSA*. As shown in Figure 4.4, the page gives a list of P-Spec titles. For the complete description of the functions, refer to the *NITSA* CD-ROM (currently Version 4.0) under Logical Architecture, or to the Logical Architecture Database, which is also contained on the CD-ROM. The details page also includes a list of user service requirements that may be applicable. These user service requirements are the functional requirements that were used to define the *NITSA* and can be found as well on the *NITSA* CD-ROM. The complete functional requirements are reproduced on the CD-ROM, so there is no need to go to the *NITSA* material to get the full information.

One final connection that can be made between the statewide or regional ITS architecture and the design step is in the identification of applicable ITS standards for the project. The regional ITS architecture contains a mapping of information flows to ITS standards. When the ITS project architecture is created as described above using the Turbo Architecture software, an output that is readily available from the tool is a set of applicable standards for the project. This set serves as a starting point for the specification of ITS standards as part of the project specifications (either in procurement specifications, system specifications, or possibly communications specifications). In addition to the identification of standards, the testing procedures should be considered as part of the project specification. These procedures are critical to the later validation steps of the overall systems engineering process. Documenting the set of applicable standards and testing procedures, along with the rationale for the standards selected for the project, will satisfy the *Rule 940* requirement for identification of applicable ITS standards and testing procedures.

Figure 4.4 – Example of the Detailed Requirements Information Selectable from each Equipment Package Web Page



SUNGUIDE Florida's Intelligent Transportation System

Florida ITS Architecture

- Home
- Statewide
- District 1
- District 2
- **District 3:**
  - by Stakeholder
  - by Entity
- District 4&6
- District 5
- District 7
- Turnpike

Send Your Comments

STATE OF FLORIDA DEPARTMENT OF TRANSPORTATION

TMC Incident Detection Equipment Package

District 3

Description:	This Equipment package provides the capability to traffic managers to detect and verify incident. This capability includes analyzing and reducing the collected data from traffic surveillance equipment, including planned incidents and hazardous conditions.
Included in:	Bay County Transportation Management Center City of Pensacola Traffic Management Center City of Tallahassee Transportation Management Center Escambia County Traffic Management Center Escambia/Santa Rosa County Multimodal Transportation Operations Center FDOT District 3 Tallahassee Transportation Management Center FDOT District 3/FHP Pensacola Transportation Management Center Okaloosa County Transportation Management Center Traffic Signal Control Systems Turnpike Traffic Management Centers Walton County Transportation Management Center
Processes:	1.3.1.1 Analyze Traffic Data for Incidents 1.3.1.2 Maintain Static Data for Incident Management 1.3.2.1 Store Possible Incident Data 1.3.2.2 Review and Classify Possible Incidents 1.3.2.3 Review and Classify Planned Events 1.3.2.4 Provide Planned Events Store Interface 1.3.2.5 Provide Current Incidents Store Interface 1.3.4.2 Provide Traffic Operations Personnel Incident Data Interface 1.3.4.3 Provide Media Incident Data Interface

User Service Requirements (fully or partially addressed):

1.0	TRAVEL AND TRAFFIC MANAGEMENT
1.7	INCIDENT MANAGEMENT
1.7.0	ITS shall include an Incident Management (IM) function. Incident Management will identify incidents, formulate response actions, and support initiation and ongoing coordination of those response actions. Six major functions are provided which are (1) Scheduled Planned Incidents, (2) Identify Incidents, (3) Formulate response Actions, (4) Support Coordinated Implementation of Response Actions, (5) Support Initialization of Response to Actions, and (6) Predict Hazardous Conditions.

4.2 Deployment of ITS Field Elements

4.2.1 Requirements Analysis

The analysis of requirements is used to develop functional and performance requirements for the system and its components. Requirements can and will change over the life cycle of the system. The systems engineering process is an iterative process and, through each iteration of the requirements analysis, the requirements will become more detailed.

- Often times, the stakeholder requirements are conflicting. Where possible, these requirements should be ranked from the most important to the least important.

- Establish and maintain a decision database as a means to maintain requirements traceability. This database will also become the baseline for documenting the requirements and specifications for the system and elements as part of the change management and configuration management processes.

