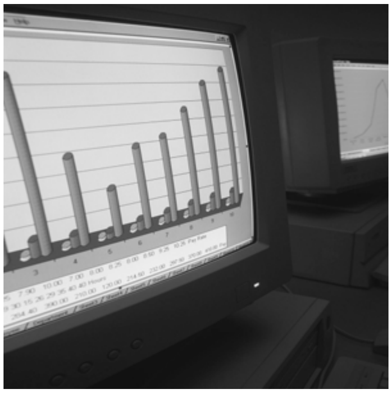


Testing Programs for Transportation Management Systems

A Technical Handbook

February 2007

U.S. Department of Transportation
Federal Highway Administration

Notice

This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange. The United States Government assumes no liability for its contents or use thereof. This report does not constitute a standard, specification, or regulation.

The United States Government does not endorse products or manufacturers. Trade and manufacturers' names appear in this report only because they are considered essential to the object of the document.

1. Report No. FHWA-HOP-07-088	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle Testing Programs for Transportation Management Systems: A Technical Handbook		5. Report Date February 2007	
		6. Performing Organization Code	
7. Author(s) Mr. Robert Rausch, P.E., TransCore		8. Performing Organization Report No.	
9. Performing Organization Name and Address TransCore 192 Technology Parkway, Suite 500 Norcross, GA 30092		10. Work Unit No. (TRAIS)	
		11. Contract or Grant No. DTFH61-01-C-00180	
12. Sponsoring Agency Name and Address Operations Office of Transportation Management Federal Highway Administration 400 Seventh Street, SW Washington, DC 20590		13. Type of Report and Period Covered Final Report July 2006 – February 2007	
		14. Sponsoring Agency Code	
15. Supplementary Notes FHWA Task Order Manager: Tom Stout (HOTM)			
16. Abstract Testing Plans for Transportation Management Systems: A Technical Handbook to provide direction, guidance, and recommended practices for test planning, test procedures, and test execution for the acquisition, operation, and maintenance of transportation management systems and ITS devices.			
17. Key Word test, testing, acceptance, verification, test plan, installation, transportation management system, intelligent transportation system	18. Distribution Statement No restrictions		
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of Pages 170	22. Price

This page intentionally left blank.

Table of Contents

1. Introduction..1
1.1 Background..1
1.2 Objectives..2
1.3 Document Structure..2
2. Testing: A System Engineering Life Cycle Process..4
2.1 Overview..4
2.2 The System Life Cycle..4
2.3 Test Planning..5
2.4 Summary..14
3. The Role of Testing in the Project Life Cycle..15
3.1 Overview..15
3.2 National ITS Architecture..15
3.3 Transportation Management Systems Procurement..21
3.4 Project Management, Staffing and Training..25
3.5 Documentation..26
3.6 Configuration Management..27
3.7 Summary..29
4. An Introduction to Testing..30
4.1 Overview..30
4.2 Test Methods..30
4.3 Test Approach..31
4.4 Test Types..33
4.5 Test Identification and Requirements Traceability..37
4.6 Test Descriptions..37
4.7 Test Requirements and Resources..37
4.8 Test Execution..40
4.9 Summary..43
5. Planning a Project Test Program..44
5.1 Overview..44
5.2 A Building Block Approach..44
5.3 The Product Maturity Concept..48
5.4 Risk..51
5.5 Summary..51
6. Hardware Testing..52
6.1 Overview..52
6.2 What Types of Testing Should Be Considered?..52
6.3 When Should Testing Occur?..68
6.4 Hardware Test Phases..71
6.5 Other Considerations for the Hardware Test Program..80
6.6 Summary..85
7. Software Testing..87
7.1 Overview..87
7.2 What Types of Testing Should Be Considered?..88

7.3 When Should Testing Occur? 103
7.4 Software Test Phases 104
7.5 Other Considerations for a Software Test Program 106
7.6 Summary 107

8. System Level Testing 109
8.1 Overview 109
8.2 Subsystem Testing 109
8.3 Systems Testing 110
8.4 Summary 111

9. Other Testing Considerations 112
9.1 Overview 112
9.2 Meaning of Shall, Will, May, Should and Must in Requirements Statements 112
9.3 How To Write Testable Requirements – Do’s and Don’ts 113
9.4 Test Pass/Fail Criteria 114
9.5 Test Reporting 114
9.6 Test Failures and Re-Testing 115
9.7 Testing Timeframes 116
9.8 Testing Organization Independence from Developers 117
9.9 Testing Relevancy and Challenges 118
9.10 Issues Affecting System Reliability 119
9.11 Testing Myths 126
9.12 Testing Tradeoffs 127
9.13 Estimating Testing Costs 129
9.14 Summary 130

10. Additional Resources 131

References 134

Appendix A. Example Verification Cross Reference Matrix 135

Appendix B. Sample Test Procedure 142

Appendix C. Sample System Problem / Change Request (SPCR) Form 153

Appendix D. Example Application of NTCIP Standards 158

List of Figures

Figure 2-1. Systems Engineering “V” Model....5
Figure 3-1. National ITS Architecture Components....17
Figure 3-2. National ITS Architecture Interconnect Diagram....19
Figure 3-3. System Document Tree....27
Figure 5-1. Building Block Approach to Testing....47
Figure 6-1 NEMA Temperature Profile....58
Figure 6-2. NEMA Relative Humidity Profile....59

List of Tables

Table 2-1. Requirements Sources....6
Table 4-1. Test Levels and Example Test Responsibilities....32
Table 7-1. NTCIP-Related Web Sites....92
Table 9-1. Common Failure Causes and Mitigating Factors....119
Table 9-2. Failure Modes and Effects....121

Acronyms and Abbreviations

AASHTO	American Association of State Highway and Transportation Officials
AC	Alternating current
ANSI	American National Standards Institute
ATC	Advanced Traffic Controller
ATIS	Advanced Traffic Information System
AWS	American Welding Society
C2F	Center-to-Field
CALTRANS	California Department of Transportation
CCB	Configuration Control Board
CCTV	Closed Circuit Television
CDR	Critical Design Review
CDRL	Contract Deliverable Requirements List
CI	Configuration Item
CM	Configuration Management
CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
COTS	Commercial-Off-The-Shelf
CSC	Computer Software Component
CSCI	Computer Software Configuration Item
DAT	Design Acceptance Test
DC	Direct Current
DMS	Dynamic Message Sign
DSRC	Direct Short Range Communications
DUT	Device Under Test
FAT	Factory Acceptance Test
FDS	Field Device Simulator
FDOT	Florida Department of Transportation
FHWA	Federal Highway Administration
GIS	Geographical Information System
GUI	Graphics User Interface
HRS	Hardware Requirements Specification
HWC	Hardware Component
HWCI	Hardware Configuration Item
ICD	Interface Control Document
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
ISTEA	Intermodal Surface Transportation Efficiency Act
ITE	Institute of Traffic Engineers
ITS	Intelligent Transportation System
LCD	Liquid Crystal Display
MIB	Management Information Base
MIL	Military
MS/ETMCC	Message Set for External TMC Communications
MTBF	Mean Time Between Failures

MTTF	Mean Time To Failure
MTTR	Mean Time To Restore
NEMA	National Electrical Manufacturers Association
NMS	Network Management System
NTCIP	National Transportation Communications for ITS Protocol
NY	New York (State)
OTDR	Optical Time Domain Reflectometer
QA	Quality Assurance
QPL	Qualified Products List
PC	Personal Computer, Printed Circuit (Board)
PDR	Preliminary Design Review
R&D	Research and Development
RDBM	Relational Data Base Management
RF	Radio Frequency
RFC	Request for Comment
SDF	Software Development Folder
SEP	Systems Engineering Process
SFMP	Simple Fixed Message Protocol
SNMP	Simple Network Management Protocol
SPCR	System Problem/Change Request (Form)
SRS	Software Requirements Specification
STMP	Simple Transportation Management Protocol
TEA	Transportation Equity Act
TEES	Traffic Engineering Electrical Specification
TERL	Traffic Engineering Research Laboratory
TCIP	Transit Communication Interface Protocol
TMC	Traffic Management Center
TMDD	Transportation Management Data Dictionary
TMS	Transportation Management System
TTP	Technical Test Procedure
USDOT	United States Department of Transportation
V&V	Verification and Validation
VCRM	Verification Cross Reference Matrix

1. Introduction

1.1 Background

Transportation Management Systems (TMS) are composed of a complex, integrated blend of hardware, software, processes, and people performing a range of functions. These functions typically include data acquisition, command and control, data processing and analysis, and communications.

Developing, implementing, operating, and maintaining a TMS is a challenging process in many areas and for a variety of reasons. It takes many different skill sets to deploy a TMS ranging from program management (finance, scheduling, human resources, etc.) to specialized software and hardware skills (operating system device drivers, communications protocol troubleshooting, electrical/electronic engineering, etc.).

Testing is an important part of the deployment of a TMS. The purpose of testing is two-fold. First, testing is about verifying that **what was specified is what was delivered**: it verifies that the product (system) meets the functional, performance, design, and implementation requirements identified in the procurement specifications. Hence, **a good testing program requires well-written requirements** for both the components and the overall system. Without testable requirements, there is no basis for a test program.

Second, testing is about managing risk for both the acquiring agency and the system's vendor/developer/integrator. The test program that evolves from the overarching systems engineering process, if properly structured and administered, facilitates the process of managing the programmatic and technical risks and helps to assure the success of the project. The testing program is used to identify the point at which the work has been "completed" so that the contract can be closed, the vendor paid, and the system shifted by the agency into the warranty and maintenance phase of the project. Incremental testing is often tied to intermediate milestones and allows the agency to start using the system for the benefit of the public.

Consider the risks and results described in the following two systems:

- The first system required 2700 new, custom designed, field communication units to allow the central traffic control system to communicate with existing traffic signal controllers. The risk associated with the communication units was very high because once deployed, any subsequent changes would be extremely expensive to deploy in terms of both time and money. As in many TMS contracts, the vendor supplied the acceptance test procedure, the specifications defined the functional and performance requirements that could be tested, and the contract terms and conditions required the test procedure verify <u>all</u> of the functional and performance characteristics. The execution of a rigorous test program in conjunction with well-defined contract terms and conditions for the testing led to a successful system that continues to provide reliable operation 15 years later.

On the other hand, deploying systems without adequate requirements can lead to disappointment.

- In the second case, the agency needed to deploy a solution *quickly.* The requirements were not well defined and, consequently, the installed system met some but not all the needs of the

> acquiring agency. Since the requirements were not well defined, there was no concise way to measure completion. As a result, the project was ultimately terminated with dissatisfaction on both the part of the agency and the integrator. Ultimately, the agency decided to replace the system with a new, custom application. However, for this replacement system, the agency adopted a more rigorous system engineering process, performed an analysis of its business practices, and produced a set of testable requirements. The replacement system was more expensive and took longer to construct, but, when completed, it addressed the functional requirements that evolved from the review of the agency's business practices. Testing assured that the second system met the requirements and resulted in a minimal number of surprises.

In both these cases, testing made a critical difference to the ultimate success of the programs.

1.2 Objectives

The objective of this handbook is to provide direction, guidance, and recommended practices for test planning, test procedures, and test execution for the acquisition, operation, and maintenance of transportation management systems and Intelligent Transportation Systems (ITS) devices.

This handbook is intended for individuals that are responsible for or involved in the planning, design, implementation, operation, maintenance, and evaluation of the TMS for public agencies. Targeted end users of the handbook are first-level supervisors (managers and supervisors) and technical staff that may include transportation planners, traffic engineers and technicians, construction and maintenance engineers, and traffic management center (TMC) staff.