- Develop a plan to analyze the requirements.[11] This plan may include the following tasks:
    - Define and quantify stakeholder expectations;
    - Identify and define constraints impacting design solutions [e.g., National Transportation Communications for ITS Protocol (NTCIP) standards' maturity, costs, and the capability of interfacing systems);
    - Identify external constraints (e.g., regulations and laws, legacy devices, etc.);
    - Identify operational scenarios;
    - Identify and define systems effectiveness measures;
    - Identify system boundaries [e.g., what falls outside the control of the system such as an arterial management system (AMS)];
    - Identify functional and physical interfaces to external or higher-level interacting systems;
    - Define environments for each operational scenario (e.g., weather, topology, time, road vibration, etc.);
    - Define functional requirements;
    - Define performance requirements;
    - Define modes of operation;
    - Define technical performance measures;
    - Define physical characteristics; and
    - Define human factors.

- The requirements and needs define the technology. Do not select an ITS technology and then define the requirements to meet the technology.

---

[11] *Systems Engineering Fundamentals*, Fort Belvoir, Virginia: Defense Acquisition University Press, 2000, p. 41-44. <http://clc.dau.mil/kc/no_login/portal.asp?strRedirect=LC_CIA>

- Well-written requirements are clear, complete, measurable, consistent, achievable, testable, verifiable, and in line with the expectations and needs of the stakeholders.

- Analyze the requirements for ambiguities, conflicts, and omissions so that there is a single, validated set of requirements.

- Ensure traceability for the requirements by numbering the components of every step:
  - Numbering should be hierarchical and sequential;
  - Reference the requirements both forwards and backwards within the processes so as the requirements become more technical and detailed through iterations of the systems engineering process, they can be mapped to the original stakeholder requirements; and
  - Depending on the complexity of the requirements, it may be helpful to use third party traceability software to support the process.

- Determine the ITS standards, in particular the NTCIP standards, that may apply to the field device. Develop a thorough understanding of the meaning and acceptance status or implementation stage of those standards. If staff resources do not include experts in these areas, hire a consultant that is an expert with the applicable standards.

  SpecWizard is a new software tool being developed through the FHWA to help users create accurate and unambiguous NTCIP specifications for ITS field elements. This tool should be available for distribution by the FHWA in the near future.

4.2.2 Analysis of Alternate System Configurations and Technologies

The selection of a system configuration or ITS field element technology should include a trade-off analysis between performance, costs, and the operational effectiveness of the element. Often, the simplest and/or off-the-shelf solutions provide the best results. The selection criteria should be weighted according to the preference of the customer. The selection criteria should be unbiased and the evaluation method documented.

- Identify alternative systems or technologies (e.g., by using techniques such as technology surveys, brainstorming, review of similar systems, and requests for information from vendors).

- Use trade studies (a formal decision-making methodology) to analyze alternative system configurations.

- Evaluate comparison costs and benefits using a utility-to-cost or benefit-to-cost analysis. Evaluate vendor responses based on the weighted selection criteria previously developed. Document analysis results.

4.2.3 Concept Designs and Master Plans

Document the concept of operations. This document should include the stakeholder goals and objectives, how the system or field elements will be deployed, practices and procedures, expectations, utilization environments, measures of effectiveness, and life cycles.

- The operational concept should be an accurate reflection of stakeholder needs.
- Use block diagrams in both text and graphical formats, if possible, to define and depict system concepts.
- Define and depict the ITS field element components and their relationships to the system.

4.2.4 Design Criteria Packages

The design specifications for the ITS field elements should be detailed and traceable to the requirements.

- The specifications are derived from the requirements.
- At least one specification should be defined for each requirement.

4.2.5 Determine the Method of Procurement

The field device subsystems can be procured by inclusion in the system project or can be procured separately. Low bid, task order, and design-build are successful procurement methods with field devices if no software or system integration is involved. Recommended considerations include:

- Agency procurement and contracting early in the process can maximize cost and time savings;
- ITS field equipment vendors and/or suppliers pre-qualification typically reduces procurement time and cost, plus assures quality products that meet specifications; and
- Pooled agencies and organizations that use indefinite quantity procurements usually obtain pricing breaks and quantity discounts for field equipment.

4.2.6 Design

The design process selects the components and their interconnections so as to meet the system requirements. The plans, specifications, and engineer's estimates are the results of a traditional transportation design project.