The guide is an introduction to the technical discipline of testing and provides a foundation for understanding the role, terminology, and technical issues that are encountered while managing a test program. Because it is an introductory guide, other testing resources are referenced for additional information.

The guidance provided herein is best utilized **early** in the project development cycle, prior to preparing project and acquisition plans. Many important decisions are made early in the system engineering process that affect the testing program. The attention to detail when writing and reviewing the requirements or developing the plans and budgets for testing will see the greatest pay-off (or problems) as the project nears completion. Remember that a good testing program is a tool for both the agency and the integrator/supplier; it typically identifies the end of the development phase of the project, establishes the criteria for project acceptance, and establishes the start of the warranty period.

1.3 Document Structure

This handbook provides an introductory guide to transportation management system (TMS) testing. It begins by discussing testing within the system engineering life-cycle process and stages in Chapter 2. This discussion introduces the system engineering process and identifies the sources of requirements that are the basis for testing and ultimate system acceptance. Chapter 3 provides an overview of the TMS acquisition process starting with the development of a regional architecture for the TMS. It then discusses system procurement considerations and practices with emphasis on how the test program affects various phases of the system's life cycle, including post-acceptance operation and maintenance. The material on the testing role in the system engineering process and the project life cycle is necessary to set the stage for testing and to put it into context.

Chapter 4 addresses the basics of testing. It discusses testing methods, planning, test development, resources, and execution. This chapter tries to fold the entire technical discipline into a compact presentation. It will not make you an expert but it will introduce you to the terminology and many of the concepts that will be used by the system vendors. Chapter 5 focuses on planning a project test program. The broad-based technical discussion of testing fundamentals provides material that a TMS project manager must be familiar with to implement the project's testing program and fit it into the overall TMS deployment program. The success of the project will depend on the test program that evolves from this system engineering process and, when properly structured and administered, the test program allows the programmatic and technical risks to be managed. Understanding the role of testing, its terminology, and technical issues is key to managing a test program for a TMS deployment project. This material also allows the project manager to discuss the program in detail with the technical experts that may be employed to assist with development and implementation.

Chapters 6 and 7 discuss hardware and software testing respectively and provide some real-world examples. Chapter 8 addresses testing at the subsystem and system levels. Chapter 9 provides guidance and recommendations on such topics as the meaning of "shall" and "will" in requirements statements, how to write testable requirements, and pass/fail criteria. This chapter also includes helpful information and suggestions on test reporting, testing timeframes, testing organization independence, testing relevancy and challenges, failure modes and effects, testing myths, and estimating test costs. Chapters 6-9 are much more pragmatic discussions of testing, presenting lessons learned and providing practical guidance for planning, developing, and conducting testing.

Chapter 10 provides a list of available resources for further investigation. These are web sites that address TMS-relevant standards, organizations and associations that provide training in the testing discipline, and organizations involved with the process of establishing standards for ITS. These resources are provided as starting points and may appear in the results of web searches regarding testing. The handbook includes four appendices: Appendix A – Example Verification Cross Reference Matrix, Appendix B – Sample Test Procedure, Appendix C – Sample System Problem/Change Request Form, and Appendix D – Example Application of NTCIP Standards.

Throughout the document, the term "vendor" refers to the supplier, developer, or integrator of the system.

2. Testing: A System Engineering Life Cycle Process

2.1 Overview

This chapter discusses testing within the system life cycle and the system engineering process. The focus is on the information needed to produce a testing program that fits the needs of the acquiring agency and the complexity of the system.

The system engineering process describes the application of technology and management practices to building complex systems of equipment, software, people, and procedures to satisfy specific needs. The process is tailored to the needs being satisfied and the complexity of the envisioned system. In some cases, the needs may be simple, well defined, and satisfied by modifying an existing system. At the other end of the spectrum are cases where the needs are extensive, broad, and diverse, resulting in a new, complex system that may have unanticipated impacts and relationships. The *system engineering process* provides a process framework to address problems on both ends of the spectrum.

As the quantity of software-intensive systems has exploded over the past 60 years, many excellent references for the system engineering process have been written. Information on system engineering can be obtained from resources like the FHWA's *Building Quality Intelligent Transportation Systems Through Systems Engineering* and from other resource organizations listed in the last chapter of this guide. This document focuses on the testing process and its role within the systems engineering process.

The reader is advised that not all of the items described here will be applicable to every project, as the size and complexity of the project will determine what is applicable. In addition, unless the acquiring agency has extensive technical expertise and experience in systems engineering, it should anticipate utilizing outside expertise. This should not dissuade the agency from augmenting its staff or developing these skills in-house, but one should not begin a complex TMS project without expert assistance and an in-depth understanding of what is involved in the system engineering life-cycle process.

2.2 The System Life Cycle

The *systems engineering process* (SEP) is a methodology and tool for managing a system's life cycle starting with concepts and ending with the system's retirement. It is a highly structured method to facilitate the development, maintenance, refinement, and retirement of dynamic, large-scale systems consisting of both technical components (Figure 2-1 is the Systems Engineering "V" Model for ITS that details the various stages that occur within the system's life cycle.

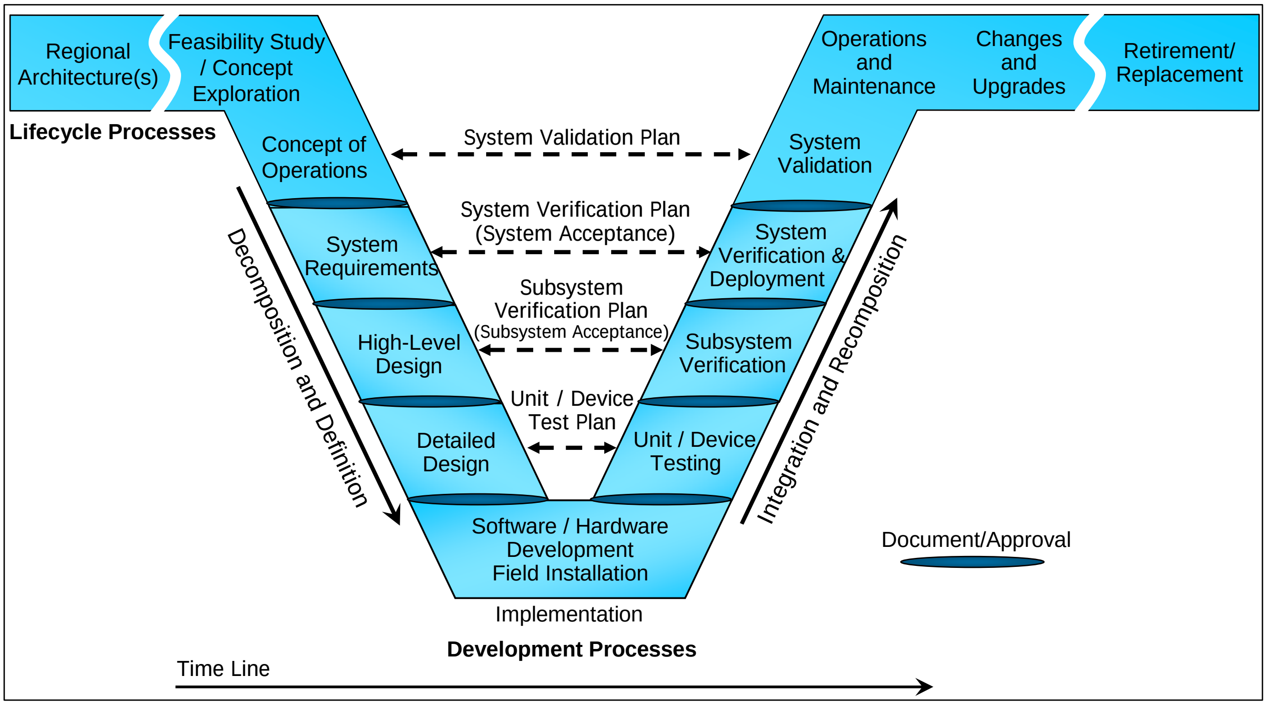


Figure 2-1. Systems Engineering "V" Model

While testing is shown as **one** stage of the life cycle, it is important to understand that testing is also a **continuous process** within the life cycle. Testing begins with writing the requirements; each requirement must be written in a manner that allows it to be tested. During the design stages, testing will be a consideration as design trade-offs are evaluated for their ability to satisfy the requirements. New requirements may emerge from the designs as choices are made to satisfy the requirements within the project's constraints. Hardware components, software components, subsystems, and systems will be verified during the implementation and testing stages. Final system-level tests will be performed to accept the system and demonstrate the system's readiness for production service. However, testing activities will not end once the system is in operation; it will continue as the operations and maintenance staff perform corrective, adaptive, and other system maintenance activities.

The following sections will discuss these process activities in more detail with respect to the input information, selected processes, and the results from the testing process while verifying the transportation management system.

2.3 Test Planning

This section provides an overview of test planning starting with the information that goes into developing the test program. It then summarizes testing activities within the system engineering process and describes the products produced by these activities. Emphasis is placed on the system requirements that will serve as the baseline for the development of the overall test plan for the completed system.

2.3.1 Inputs

Testing starts with well-written requirements. From the testing perspective, the system specification document must be written with “testable” requirements. Without testable requirements, the testing budget cannot be used efficiently to ensure that the system performs according to its design. The testing process verifies that each requirement has been met; if it is not testable” there are no grounds for acceptance of the work. Table 2-1 lists various documents that will contain requirements for the system.

It is important to realize the requirements will be repeated in the various documents as higher level requirements are carried down into more detailed design documents.

Table 2-1. Requirements Sources

Documents that Will Contain Requirements for the System	
• Contract Documents	• Hardware Requirements Specification (HRS)
• Concept of Operations	• Software Requirements Specification (SRS)
• System Definition	• Interface Control Documents (ICD)
• System Design Report (SDR)	• Product Data Sheets

Once the requirements have been documented and approved (i.e., accepted by the project stakeholders and the project management), the process of requirements analysis can begin. In this phase, the requirements are analyzed and decomposed by type (e.g., operational, functional, interface, physical, electrical, etc.) and then allocated to hardware or software subsystems for implementation.

Operational requirements typically define and describe how the system operates and what features are visible to system operators and other users. These requirements are usually derived directly from the concept of operation document and are at a high level.

Functional requirements describe and specify the underlying functionality behind the system's operational features, specific capabilities to be provided by designated subsystems, user interactions, equipment types and quantities to be interfaced, and other system characteristics. These are the critical functional requirements. Less critical, lower level functional requirements will be identified and derived during the hardware and software design phase. Note that it is important that performance requirements (for the system) be clearly identified – if they matter. For example, the screen size, refresh rates, and screen-to-screen navigation times may be critical for a graphical user interface. These must be quantified to be testable; simple statements such as “easy to use” or “refresh quickly” provide little if any measurable guidance for either the designer or the test developer.

Interface requirements detail the data exchange between the hardware and software subsystems and with other external systems. Here the data types, formats, message structure, transfer methods, and protocols are defined to the level of detail necessary to begin hardware and software design.

Interface requirements between this system and other external systems are captured in interface control documents (ICD), which are agreements between two or more parties that detail the physical, functional, and electrical interface requirements of the data exchanges and the data handling responsibilities of each party to the agreement.