- During the design process, revisit the functional architecture and specifications to verify that the physical design can perform the requirements at the expected performance level.

- In terms of field elements, consider the following in the design:
    - Reliability;
    - Maintainability;
    - Availability; and
    - Affordability.

4.2.7 Verification

The verification process ensures that the system and elements have been designed and installed correctly. In other words, do the system and its elements meet the requirements? The process begins in the design activity.

- The derived requirements and design specifications must be testable. As noted in *Section 4.2.1, Requirements Analysis*, the requirements must be single statements that are unambiguous, understandable, and verifiable, and directly correspond to the specifications.

- Develop a systems test plan that will verify that the system and elements meet the requirements. Elements of the plan should include:
    - Review of documentation during implementation and integration;
    - Field inspection during implementation;
    - Factory tests;
    - Unit tests;
    - System tests; and
    - Acceptance tests.

4.2.8 Risk Management

Managing an installation requires good project management skills. These skills include planning the schedule, estimating the resource requirements (contractor and agency), and tracking the project costs and schedule.

A risk assessment should be performed to determine the likelihood of occurrence, the potential impact, and the mitigation costs for the identified risks. Some of the risks most frequently encountered are personnel shortfalls, unrealistic schedules and budgets, incorrect user interface(s), incorrect functions, scope creep, field device shortcomings, external dependencies (e.g., devices that are not available on time, subcontractor delays, etc.), and unrealistic technical requirements.

- Develop a work breakdown structure for the deployment contract.

- Develop a risk management plan. Define each risk, its likelihood of occurrence, and its potential impact. Define the mitigation methods and the monitoring and control measures to be taken if the risk occurs.

- For competitive procurements, define the evaluation process to be used in making a selection.

4.2.9 Installation and Integration

The implementation process should include intervals where field devices are tested on a limited basis to make sure that it is on the right track. It is better to catch problems early than to install all of the devices and then find out that they do not meet the requirements or perform as intended.

- Testing should be performed at the unit level for each field device (e.g., controllers, detectors, signs, etc.) to determine that all the equipment meets the functional and environmental tests.

- Unit level tests can be performed at the factory and in the field prior to device integration into the system or subsystem.

- Acceptance testing should be a formal, step-by-step process that tests the actual operation of the devices and should include:
  - Requirements testing by pre-defined test procedures and documentation of the test results in a test report;
  - Pass/Fail criteria definitions for each requirement;
  - Required observation periods for a predetermined amount of time to ensure continuous proper operation; and
  - During the observation period, the criteria for failures requiring a restart of the observation period should be clearly defined and included in the appropriate test report.

- Acceptance tests should be based on a matrix that is a function of the requirements, specifications, implementations, and the test procedures to ensure that all requirements are tested and passed.

4.2.10 Validation

The validation process determines whether the end product, or the component or field element, matches the needs of the stakeholders as defined in the concept of operations. The entire systems engineering process is an iterative process. If a determination is made during the validation process that the field element or its operation does not match the requirements, a review of the requirements should be initiated.

- Establish validation criteria of stakeholder needs and the operational concept at the outset of the design process.

- Validate the requirements by establishing that no new issues have been introduced and no issues are unresolved in the system as designed.

- Validation activities can include complex mathematical modeling and/or simulations, using CORSIM for analysis of the dynamic message signs (DMS) to show reduction in congestion, visual inspections (to verify the physical design features), demonstrations, or testing. The method should match the complexity of the subsystem or component and should consider the potential risks.

- The validation process should include the assurance that the design solution integrates properly with other systems or field elements.

- As part of the acceptance testing of the system and elements, an operational validation should be performed to confirm that the capabilities of the designed system and elements meet the operational needs.

4.2.11 Operations

Systems engineering processes can be used as tools for operations and maintenance activities to improve the systems and components. Systems engineering can be used to manage change, including design modifications to the ITS field elements and developing operator procedures and training for ITS services.

- Over time, the system should be reevaluated from the perspective of operations and maintenance and the needs and requirements redefined.

- Documentation should be maintained for each ITS field element, including as-built drawings, operations and maintenance manuals, and warranty and maintenance information.