Once again, it is important to include performance requirements (data rates, connect time, retry times, etc.) in the interface requirements document. It is also important to identify the requirements for “anomalies”; i.e., how to handle problems such as lost packets, poor throughput, intermittent connections, and corrupted data.

As the system engineering process moves forward, new requirements will emerge. They may come from the resolution of development issues, or just be a great idea that came to mind while personnel discuss the merits of the system’s operation. Regardless of how they are identified, the important point is to have a process in place to recognize the new requirements formally and roll them into the existing requirements. This process is incorporated into the *configuration management plan* that defines the approved process for the development, review, and approval of changes to the system and any new or modified functional requirements to be implemented. The configuration management plan also defines how those changes are documented, tested, and accepted.

It is important to recognize that new requirements will require additions to the test procedures and may affect the test program already developed. These are the natural consequence of change; it is important that such changes be managed and that impacts on the testing program be assessed when considering the change because the cost of the change may affect many other aspects of the project. (Note that change is inevitable; one probably does not want to prevent changes, but a plan to manage change as it occurs is important.)

2.3.2 Test Plan Preparation

Test planning is a key element in the acquisition and deployment of a TMS. The acquiring agency must become involved in this aspect of the test program to ensure that the testing program truly reflects the agency’s understanding of the requirements. Where the contractor or vendor is assigned the responsibility and authority to develop the test plan, the test plan and test procedures are often carefully crafted so that few tests fail. The system or device under test is often not stressed or tested at the boundaries (max and min conditions), to determine what breaks so it can be fixed. This does not mean that the agency must develop the test plan itself, but it does mean that the agency must carefully review the test plan and test procedures prior to approval to ensure that meaningful testing is performed. The agency should seek professional assistance in this area if it is not comfortable with the risk of performing this task in-house. After all, the agency wants to be assured that the system will operate under all conditions, not just ideal conditions, and that the testing verifies all requirements (where practical), not just the vendor-selected requirements.

The system specification developed during the requirements definition phase establishes the requirements baseline for the development, implementation, qualification testing, and acceptance of the system. The system test plan defines and documents a test program that assures all of the elements of the system are subjected to a structured review process specifically directed at verifying compliance with the physical, functional, and operational requirements at each level of system development and deployment.

Successful testing at one level is a prerequisite for testing at the next higher level so that the determination of compliance is cumulative and leads to successful demonstration of system compliance. The system test plan covers all levels of testing for both hardware and software and defines the test responsibilities of both the providers and installers of the system.

2.3.3 Testing Activities

Testing activities are tasks that are specific to the project's current life cycle stage and processes that span the entire life cycle. For example, unit testing activities occur during the implementation stage while management of the Verification Cross Reference Matrix (a.k.a. Traceability Matrix) spans many stages of the life cycle and would occur in parallel to unit testing. This section highlights the many testing activities that occur during each system life-cycle stage and those that span stages.

2.3.3.1. Requirements

The testing staff should become involved during the requirements development phase of the project. The testing staff's primary goal during the requirements development is to ensure that each requirement can be tested. The verification cross reference matrix (VCRM) is written during this stage and it becomes a key document for the project to be maintained under configuration management along with the requirements documents.

2.3.3.2. Design

Design reviews are another area requiring a significant participation of the testing staff. As the design progresses, the test team tracks the allocation of requirements to the configuration items presented in the high-level design documents. A preliminary design is generated to define the architecture of the system and provides for a high-level distribution of the requirements to the system components. A detailed design is then prepared to provide a complete description of the hardware and software design by detailing the functions, behavior and interfaces for all hardware components and computer software components within their respective configuration items. Each hardware and computer software component defined is traceable back to the requirements allocated to a specific configuration item from the respective hardware requirements specification or software requirements specification. The detailed design documents describe the hardware and software designs at a level suitable for the start of prototyping and unit testing.

While it is not necessary for the acquiring agency to attend all of these intermediate reviews, they do afford the agency an opportunity to become familiar with the system architecture, components, and interfaces being developed and they occur so that the agency can redirect the design if necessary. Agency participation also sets the stage for understanding the testability of the design and how the test procedures will verify that the implementation will meet the requirements.

A critical design review (CDR) is scheduled when the detailed design documents are completed. The review is typically conducted by the hardware or software development team leader and should include program management, systems engineering, integration and test, and operations staff representatives. Due to its critical nature, this review should be attended by the acquiring agency, and agency attendance <u>should be codified as a contractual requirement</u>. The CDR is the last opportunity to review the hardware and software designs and identify deficiencies before they become errors and omissions that present themselves in subsequent testing phases. It cannot be stressed enough that identifying and resolving the deficiencies at this time is economically vital to the successful financial management of the project.

It is important that the agency employ the resources necessary to feel confident during the CDR and be prepared to "sign-off" on the results. The agency must be able to relate the detailed design to the requirements; the agency needs to recognize that the frame of reference of the designer/implementer (often the programmer) is very different, and “assumptions” which may seem obvious to the agency must be concisely documented to avoid omission or misinterpretation by the designer/implementer. This is another instance where the development of the test procedure should be stressed early in the project. The test procedures will be based on the requirements and represent the agency’s view of the final expected operation.

From the test program perspective, the CDR and approval of the detailed design represents a significant milestone. From this point forward in the project, testing will be performed on actual hardware and software. It is also important to be able to trace all of the requirements to an element of the design; this is an opportunity for both the developer and the agency to ensure that all requirements have been included in the detailed design and that the test procedures are being developed to verify the original requirements.

2.3.3.3. Implementation and Unit Testing

During the implementation phase of the project, many hardware and software components are built and the unit-testing process begins. The acquiring agency must decide the appropriate level of unit testing to fund (i.e., the costs for participation of agency personnel or experts hired by the agency, travel expenses, equipment, etc.). The agency may leave the unit-testing program to the vendor, may require reviews of the vendor’s unit testing documentation, or may decide to participate in the unit testing. These choices are discussed in detail in later chapters but, for now, the emphasis is on the start of the test program.

The steps to implementing the hardware design are relatively straight forward; there may be a prototype development, which is evaluated and tested based on the design and construction requirements identified in the procurement specifications. This phase is generally followed by the development of production units, which will be the devices used by the project.

Fabrication of the productions units is next. This activity should be monitored or tested to be sure it is consistent with the design and construction standards in the procurement specifications. Once the production units are completed, the agency generally conducts detailed unit testing of the device(s). Embedded firmware (i.e., software that is an integral part of the hardware component and typically contained on a computer chip within the device) that represents a new or modified design should be treated like software and subjected to the same rigorous development and test program as new or modified software. Hardware components are combined and integrated into deliverable hardware configuration items that are defined in the detailed design.

Creation of the software source code also begins with the acceptance of the CDR. This can be an iterative process as the design matures, particularly where performance issues are key elements of the design and where prototyping of graphical user interface (GUI) screens is necessary. Coding, including code modifications to existing baseline source code, are accomplished in accordance with the software development plan and software design and coding style standards specified in the procurement specifications for the target computer platform(s). Software components are combined and integrated into deliverable computer software configuration items that are defined in the detailed design.

Build is the process of combining code units and data structures into executable components for the purposes of testing their interfaces with each other and with system resources. This is a necessary step in developing an operational version of the software system. A software ***build*** typically consists of multiple computer software configuration items and is loaded and tested together on the development computer system.

Subsequent to successful unit testing and integration in the manufacturing environment, factory acceptance testing can begin for those components that are not standard products, on a qualified products list, or require a unique test environment that cannot be easily achieved at the ultimate installation site (e.g., dynamic message signs). Components delivered to the site for installation should be subjected to receiving inspections and functional testing as detailed in the procurement specification. Conditional acceptance (and partial payment) can be given with respect to delivered quantities, but final acceptance (and some payment) should be reserved until subsystem operational testing, including any required burn-in periods, have been completed. It is important to remember that there may be considerable embedded software in most of today's ITS devices; as such, testing both at the factory and on-site should be extensive and should always test the complete product. There have been situations where vendors have delivered complete devices with "diagnostic" software that could verify proper hardware operation, but the final device functionality was not included. Hence, the agency paid for a product that did not meet their specifications because many of the required operations had not been completed. This can occur when a vendor is seeking some cash flow for large scale devices that may take months to be installed. How the agency deals with this situation should be negotiated and managed with the vendor, but they agency should recognize that it is at risk because the delivered product may not meet the contractual requirements. If the vendor should "disappear" before system completion, the agency may be in a position where it is difficult to have the product "finished" and made operational.

Following successful integration in the development environment, a delivery release version of the software system is generated for installation in the operational (production) environment. Detailed installation procedures should ensure that the new software version can be installed in the operational environment and replace the previous version with a minimum disruption to ongoing operations.

Integration typically refers to bringing together the hardware and software subsystems to achieve full system functionality. The system hardware and computer software configuration items are integrated into fully functional subsystems, which provides the first opportunity to exercise and test hardware/software interfaces and verify operational functionality in accordance with the specifications.

2.3.3.4. Integration Testing

The system test plan defines the test methodology for both the hardware and software systems comprising the TMS. It describes the testing approach and the levels of testing that are necessary to verify compliance with all the system specification requirements. This testing will be based on the methods of verification that are included in the system specification's requirements verification cross reference matrix. The top-level descriptions of the system test procedures in this plan typically assume that all applicable requirements are implemented prior to start of system level tests. However, most systems are deployed incrementally; hence, successive versions of the system test plan will be necessary, each describing the current deployment's testing requirements. The system test procedures that are developed must be tailored to the specific system configuration planned for each stage of deployment. As a result, the system test procedures that follow from the test plan, the test descriptions, test procedures, and test steps will reflect the verification process for only those requirements and partial requirements actually implemented at the time of the test. For subsequent deployments, the test procedures will need to be modified to incorporate the expanded requirement sets included in those deployments.

Acceptance testing completes the formal testing process. The acceptance test is the responsibility of the agency and is the last opportunity to make sure that the system's equipment and software meets the agency's performance and operational needs and is in full compliance with all the requirements identified throughout the project. From the vendor's standpoint, satisfactory completion of the acceptance test and approval by the agency means that the vendor has completed its contractual obligation to the agency and is due final payment.[1] Once the acceptance test has been completed, the agency "owns"[2] the system and is responsible for its operation and maintenance. Typically, it also signifies the start of the warranty period and the transfer to the agency of the hardware and software support contracts and licenses maintained by the vendor during the system development and implementation phases.

For a large, complex TMS that will be implemented incrementally over time, acceptance testing will most likely involve a number of vendors and the agency. The agency should consider conducting a number of lower level acceptance tests to accept individual components or subsystems and allow vendors to receive progress payments or partial payment at delivery, following site installation, or following initial subsystem operation. This strategy affords the agency the opportunity to start some operations early and gain operational experience while other elements of the system are being developed or deployed.

2.3.3.5. Operations

There are two aspects to systems operations that are critical to the long-term success of any complex system: 1) problem reporting and 2) configuration management or change control.

Any anomalous system behavior or suspected problems with system hardware or software functionality should be recorded [3] on a *System Problem/Change Request* (SPCR) form (see Appendix C for an example) and reported promptly by operations and maintenance personnel. Problems of a high operational impact should be reported to the operations shift supervisor or TMC manager immediately. It is important that as much information about the suspected problem and conditions at the time be recorded as soon as possible during or following the event. Problem investigation may require repeating an existing test procedure, modifying one, or creating a new one to gain enough information to isolate and resolve the problem. Problem resolution will almost always require a change to the current baseline system, possibly affecting operational procedures, maintenance actions, training, hardware and/or software, and/or system requirements. Problem clearance will require testing to verify that the problem has been resolved. If requirements change so must the test procedures to verify that the new requirements have been met. Finally, regression testing (see Section 4.4.8) must be performed to ensure that no additional problems have been introduced.