- Procedures for operations, maintenance, and configuration management should be followed.

- Records for all modifications and maintenance activities should be maintained.

4.2.12 Program Management

In order to successfully operate and maintain the field devices, management of the program should ensure that adequate resources are available, including skilled and trained staff and an adequate budget. In addition, relationships with the stakeholders and participating agencies need to be maintained.

- If agency staffing is not available, outsourcing should be considered.
- Continue to maintain strong and effective working relationships with the stakeholders after the ITS project has been deployed.
- Conduct regular operational and maintenance briefings with agency personnel.
- Monitor, measure, evaluate, and report system performance and benefits.
- Establish performance requirements and criteria for managing and operating ITS services.
- Develop staff training programs.
- Provide necessary maintenance and operational resources.

5. Conclusions and Recommendations

5.1 Conclusions

Systems engineering is a powerful tool for improving the development, management, implementation, and operations of complex ITS projects. The systems engineering process model can prove to be an invaluable tool for controlling the costs, schedule, and performance of a complicated system. Due to the nature of the systems engineering processes that focus on the early definition of requirements, alignment with the goals and objectives of the FDOT and other stakeholders is virtually assured by simply following the *SEMP* once it is adopted. The systems engineering process model is a proven method for ensuring that projects deployed under its direction represent the most cost efficient and effective utilization of public resources and result in a fully integrated and coordinated system that appears seamless to its users and operators.

Currently, most of the FDOT ITS Office's district programs use some level of systems engineering principles, either formally or informally, in their project concept development efforts. However, projects that have progressed beyond the initial planning and concept or architecture development stage have been very limited in number. This fact, coupled with the relatively recent establishment of the Central Office's ITS Program, translates into a limited background for systems engineering process implementation and a limited basis for analysis of its capability. In spite of these limitations, the overwhelming evidence that embracing a systems engineering methodology can drastically improve an agency's efficiency is a solid foundation for continuing the work begun in this document and pursuing the development and completion of a comprehensive *SEMP*.

Other benefits of adopting a systems engineering methodology include improved systems reliability during and after the deployment stage of the project and the enhanced ability to identify and solve problems and address issues when they do occur as a result of the activities of the configuration management process. The meticulous documentation of the project development, from concept exploration through requirements definition and implementation and especially the verification and validation steps, provides an excellent "roadmap" for understanding and solving system and integration problems.

In addition to these advantages, the requirements of the FHWA's *Rule 940* are satisfied as a result of following the *SEMP*. The focus on the early definition of user requirements achieved by following a formal process allows the project manager to:

- Explore and explicitly state the needs of the stakeholders in terms of functionality (i.e., define *what* is needed);
- Transform user needs into design requirements and specifications (i.e., define *how* to meet the needs);
- Create documentation for all steps of the process thereby providing a basis for all decisions, changes, and future modifications, additions, and retirement/disposal issues;

- Provide documented evidence and ensure that all performance, functional, and user requirements are properly defined and properly addressed; and

- Assure that all possible and practical alternative configurations are explored and the best candidate selected throughout the project life cycle steps (including technology options, procurement options, and the technical merits and cost versus relative value of each).

As well as the above, the relevant portions of the regional architectures being implemented and any testing procedures and ITS standards being implemented must be identified for each project.

Besides meeting the requirements of *Rule 940*, other benefits of using systems engineering can be realized. The process itself is a tool for improvement. By following the processes for the Management and Environmental Categories of the *EIA/IS 732* model, the processes themselves and the staff responsible for implementing them become a "tighter ship" for systems engineering, simply by virtue of the nature of the processes. In short, *systems engineering is a self-improving process.*

Furthermore, each iteration of the process serves to ensure that the previous one was done properly. In other words, the process is self-checking. With advantages such as these, it is evident that implementing systems engineering is a positive step towards improved efficiency for all participants. In other words, *systems engineering is a self-checking process.*

5.2 Recommendations

Systems engineering can be applied to large complex problems and projects as well as small ones. For ITS projects, these processes can be applied not only to high-level concept planning such as the development of architectures, but also to detailed design such as the implementation of field devices. Once properly customized, the systems engineering method can be used for all individual deployment projects and as a tool to manage the overall program more effectively. In order to ensure proper implementation of the systems engineering processes, a methodology based on a model should be selected and adopted. The model is the key element that formalizes the closing of the gap between what is needed and how those needs will be satisfied.

Based on the results of the analysis of the current status of the FDOT ITS Program's capability maturity for systems engineering, and to the extent that the analysis was modeled after the *EIA/IS 732-2, SECM Appraisal Method*, an initial level of systems engineering capability currently exists in the ITS Program. In order to expand and improve this level of capability maturity, the ITS Program should proceed with the development of a comprehensive *SEMP* using the results of this document as a starting point and guideline. The elements of the Technical Category, as shown in this document, already exist in the ITS Program and should be planned to be brought up to a higher level, while the elements in the Management and Environmental Categories should be planned to be implemented to at least an initial level of capability.

In order to accomplish these objectives, the leaders of the FDOT ITS Program should reach a consensus on the basic model and methodology to be employed by establishing a *SEMP* Committee to guide and oversee the development of the plan. The overall goals of the *SEMP* development effort, along with any interim goals for reaching full-scale implementation, should be decided, agreed to by all the committee members, and documented for future reference. This effort might include drafting a policy statement on the need for and desire to develop consistent systems engineering approaches to ITS developments in Florida. This policy statement should be general in nature and refer to the *SEMP* itself for specific implementation guidance.

Once the model and methodology have been selected, the practices to accomplish the objectives should be determined. This work would entail identifying the specific activities that will lead to or assist in improving systems engineering capability and specify the details of implementing the process activities. Ideally, the *SEMP* will recommend methods and tools for a project manager to use in developing ITS services in Florida. These activities are the heart of the SEMP and will be used to guide the development of ITS projects from the conceptual stage through acceptance, operations, and management. In addition to the project management process activities, the specific activities related to program management and environmental improvement will be defined and documented in detail. Specifically, the *SEMP* will:

- Define the sub-processes of the systems engineering approach including inputs and outputs;
- Define criteria for the successful completion of each major activity in the systems engineering approach;
- Define a set of standard methods and technical tools for use in the systems engineering approach;
- Define standard scopes of work, work effort guidelines, and schedules for critical activities;
- Establish a formal process for implementing and improving systems engineering functions; and

- Define roles and responsibilities for various stakeholders.

As can easily be seen from the above information, the activities involved in the development of the *SEMP* closely correspond to the steps outlined in the Vee diagram, shown in Figure 2.6. In fact, this model will be used as the approach for the development of the *SEMP*. The conceptual definition phase will become the concept of *SEMP* implementation and utilization. The requirements definition phase relates to the documentation of stakeholders' roles and responsibilities in terms of functionality and success criteria. The design phase involves specifying the "nuts and bolts" of how the requirements identified in the previous step will be satisfied for various types of ITS projects, including, but not limited to, ITS software, ITS telecommunications, IMS services, ATIS, ATMS, and ITS operations and maintenance. The implementation phase entails the production of the *SEMP* itself, while employing the "cross-cutting" activities for validation, verification, and configuration management purposes. The operations and management phases equate to the definition, selection, and implementation of the practices related to process and environmental improvement (i.e., the Management and Environmental Categories of *EIA/IS 732*), to provide continuous enhancement of the processes themselves and of the *SEMP* as a whole.

The means to standardize the identified practices should be decided and documented so that all future ITS project developers and managers can access and reference them in the scopes of work or technical special provisions for project implementation. If the determination is made that no appropriate means for standardizing systems engineering practices exist currently, then the *SEMP* Committee should explore the possibility of developing new manuals or procedures.

The importance of selecting the proper venue for standardizing systems engineering processes cannot be over-emphasized. Although some project development material currently exists within standard FDOT documents, such as the Project Development and Environmental (PD&E) Manual and the Plans Preparation Manual Design volumes, very little is specifically related to ITS projects and no formal manual utilizing a systems engineering methodology exists. The fact that ITS projects typically involve new or unproven technologies and software and rely heavily on communications networking technology makes them quite unique and very different from traditional FDOT projects. Therefore, taking a new and unprecedented approach to their development is absolutely necessary. The *SEMP* Committee should carefully consider this issue and determine whether a stand-alone ITS project development manual, based on the principles of systems engineering, might be the best solution.