[1] Of course, this depends on the payment terms of the contract.

[2] Ownership in this case refers to the project's changing from an implementation to an operational phase. Contract terms should anticipate this event and explicitly define each parties responsibilities with respect to its occurrence. This is especially true when the contract involves intellectual property, contractor maintenance periods, or "burn-in" periods.

[3] A rule of thumb, if there isn't a written report, it didn't happen.

The SPCR provides a control point for managing corrections and changes to the system. Only approved SPCRs should be implemented, otherwise chaos will soon follow due to loss of change management and configuration control. It is important that problems be resolved in a controlled and orderly manner under the direction of the configuration control board (CCB) to ensure that the appropriate corrective actions are identified and followed so that problem resolution is both timely and cost effective. Because there is almost always a cost and schedule impact to all but the most routine problem resolution and maintenance activities, it is necessary to thoroughly understand and evaluate impacts to both ongoing and future operations.

Strict attention to problem reporting and change control will extend the lifetime and usefulness of the system and preserve its operational functionality as intended.

The operating agency must also be mindful that many of today's systems will experience anomalies during the life of the systems. After all, the basic platforms (computer and operating systems) do not exhibit the stability of systems used a decade ago. How often does one need to reboot a workstation? The agency should work with the software supplier to identify those problems needing immediate remedial action as well as those problems for which there may be acceptable work-arounds until the problem can be identified and fixed in subsequent releases. It is also worthy to note that some anomalies may be irreparable because they are the function of problems in the underlying operating system or other third party software (e.g., databases, development tools, browsers, run-time libraries) – and the ATMS software provider may be at the mercy of these vendors to correct the problem.

As noted above, software changes will necessitate significant regression testing with the associated costs. Unless the problem is mission critical, it should be packaged with managed software releases, at which time more extensive testing can take place.

Operational activities must also be planned for and conducted rigorously for the long term welfare of the system. System policies and procedures governing backup and data archiving must be designed to accommodate the hardware architecture (e.g., disk drive configuration, use of RAID, clustering, etc.) and the application's tolerance to down-time. The time spent making software backups, archiving system data, and managing the data is not truly appreciated until significant hardware failures or maintenance updates occur. The established policies will impact the time required to properly prepare for the maintenance activities as well as recover from hardware failures. These issues should be considered throughout the system life cycle in order to meet system performance and expense requirements. All of the computer systems will fail at some time or another; it is critical to establish operational procedures and verify (test) the backup procedures, backup media, and recovery procedures as part of the system acceptance testing.

2.3.3.6. Maintenance

Maintenance is a system life-cycle process that is also governed by configuration management procedures that are documented in the configuration management plan and the system maintenance plan. Every system maintenance activity will require some level of post-maintenance testing to ensure the operational functionality has been preserved or restored. A detailed system maintenance plan should be developed in concert with the system requirements specification and system test plan since many elements of the system maintenance plan are contractually binding on the various providers of system components and services and, therefore, should be included in the procurement specification.

Hardware maintenance involves repair or replacement of malfunctioning components, usually without affecting the configuration of the system; such activities can be handled outside of the configuration control board using maintenance tickets and work orders. Only those hardware maintenance problems that require a configuration change are recorded on a system problem/change request form and must be worked through the configuration control board (CCB).

Software maintenance involves implementing changes to a controlled software baseline (release version) for the purposes of correcting errors (bug fixes), adapting to environmental changes (both data and interfaces), and implementing enhancements (adding new features and revising or deleting old ones). Once an operational version of the software is placed under configuration control, all changes, whether corrections, deletions, or enhancements, should first be recorded on an SPCR form and submitted to the CCB for approval. Another form of software maintenance involves updates to operating systems and other COTS software; these must be managed in the same manner as a typical bug or new feature because "simple" operating system upgrades can have unexpected consequences. Such updates must be carefully managed and be subjected to regression testing as well.

The system maintenance plan provides the procedures and forms necessary to report and track system maintenance activities. It enables the maintenance staff to perform preventative maintenance actions, to isolate reported malfunctions to the hardware and software component level, and to perform and record component removal, replacement, and/or software fixes and upgrades. It also outlines the procedures and parameters for notification of maintenance personnel as well as their responsibilities. The system maintenance plan also applies to any outside maintenance contractors and vendors or manufacturers.

The maintenance plan should apply to all maintenance activities associated with the operation of the system subsequent to formal acceptance by the acquiring agency. Delivered and/or installed components that have not completed formal acceptance testing are not typically covered by this plan. Where possible, the responsibility for maintenance, including warranty, repair and replacement, and re-installation, rests with the supplier or installation contractor until the acquiring agency has formally accepted the component. Such terms must be clearly spelled out in the procurement specification.

There are a number of routine maintenance issues and activities that should be specifically addressed in the maintenance plan, including:

- Designating maintenance clearance notifications that specify who is responsible for making the notification and to whom.
- Implementing hardware and software upgrades, including notifying operations and maintenance personnel of any operations and procedural changes.

It is important that a failure and repair process be established to handle suspect or defective field devices properly. One such approach is to include the purchase of bench testers that can be connected to the field device that will automatically test and verify all aspects of the device operation. This can be used to verify that devices removed from the field are truly defective; it can also be used to verify that units returned as "repaired" from the vendor can be tested to ensure that they comply with the original specifications. If this approach is taken, then the procurement specifications will need to identify the requirements for the bench testers, and these might include module testers (for DMS subassemblies) or full-blown test intersections with calibrated input simulators. If the agency has a significant number of devices, it may want to explore adding environmental testing capability to its shop so that field devices can be subjected to environmental testing either for repairs or as part of an incoming inspection and testing program. If this approach is taken, it will require space, electrical planning, and staff training,

Agencies typically find it useful to maintain a system “test-bed” to attempt to duplicate system anomalies and perform detailed investigations without impacting the operation of the production system. The test-bed system is ideal when it is an exact duplicate of the production system; however, financial considerations generally result in the test-bed being a scaled down installation of processors, communications equipment, and software. The greatest benefit of the test-bed is the ability to repeat tests without compromising the integrity of the production system and is well worth the additional expense. Where it cannot connect into the production system, the test-bed should include simulators to stress test the software and communications infrastructure so that the likelihood of problems when they are installed on the production system is minimized.

2.3.4 Products of the Testing Program

The testing program produces many documents to verify that the system requirements are incorporated into the as-built system. These include many “low-level” reports such as individual unit test reports, inventory inspection reports, interface testing, and sub-system test documents. At a higher level are the verification cross reference matrix and the system acceptance test documents that tie the entire system engineering process to the completion of the contract. The system problem/change request database is another product that should be maintained for the life of the system. It will provide a measure of the system’s operation and can be an invaluable tool for tracking the resolution of system anomalies.

Organization of these products will have a significant impact on the success of the overall test program. In many cases these products will be referenced as higher level testing is conducted and during configuration management activities in the maintenance and operation life-cycle stages. It is well worth the effort to maintain these documents electronically so that they can be searched and referenced in an economical manner.

2.4 Summary

Throughout this chapter, the role that testing plays in the acquisition of a TMS and during the life cycle process has been emphasized. At each step, the importance of well-written, testable, non-ambiguous requirements has been stressed as being the foundation for both the procurement and the development of test procedures and test plans for the project. The origins of the requirements have been described as well as how and where they are documented. The chapter examined how test plans and test procedures are developed from these requirements and used for verification while the system is being developed and implemented as well as later during normal operation and maintenance. In addition, the importance of problem reporting and configuration management was discussed and identified as being critical to the long-term success of the project.

To re-iterate a basic theme of this handbook, **testing is about verifying the requirements; without testable requirements, there is no basis for a test a program.** The following chapters examine the practical aspects of system acquisition and hardware, software, and system testing.

3. The Role of Testing in the Project Life Cycle

3.1 Overview

This chapter leads the reader through the complete TMS life cycle and explains the role of testing in system definition and acquisition. It begins by discussing TMS acquisition starting with the development of a regional architecture as outlined by the National ITS Architecture. It then goes on to address TMS procurement guidelines and practices (including contract types, contract type selection based on risk allocation, and procurement specifications), project management, documentation, and configuration management.

The section on the National Architecture was included because for most regions it is a starting point for identifying user needs and establishing the regional concept of operations that will provide the major input to the design of the TMS. It is likely that all of the TMSs within the region will start with the high-level requirements defined in the regional architecture. These will then be refined and developed into the specific TMS project concept of operations and high-level requirements. Therefore, it is important to understand the role that the national and regional architectures play when establishing a TMS deployment program.

3.2 National ITS Architecture

The following sections describe the underlying architecture used to build a TMS and how that architecture came to be.

The Intermodal Surface Transportation Efficiency Act of 1991 (ISTEA) initiated Federal funding for the National ITS program. The program at that time was largely concentrated on research and development and operational tests of emerging technologies for ITS programs. A key component of the program was the development of the National ITS Architecture.

The purpose of the National ITS Architecture was to provide a common structure for the design of ITS throughout the country. As this architecture has evolved, it has directed attention to defining the functions that could be performed to satisfy user requirements and how the various elements of ITS might connect to shared information. The National ITS Architecture is neither a system design nor a design concept; the purpose of preparing an ITS architecture is to ascertain:

- The functions that are required for the ITS.
- The physical entities or subsystems where these functions reside.
- The information flows that connect these functions and the physical subsystems together into an integrated system.

With the passage of the Transportation Equity Act for the 21st Century (TEA-21), ITS projects funded through the Highway Trust Fund were required to conform to the National ITS Architecture and applicable standards. Conformance with the National ITS Architecture was interpreted to mean using the National ITS Architecture to develop a regional ITS architecture and requiring all subsequent ITS projects adhere to that regional ITS architecture as well. The National ITS Architecture is to be used as a resource in the development of the regional ITS architecture, and the subsequent regional ITS architecture is to be on a scale commensurate with the scope of ITS investment within the region.

This legislation imposed the following requirements:

- All ITS projects funded by the Highway Trust Fund shall be based on a systems engineering analysis on a scale commensurate with the project scope.
- Compliance with the regional ITS architecture must be in accordance with United States Department of Transportation (USDOT) oversight and Federal-aid procedures, similar to non-ITS projects.
- ITS projects funded by the Highway Trust Fund and the Mass Transit Account must conform to a regional architecture.
- Projects must use USDOT-adopted[4] ITS Standards as appropriate.
- Regions currently implementing ITS projects must have a regional architecture in place in *4 years,* while regions not currently implementing ITS projects must develop a regional ITS architecture *within 4 years* from the date the first ITS project advances to the final design.
- Major ITS projects should move forward based on a project-level architecture that is coordinated with the development of the regional ITS architecture.

The concept of operations document should identify the roles and responsibilities of participating agencies and stakeholders for the regional ITS architecture developed for the TMS.

3.2.1 National ITS Architecture Description

The National ITS Architecture consists of logical and physical components as depicted in figure 3-1 below and described in the following paragraphs.

[4] This phrase has led to significant confusion. The USDOT must invoke a rule-making program to formally adopt ITS standards which become mandatory under this clause. To date, the USDOT has not undertaken rule-making procedures for the ATC, 2070, NTCIP, TMDD, 1512, or J2354 standards and hence their use is optional, although strongly encouraged. Adherence to the evolving national standards is critical for the piece-wise deployment of ITS technology and systems within a region and hence adherence to these standards is essential.

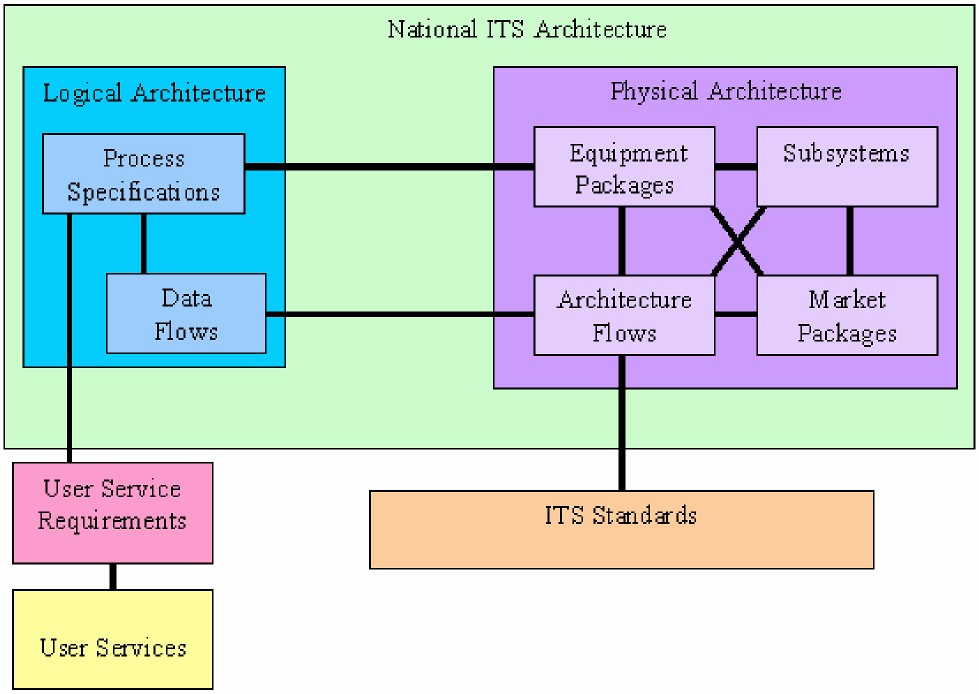


Figure 3-1. National ITS Architecture Components

3.2.2 Logical Architecture

While the user services identified the ITS needs of the agency's region, the logical architecture defines the functions that need to be carried out to support the selected user services. The logical architecture covers all of the user service requirements and is technology independent. It identifies the boundaries of the architecture, the functions to be performed, and the relationship between functions. Entities (e.g., businesses, organizations, vehicles, devices, systems) outside the ITS boundaries are referred to as terminators. What the logical architecture does not define is where the functions are performed and how they are implemented.

3.2.2.1. Process Specifications

Processes are the functions identified in the logical architecture that are required to support the agency's selected user services. The logical architecture presents processes in a top-down fashion beginning with the general processes that are decomposed into more detailed processes. The general processes are defined in terms of more detailed processes using data flow diagrams.

3.2.2.2. Data Flow Diagrams

Data flow diagrams show the information that is transferred between processes or between a process and a terminator in the logical architecture.

3.2.3 Physical Architecture

While the logical architecture provides a "functional" view of the system, the physical architecture provides a representation of how ITS should provide the required functionality. Although not a detailed design, the physical architecture takes the processes previously identified in the logical architecture and assigns them to physical entities (subsystems or terminators). Furthermore, the data flows from the logical architecture that originate at one subsystem and end at another are grouped together in physical architecture flows.

3.2.3.1. Subsystems

Subsystems are individual pieces of ITS defined by the National ITS Architecture. They are the principle structural element of the physical architecture view. Subsystems are grouped into four classes: Centers, Field, Vehicles, and Travelers (see figure 3-2). Example subsystems are the Traffic Management Subsystem, the Vehicle Subsystem, and the Roadway Subsystem. These correspond to the physical world of traffic operations centers, automobiles, and roadside signal controllers.

3.2.3.2. Architectural Flows

Architectural Flows represent the information that is exchanged between subsystems and terminators in the physical architecture. These architecture flows and their communication requirements define the interfaces that form the basis for much of the ongoing standards work in the national ITS program.

3.2.3.3. Equipment Packages

Equipment packages are the building blocks of the physical architecture subsystems. They partition the subsystems into deployment-sized pieces.

3.2.3.4. Architecture Interconnect Diagram

Figure 3-2 shows the top-level architecture interconnect diagram, which depicts the subsystems for full representation of the National ITS Architecture and the basic communication channels between these subsystems.

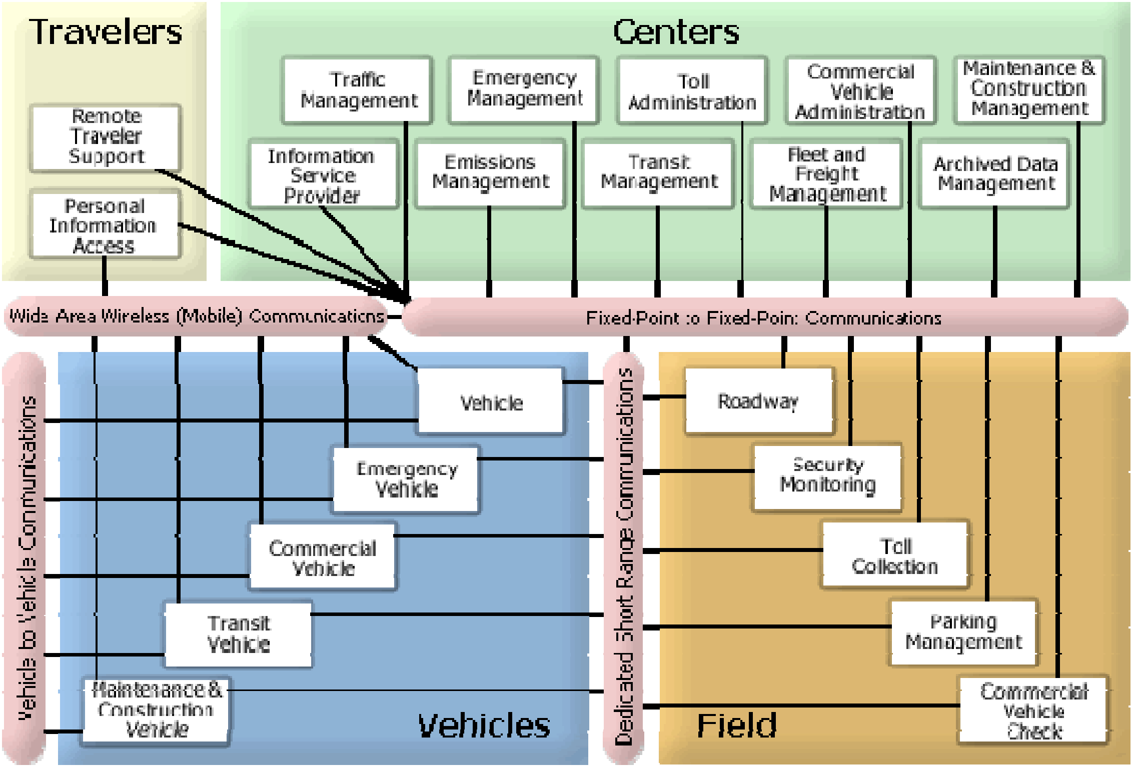

Travelers
Remote Traveler Support
Personal Information Access
Centers
Traffic Management
Emergency Management
Toll Administration
Commercial Vehicle Administration
Maintenance & Construction Management
Information Service Provider
Emissions Management
Transit Management
Fleet and Freight Management
Archived Data Management
Wide Area Wireless (Mobile) Communications
Fixed-Point to Fixed-Point Communications
Vehicle to Vehicle Communications
Vehicle
Emergency Vehicle
Commercial Vehicle
Transit Vehicle
Maintenance & Construction Vehicle
Vehicles
Dedicated Short Range Communications
Roadway
Security Monitoring
Toll Collection
Parking Management
Commercial Vehicle Check
Field

Figure 3-2. National ITS Architecture Interconnect Diagram

3.2.4 User Service Requirements

User service requirements are a specific functional requirements statement of what must be done to support the ITS user services. The user service requirements were developed specifically to serve as a requirements baseline to drive the National Architecture development. The user service requirements are not to be construed as mandates to system/architecture implementers, but rather are directions to the National Architecture Team.

3.2.5 User Services

User services document what the ITS should do from the user's perspective and include a broad range of users, including the traveling public as well as many different types of system operators. The concept of user services allows system or project definition to begin by establishing the high-level services that will be provided to address identified problems and needs.

3.2.6 ITS Standards

Standards are an important aspect of the National ITS Architecture. They provide the means by which compatibility between systems can be achieved. They also promote multi-vendor interoperability and ease of integration (if properly specified). As part of the National ITS Architecture development process, USDOT initiated an effort to accelerate development of consensus-based standards using the interconnection requirements (i.e., interfaces) defined in the architecture.

The specific intent of these standards development efforts is to provide a number of long-term benefits, including interoperability, increased competition, expandability, lower costs, and increased system integration.

Agencies can take advantage of these standards as they emerge by specifying their use in procurement packages. The standards also provide many options and require specific implementation choices when used with each TMS project. The applicable options will vary according to the operational concepts incorporated into the design and must be directly specified. Each selected option will have specific impacts on the testing process later in the procurement process. Many projects have not met their expectations because individual project specifications incorporated the standards by reference and did not identify the individual options necessary to fulfill the operational intent of the system. An appropriate level of planning and engineering activities must be performed in applying the standards to the acquisition of TMS projects.

Among the pertinent national ITS standards development activities in process are the suite of standards being developed under the National Transportation Communications for ITS Protocol (NTCIP) effort. There are also a number of other existing communication and information-based standards that are applicable to ITS projects.

The following list some of the applicable ITS standards with respect to data elements, message sets, and communication protocols.

- Data elements:
    - Traffic Management Data Dictionary (TMDD) – reference the Institute of Transportation Engineers (www.ite.org/TMDD).
    - NTCIP standards for the communications with the various ITS devices such as traffic controllers, dynamic message signs, environmental monitoring stations, CCTV cameras and switches, ramp controllers, etc. (reference www.ntcip.org) and refer to document NTCIP 9001 for a guide to these standards.
    - Advanced Traveler Information System (ATIS) data dictionary (SAE J2354).
- Message sets:
    - Transit Communication Interface Profile (TCIP) (APTA).
    - Message Sets For External Traffic Management Center Communications (MS/ETMCC – now part of the ITE TMDD effort).
    - Incident Management Message sets (IEEE-1512).
    - Advanced Traveler Information System (ATIS) message sets (SAE J2354).
- Communication protocol:
    - National Transportation Communications for Intelligent Transportation Systems (ITS) Protocol (NTCIP). These protocols identify center-to-center and center-to-field communications protocols for a variety of media (e.g., analog telephone, Ethernet).

In addition to the ITS-specific standards involving communications data, messages, and protocols, existing and emerging standards are available for many ITS devices and/or their interfaces (e.g., signal controllers, video cameras, dynamic message signs, etc.). These standards can be reviewed and considered for incorporation in an ITS project based on the engineering analyses. A list of references in the last chapter provides access to many of the standards organizations discussed in this section.

3.3 Transportation Management Systems Procurement

Once a regional architecture and at least a preliminary concept of operations have been developed, system procurement and funding issues can be addressed. The following are some typical contract types that can be used;[5] each necessitates a different level of direct participation and technical expertise by the acquiring agency. The type of contract selected will also dictate who prepares procurement specifications and the management structure needed to oversee the process. It will also have an impact on who develops and performs the testing program throughout the life cycle of the project.

3.3.1 Contract Types

The following table describes some of the different contract types that can be used to acquire a TMS and its subsequent ITS devices and explains what the agency's responsibilities would be as a consequence of using each type.

Table 3-1 System Acquisition Contract Types and Agency Responsibilities

Contract Type	Description	Acquiring Agency Responsibilities
Design	A qualified **design engineering contractor** is engaged to develop the detailed design that includes hardware and software requirements, procurement specifications, installation drawings, and integration and test plans and procedures. The designer is usually required to provide support during the construction (implementation) phase to correct design deficiencies and provides oversight and design support for changes.	The acquiring agency is required to manage the design contract and review and approve the design documents, procurement specifications, installation drawings, and the test plans and procedures. The resulting system will depend on the design engineer's understanding of the agency's needs and requirements.
Build	A separate, qualified **implementation contractor** is needed to verify and implement the design and installation drawings, develop or purchase the hardware and software components specified by the design contractor, and install and test those components. The implementation contractor may also be required to develop training plans and conduct training for system operations and maintenance personnel.	The acquiring agency is required to manage the construction contract, inspect and accept installations, approve changes and training plans, and witness and approve acceptance testing.

[5] Additional contracting guidance and other contract types can be found in *NCHRP Report 560: Guide to Contracting ITS Projects,* by K. Marshall and P. Tarnoff, published by the Transportation Research Board, Washington, DC: 2006.

Contract Type	Description	Acquiring Agency Responsibilities
Design/ Build	A single qualified **contractor** is engaged to create the design and then implement it.	The acquiring agency is required to manage the design/build contract; review and approve the design documents, procurement specifications, installation drawings, training plans, test plans and procedures; inspect and accept installations; approve changes; and witness and approve acceptance testing.
System Integrator	A separate qualified **integration contractor** is engaged to oversee the design and implementation phases and is usually responsible for reviewing and recommending approval of the design documents, procurement specifications, installation drawings, training plans, test plans and procedures; inspecting and accepting installations; conducting system level operations and maintenance training; and conducting and supporting testing. System integrators are contracted for design and implementation services under either separate agreements or a single agreement; however, some service elements of the system integration and test phase may be carved out for subcontractor integration contracts.	The acquiring agency is required to manage the lead system integrator contract, approve documentation and changes, and witness and approve acceptance testing.
System Manager	A qualified contractor is engaged to direct and manage all phases and aspects of the system development and implementation. The system manager designs and implements the system or contracts and manages the contracts for those elements of the design/build that the system manager does not perform. The system manager may also be responsible for the ongoing operations and maintenance of the system after it is accepted.	The acquiring agency is required to approve procurement practices and any major contracts let by the system manager. The acquiring agency is also required to oversee the system manager's contract, participate in the approval of documentation and changes, and witness and approve acceptance testing. The operating agency may desire to provide staffing for various operations and maintenance positions. How well the system meets the needs of the agency will depend on the relationship and understanding between the manager and the agency.

3.3.2 Contract Type Selection

Each of the above contract types exposes the acquiring agency to a different level of technical and financial risk and each carries with it a different test program burden on the agency (i.e., direct participation in as well as review and approval of designs, test plans, and procedures). The following table contrasts the contract types and relative financial and technical risk allocation against the agency's direct involvement in the test program for a multimillion-dollar TMS project.

Table 3-2 Contract Types: Agency Risk Allocation and Test Program Burden

Contract Type	Financial Risk	Technical Risk	Test Program Burden
Design	Low	High	High
Build	Medium	Medium High	High
Design/Build	Medium	Medium	Medium
System Integrator	Medium High	Low	Low
System Manager	High	Very Low	Very Low
Notes: A low *financial risk* implies a small contract amount with respect to the total TMS cost (sum of procurement contracts – labor and materials plus agency contract management plus agency direct labor). A high *technical risk* implies that the agency is dependent on its in-house technical expertise to oversee the technical aspects of the hardware and software design and implementation. A medium *test burden* level indicates that the agency has sufficient technical expertise and manpower to share the burden of the test program equally with the contractor(s).			

For example, if the acquiring agency wants to use a contract type that has low technical risk (the design and implementation technical risk is borne by the contractor) and does not want very much agency participation in the test program (because of its limited technical expertise and/or ability to travel), it might select a system integrator contract type. While potentially more expensive than a design/build contract type, it would be used if the agency was willing to pay the contractor to accept most of the risk in implementing a successful TMS project. That technical risk and test program burden could be reduced even more with a system manager type contract, but at a potentially higher contract cost.

Another aspect of contract type to consider before committing to procurement is how the TMS will be deployed; i.e., all at once or incrementally, in phases by extending capabilities, or in stages with increasing levels of functionality. Generally, the availability of funding will dictate the deployment approach, but it should be noted that the overall test program costs will be greater for an incrementally deployed system because, as each phase or stage is completed, additional test planning, test procedures, and regression testing will be necessary. However, the benefits that will ultimately accrue to the public using that system will begin much earlier in the project.

Similarly, the impact of existing systems and their age should be considered when selecting the contract type. Rolling out a TMS where none previously existed has fewer risks associated with it than the replacement or even incorporation of existing TMS components. The responsibility for operating existing TMS components and integrating existing components should be weighed carefully as costs will vary with the assignment of risks and responsibilities. Testing existing facilities may be necessary to ensure that the facilities can be successfully integrated with the new system and to identify any pre-existing deficiencies that would leave the existing equipment performing below expectations with the new system.[6]

[6] How accurate is the documentation for the existing system? Was the external interface previously tested or was it simply a requirement that went untested due to a lack of resources and urgency? In latter case, who assumes the risk when the interface does not meet its documented characteristics?

When the TMS is being incrementally deployed, the procurement specifications should identify certain phases or stages at which substantial portions of the system are acceptance tested in order to allow those portions to be used operationally. For example, if the system is being deployed along a transportation corridor that extends for many miles, it may be prudent to break the procurement into manageable geographic phases such that each phase covers a contiguous segment of the total system corridor and can be procured and accepted separately. This allows concurrent phases, when funding permits, and different developers, vendors, or contractors to be used in each phase if desired. If these procurements are properly structured, i.e., the first phase contains a functional TMC and the initial communications infrastructure to connect field devices along part of the corridor, then meaningful traffic management operations can start following acceptance of the first phase while development, installation, and testing activities are ongoing in subsequent phases. One important consideration for this approach, however, is the need to test the TMC system and subsystem capacity for the fully configured build-out; for example, if the final configuration will include 100 dynamic message signs (DMS) and 2,000 controllers, even though the initial phase has only 10 DMS and 100 controllers, the testing program for the TMC systems must demonstrate management capability (and communications support) for the fully configured system.

3.3.3 Procurement Specifications

The procurement specifications, whether developed by the acquiring agency or under one of the above contract types, must contain all the system requirements and acceptance test requirements. Poor or missing requirements will result in inadequate designs; missing capabilities and functionality; and performance, operations, and maintenance shortfalls. Furthermore, correcting these deficiencies will impact both costs and schedule; therefore, it is important that the procurement specifications include detailed, unambiguous, precise, and measurable requirements and that those requirements are written such that the components can be tested, inspected, or observed to verify compliance with the requirements. Without a strong procurement document and well-written requirements, there is little basis for the test program and ultimate system acceptance.

Testing is about verifying that what the developer or the vendor has delivered complies with what was specified. As such, testing verifies that the product meets the functional, performance, design, and implementation requirements identified in the procurement specifications. The testing program should require that the developer or vendor demonstrate that their product(s) meet all of the requirements of the procurement specifications, including any standards referenced.[7] However, it is important to keep in mind that the degree to which a product/device is tested should be balanced against the cost of performing that testing and the perceived risk of the product to the success of the program, the complexity and maturity of product, and the number of units involved.

The procurement specifications should outline how the testing program fits into the overall project schedule. The schedule must allow sufficient time for the agency to review the test plan(s) and for the developer or vendor to make corrections. The testing can be complex and lengthy; a detailed test plan can easily run into hundreds of pages with the inclusion of the test environment schematics, descriptions of the test software and simulation environment, and copies of the test cases,

[7] Examples of standards includes NEMA TS2-2003 for traffic controllers, NEMA TS4 for DMS, California TEES for traffic control devices, UL requirements, and the National Electric Code.

inspections, and requirements traceability check lists. The acquiring agency will most likely need technical expertise (e.g., software, mechanical, and electrical engineers and operations and maintenance personnel) to review test plan submittals. It is recommended that prospective bidders for the project be required to submit example test procedures, at the level of detail being proposed, so that they can be evaluated for adequacy and completeness during the procurement (qualification) process. This will also afford the acquiring agency an opportunity revise final procurement specifications and to ascertain the competence of its technical staff to review and approve test plan material.

Re-use of existing procurement specifications from other related or similar projects and selection of standard products from a Qualified Products List (QPL) can greatly reduce system acquisition costs. However, this does not eliminate the need for a comprehensive review of the detailed procurement requirements, referenced standards, acceptance test procedures, and the modification thereof to assure the needs of this specific project are being met.

3.4 Project Management, Staffing and Training

The project management and staffing structure will be dictated by the scale and complexity of the proposed TMS and its acquisition and development strategy. Specifically, the development of the procurement specification(s) and selection and management of the contract type(s) for system development and implementation will determine the staffing and training needs. Additionally, future staffing needs will be determined by what contracting steps are taken to support operation and/or maintenance of the system after it is implemented. The development and implementation phases will extend for many months, if not years, and will require a large and continuing commitment of personnel and resources.

It is important to recognize, early in the process, the critical role that testing plays in all aspects of a successful project. An independent test organization (i.e., from the development and implementation organizations), managed and staffed by knowledgeable and well-qualified engineering and technical personnel, is a must. The size, technical diversity, and depth of the testing organization should be in direct proportion to the project's scale, technology, and complexity, and the contract type(s) used for procurement.

It is unlikely at the outset of the project that the agency's current staff will have sufficient technical training for all aspects of the project. This is particularly true with respect to new and emerging technologies and test techniques that may be employed on the project. Thus, staff augmentation and on-the-job training provided by vendors and consultants may be necessary and should be accounted for in the project budgeting and procurement specifications.

A large project may have both an acquiring agency that is responsible for the acquisition of the TMS and an operations agency that will operate and maintain the system once it is accepted and becomes operational. An ITS consultant that has technical expertise and practical experience in all aspects of TMS development and operations may also be employed to assist the acquiring agency and the operation agency during the developmental and operational phase of the project.

3.5 Documentation

A set of formal documentation is essential for the design, development, implementation, test, operation, maintenance, and administration of the TMS throughout the system's life cycle. The concept of operations is one of the baseline documents used to provide a general overview of the operational objectives and the management and support organizational structure envisioned for the operation of the system. This plan, along with a long-range strategic plan, provides the basis for developing critical programmatic documents, including the system architecture description, system specification, system test plan, and configuration management plan as well as the hardware/software design and requirements documents, test procedures, and training and maintenance plans. Figure 3-3 depicts an example document tree (usually included in the system specification) showing the hierarchy of these documents and how they are derived from each other. (Note: additional detail for some of documents in the document tree can be found in this document in the sections indicated above them in red in figure 3-3.)

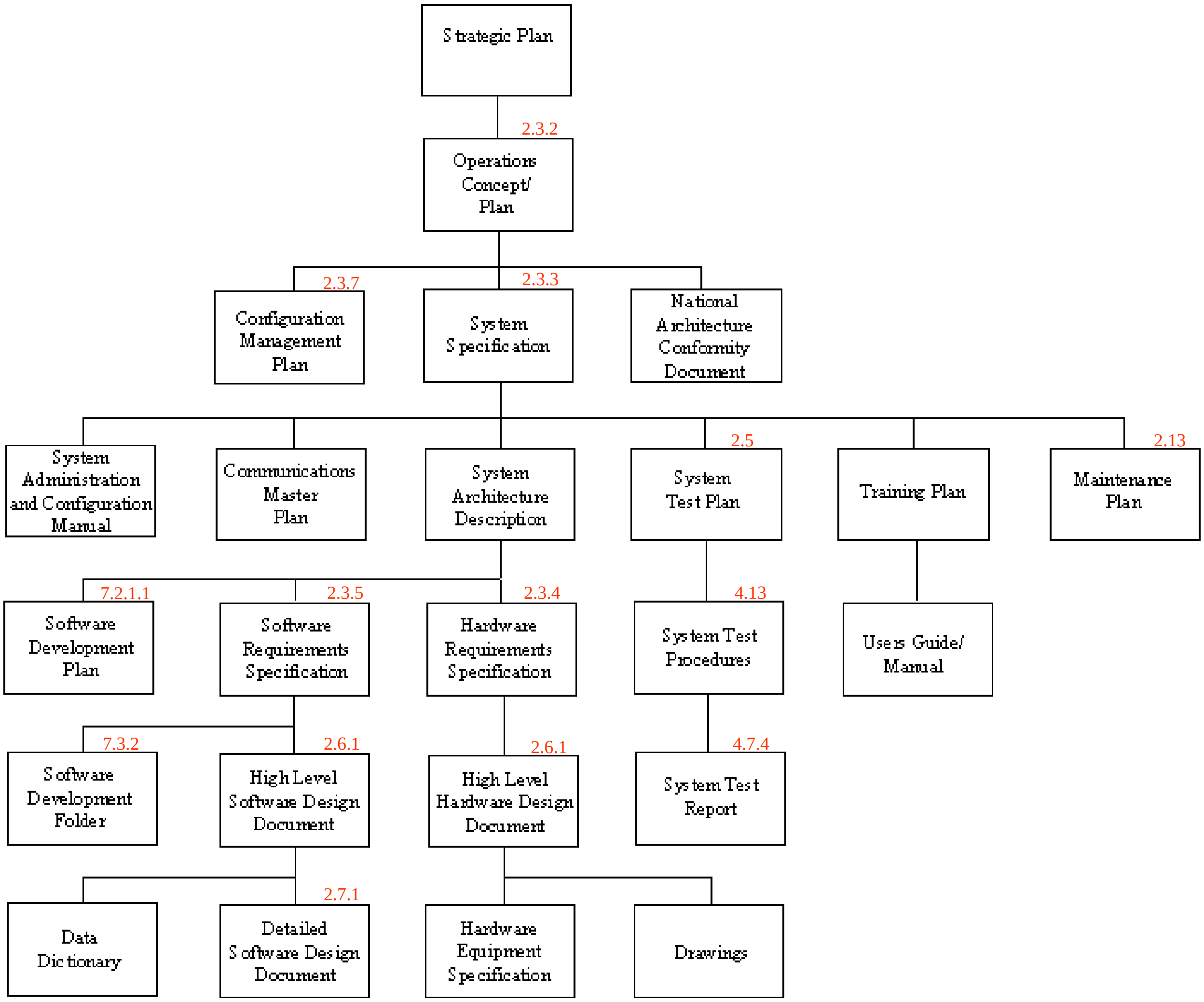

Strategic Plan
2.3.2
Operations Concept/ Plan
2.3.7
2.3.3
Configuration Management Plan
System Specification
National Architecture Conformity Document
2.5
2.13
System Administration and Configuration Manual
Communications Master Plan
System Architecture Description
System Test Plan
Training Plan
Maintenance Plan
7.2.1.1
2.3.5
2.3.4
4.13
Software Development Plan
Software Requirements Specification
Hardware Requirements Specification
System Test Procedures
Users Guide/ Manual
7.3.2
2.6.1
2.6.1
4.7.4
Software Development Folder
High Level Software Design Document
High Level Hardware Design Document
System Test Report
2.7.1
Data Dictionary
Detailed Software Design Document
Hardware Equipment Specification
Drawings

Figure 3-3. System Document Tree

3.6 Configuration Management

The objective of configuration management (CM) is change control. Change is inevitable and the rule of thumb is that things change at the rate of roughly 1 percent per month. The goal of any project is to manage change, understand its implications, and integrate the changes into the project plan and testing program. CM provides formal change control methodologies throughout all stages of system deployment to include development, implementation, installation, test and acceptance. CM provides a mechanism for maintaining formal records of the TMS configuration to include software, communications cable plant, field equipment, communication electronics, TMS hardware, and all formal system documentation including test plans, test procedures and test reports. CM also provides a mechanism for managing changes during the system's life cycle and enables the acquiring and operating agencies to effectively and efficiently plan future system enhancements and expansions.

Generally, system documents such as system test plans, system and subsystem requirements and specifications, data dictionaries, equipment installation plans, and as-built drawings are placed under CM when the final version of the contract deliverable is approved by the acquiring agency. This establishes a baseline system configuration of software, hardware, and communications infrastructure. Any changes (including those made to correct problems or deficiencies) to the baseline system configuration, in terms of software changes, new hardware components, and/or communication electronics, can be tracked in order to minimize adversely affecting the overall functionality of the system in the future. Accordingly, a formal change control process is placed under the oversight of a configuration control board (CCB). Additionally, change request forms called system problem/change requests (SPCR) are used to initiate and process changes to the system configuration and documentation.

The CCB is responsible for the configuration management and change control decisions over the life cycle of the system. It acts under the auspices and at the direction of the acquiring and operations agencies. Its decisions may have significant programmatic cost and schedule impacts. It is the function of the CCB to accept configuration items (CI) for configuration management and control and to approve or reject all requested changes to CIs under control. It is the responsibility of the CCB to ensure that the appropriate level of testing is required and completed for all approved changes (i.e., test plans must be updated based on changes approved by the CCB).

In order to perform these functions, the CCB is comprised of a team of permanent members drawn from the day-to-day system managers, operations personnel, and invited technical specialists and advisors, each with specific responsibilities and duties, who collectively manage the system configuration and any changes made to it.

The following diagram depicts a nominal structure and membership of the CCB. The board positions shown in figure 3-4 are the permanent positions. Board positions can and should be expanded as necessary to handle unique circumstances or special needs. For example, the hardware and software board positions can be augmented during system development and/or expansion phases to include the area managers responsible for the development or expansion. ITS operations engineers and planners can also serve as board members. Technical specialists, advisors, and other agency representatives may be appointed to the board to assure comprehensive oversight of proposed changes, especially when the change potentially impacts other operations or systems. The SPCR

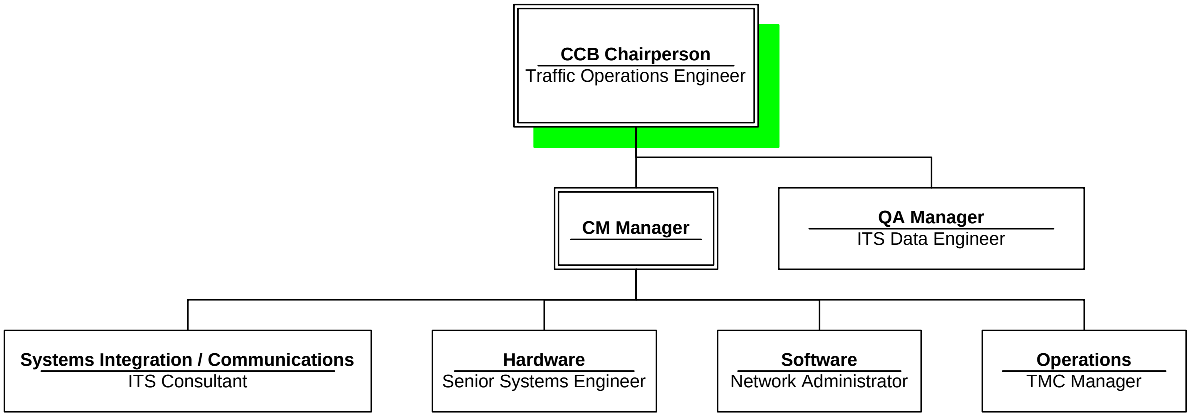


Figure 3-4. Configuration Control Board (CCB)

originator or CI specialist may also be invited to attend a CCB meeting to provide additional expertise for technical discussion of problems or requested change.

Test program documents (e.g., program plan, test plans, procedures, reports) are all CIs managed by the board. As part of the CM process, the test team representative may be required to report test results to the CCB in order to move changes to software and hardware CIs into production environments.

3.7 Summary

This chapter provided a brief overview of the National ITS Architecture and how it relates to the procurement and development of an ITS project. Next, some of the procurement avenues (contract types) available to the acquiring agency, the financial and technical risks associated with each, and the resulting testing burdens on the agency were presented. In the context of system acquisition and development, recommendations were made for project management, staffing and training concepts needed for a successful project including the use of an independent test organization. The need for staff augmentation and on-the-job training provided by vendors and consultants at the outset of the project was suggested as a means of overcoming a shortfall in agency's current staff and technical training and that this be accounted for in the project budgeting and procurement specifications. The need for formal documentation was stressed as essential for the design, development, implementation, test, operation, maintenance, and administration of the TMS throughout the system's life cycle. The chapter concluded with a discussion of the importance of creating a configuration control board (CCB), which is responsible for both the configuration management and change control decisions that may have significant programmatic cost and schedule impacts as well as for ensuring the appropriate level of testing is required and completed for all approved changes.

4. An Introduction to Testing

4.1 Overview

This chapter provides a testing tutorial that addresses the basics of testing. It discusses testing methods, planning, test development, resources, and execution. It is not of sufficient depth or breath to make the reader an expert, but it provides an introduction to the concepts and terminology that are likely to be used by system vendors throughout the test program.

4.2 Test Methods

The system specification should include a requirements verification matrix that details the test method(s) to be employed to verify each system requirement and is copied to the system test plan where three columns are added: test level, test responsibility, and test identification number. This matrix is known as a verification cross reference matrix (VCRM) or traceability matrix. The complexity and ultimate cost of the system test program is directly related to the test method(s) specified for verification of each requirement. As the test method becomes more rigorous, performing that testing becomes more time consuming, requires more resources and expertise, and is generally more expensive. However, the risk of accepting a requirement under a less rigorous testing method may be undesirable and may ultimately prove to be more expensive than a more rigorous test method. It is important to consider the tradeoffs and consequences attendant to the specification of the test methods carefully since they flow down with the requirements to the hardware and software specifications and finally to the procurement specification. The following paragraphs define the five basic verification methods. From a verification standpoint, **inspection** is the least rigorous method, followed by **certificate of compliance**, **analysis**, **demonstration**, then **test** (formal) as the most rigorous method. A vendor's certificate of compliance may be evidence of a very rigorous development and test program, but that testing is typically not defined, approved, or witnessed by the system's acquiring agency; hence, there is some risk in accepting that certification, however, that risk is often outweighed by the costs of performing equivalent testing by the acquiring agency. Remember that the vendor's test program costs are embedded in the final component pricing.

The following sections will examine what is required for each method.

4.2.1 Inspection

Inspection is the verification by physical and visual examinations of the item, reviewing descriptive documentation, and comparing the appropriate characteristics with all the referenced standards to determine compliance with the requirements. Examples include measuring cabinets sizes, matching paint color samples, observing printed circuit boards to inspect component mounting and construction techniques.

4.2.2 Certificate of Compliance

A Certificate of Compliance is a means of verifying compliance for items that are *standard products*. Signed certificates from vendors state that the purchased items meet procurement specifications, standards, and other requirements as defined in the purchase order. Records of tests performed to verify specifications are retained by the vendor as evidence that the requirements were met and are made available by the vendor for purchaser review. Examples of “vendor certification” are test results and reports from testing to verify compliance with NEMA TS2-2003 testing standards that were performed by an independent testing laboratory. Other examples include more rigorous testing and certification that may have been performed by an industry accepted testing facility for compliance with such standards as 802.11a for wireless devices. In both of these instances, the certificate represents a test process that was completed at considerable expense to the vendor and is generally only required for a new product or significant product changes.

4.2.3 Analysis

Analysis is the verification by evaluation or simulation using mathematical representations, charts, graphs, circuit diagrams, calculation, or data reduction. This includes analysis of algorithms independent of computer implementation, analytical conclusions drawn from test data, and extension of test-produced data to untested conditions. This is often used to extrapolate past performance (which was accurately measured) to a scaled up deployment. An example of this type of testing would include internal temperature gradients for a dynamic message sign. It is unlikely that the whole sign would be subjected to testing within a chamber, so a smaller sample is tested and the results can be extrapolated to the final product. Along the same vein, trying to determine the temperature rise within a DMS may require analysis based on air flow, fan ratings, vent sizes etc. Other examples include the review of power supply designs to verify that they comply with junction temperature and voltage limitations contained within the procurement specifications.

4.2.4 Demonstration

Demonstration is the functional verification that a specification requirement is met by observing the qualitative results of an operation or exercise performed under specific condition. This includes content and accuracy of displays, comparison of system outputs with independently derived test cases, and system recovery from induced failure conditions.

4.2.5 Test (Formal)

Formal testing is the verification that a specification requirement has been met by measuring, recording, or evaluating qualitative and quantitative data obtained during controlled exercises under all appropriate conditions using real and/or simulated stimulus. This includes verification of system performance, system functionality, and correct data distribution.

4.3 Test Approach

Several levels of testing are necessary to verify the compliance of all of the system requirements. A “building block” approach (see Section 5.2) to testing allows verification of compliance to requirements at the lowest level, building up to the next higher level, and finally full system compliance with minimal re-testing of lower level requirements once the higher level testing is performed.

After components are tested and accepted at a lower level, they are combined and integrated with other items at the next higher level, where interface compatibility, and the added performance and operational functionality at that level, are verified. At the highest level, system integration and verification testing is conducted on the fully integrated system to verify compliance with those requirements that could not be tested at lower levels and to demonstrate the overall operational readiness of the system.

Contractors supplying system hardware, software, installation and integration are typically required (by the agency's procurement specification) to develop and execute verification test plans and procedures. These test plans are required to trace to the contract specification functional and performance requirements and the test plans are approved by the acquiring agency. The contractors' test plans and test procedures are thus an integral part of the overall system test program. A typical system test program has five levels of verification tests. These levels are defined in table 4-1 along with an example of test responsibilities. Note that the test responsibility (i.e., the development and execution of the test procedures) typically rests with the component developer, installer, or integrator, but test approval and test acceptance is an acquiring agency responsibility. That responsibility, however, particularly at the lower test levels, may be delegated to a consultant due to the specific experience and expertise required. Actual test responsibilities will vary from this example based on the project implementation plan, resulting contracts, and the test team staffing plan.

Table 4-1. Test Levels and Example Test Responsibilities

Test Level	Test Type[1]	Test Responsibility
1	Software unit/component tests	Software Developer, ITS Consultant/Acquiring Agency
	Hardware unit/component tests	Vendors
2	Software build integration tests	ITS Consultant
	Hardware assembly/factory acceptance tests	Vendors, ITS Consultant/Acquiring Agency
3	Software chains and hardware/software integration tests	ITS Consultant
	Control center hardware integration tests	Installation Contractors, ITS Consultant
	Control center demarcation interface tests	Service Providers, ITS Consultant
	Communication/electronic hardware field equipment and field device integration tests – this may also include verification of compliance/conformance to specific standards for the interfaces and protocols (NTCIP)	Installation Contractors, ITS Consultant, Standards expert
	Communication connectivity	Installation Contractors, ITS Consultant
4	Subsystem integration and element acceptance tests	Acquiring Agency/Operations Agency
5	System integration and system acceptance tests	Acquiring Agency/Operations Agency

[1] While not shown explicitly in this table, testing also includes design and document reviews and audits of internal processes. The test team is assigned the responsibility for audits of internal processes. These audits/tests are tailored to ensure that project records and standards are being met and followed. For instance the test team can track software development folders to verify that they contain records of peer reviews and that the resolution of peer review action items have been addressed.

As part of the system test program (using the example responsibilities as shown in table 4-1), the ITS consultant is tasked to review and approve contractor submitted subsystem test plans, test procedures, and test results for the acquiring agency. The ITS consultant audits this test documentation to assure all lower level requirements verification testing is appropriate and complete, properly executed and witnessed, and formally reported. In addition, the ITS consultant assists in coordinating and scheduling system resources not under the control of the contractors conducting the tests and monitors tests as appropriate for the acquiring agency.

4.4 Test Types

This section discusses each of the test types introduced above and provides some specific guidance and recommendations for the acquiring agency with respect to conducting these tests.

4.4.1 Unit Testing

The hardware component level is the lowest level of hardware testing. Unit tests are typically conducted at the manufacturing plant by the manufacturer. At this level, the hardware design is verified to be consistent with the hardware detailed design document. The manufacturer retains test documents showing compliance with design and manufacturing standards and materials certifications. Due to the counterfeiting of parts that is possible today, material certifications are taking on greater importance. Agencies need to contractually ensure that they have access to vendor’s material records to verify the source of materials. The procurement specification should require access to these test documents. The acquiring agency may desire to witness some of this testing, particularly if the hardware component is a new design. If so, the procurement specifications should require agency participation at this level of testing.

The computer software component level is the lowest level of software testing. Stand-alone software unit tests are conducted by the software developer following design walk-throughs and code inspections. At this level, the software design is verified to be consistent with the software detailed design document. Unit-level testing is documented in software development folders. Receiving inspections and functional checkout are performed for COTS software to assure that these components are operational and in compliance with their respective specifications. The acquiring agency may desire to witness some of this testing; however, unless it has experience with software development procedures and practices, software unit test and code inspections will be difficult at best to follow and understand. Test participation at this level is best left to an independent test team with the agency reviewing the audits of development folders and project records. Agency participation in receiving inspections and functional checkout of COTS software can be useful particularly if the agency has not seen a demonstration of that software. For example, if Geographical Information System (GIS) software will be used as a primary operator interface to show geographic features, roadway networks, ITS devices, incidents, and congestion information as well as act as the principle command and control interface, then the agency needs to have an understanding of how this software works and whether it meets the agency’s operational needs. If the agency doesn’t feel it meets its needs or requirements as documented, then this is the time to select another product. Even if it meets the apparent requirements but is considered unacceptable, it is far better to address the issue sooner rather than later when the consequences are much more severe. As a side note, this type of incremental review and approval should be part of the testing program so that the agency can understand how the requirements have been translated into operational software.

4.4.2 Installation Testing

Installation testing is performed at the installation site subsequent to receiving inspections and functional testing of the delivered components. Here, checklists are used to assure that any site preparation and modifications, including construction, enclosures, utilities, and supporting resources have been completed and are available. Specific emphasis on test coordination and scheduling, particularly for the installation of communications infrastructure and roadside components, is essential to achieving successful installation testing and minimizing the associated cost and should be detailed in procurement specifications. Installation testing is an excellent point in the test program for the agency to begin involving maintenance and, to a lesser extent, operations personnel as test witnesses, observers or participants in the test procedures. The lessons learned by the agency staff during this process will be invaluable when the agency takes over day-to-day operation and maintenance responsibilities.

4.4.3 Hardware Integration Testing

Integrated hardware testing is performed on the hardware components that are integrated into the deliverable hardware configuration items. This testing can be performed at the manufacturing facility (factory acceptance tests) or at the installation site, as dictated by the environmental requirements and test conditions stated in the test procedures. The agency needs to have a presence at factory acceptance tests for new hardware or major hardware components such as a DMS.

4.4.4 Software Build Integration Testing

Software build integration testing is performed on the software components that are combined and integrated into the deliverable computer software configuration items. A software build consisting of multiple items is ideally tested in a separate development environment. This is not always possible due to the expense of duplicate hardware platforms and communications infrastructure. Again, as with software unit testing, test participation at this level is best left to an independent test team with the agency reviewing the test reports.

The agency should give serious consideration to establishing a separate test environment at the TMS facility for future upgrades and new software releases. Every effort should be made to ensure that this test system is representative of the production system and includes simulators or direct connections to the field infrastructure to closely simulate actual system conditions and worst case loading. During the life cycle of the system, especially with incremental deployment, such ongoing testing will be critical. Without a test environment of this type, it is likely that new releases will be accompanied by disruptions in system operation, which may not be acceptable.

4.4.5 Hardware Software Integration Testing

Once the hardware and software configuration items have been integrated into functional chains and subsystems, hardware/software integration testing is performed to exercise and test the hardware/software interfaces and verify the operational functionality in accordance with the specification requirements. Integration testing is performed according to the integration test procedures developed for a specific software release. Testing is typically executed on the operational (production) system unless the development environment is sufficiently robust to support the required interface testing. The agency should make an effort to witness at least some of the hardware/software integration testing especially if performed in the operational environment at your facility. This will be the first opportunity to see specific functionality; e.g., the map software in operation using the large screen display.

4.4.6 Subsystem Acceptance Testing

Acceptance testing at the subsystem and system level is conducted by the acquiring agency to contractually accept that element from the developer, vendor, or contractor. As subsystems are accepted, they may move into an operational demonstration mode, be used for daily operations, or be returned to the developer for further integration with other subsystems depending upon the project structure.

Two activities must be completed in order to commence subsystem acceptance testing. All lower level testing, i.e., unit, installation, and integration testing, should be complete. Additionally, any problems identified at these levels should be corrected and re-tested. Alternatively, the agency may wish to make changes to the requirements and procurement specifications to accept the performance or functionality as built and delivered.

Acceptance testing of the installed software release is performed on the operational system to verify that the requirements for this release have been met in accordance to the system test procedures.

Subsystem test results are recorded and reported via formal system test reports. Formal acceptance is subject to the agency's review and approval of the test report(s) and should be so stated in the procurement specification.

4.4.7 System Acceptance Testing

Acceptance testing at the system level should include an end-to-end or operational readiness test of sufficient duration to verify all operational aspects and functionality under actual operating conditions. While it may not be possible to test all aspects of the required system level functionality in a reasonable period of time, the system test plan should specify which of these requirements must be tested and which should be optionally verified given that those operational circumstances occur during the test period. The acquiring and operating agencies must be ready to accept full operational and maintenance responsibilities (even if some aspects of the operations and maintenance are subcontracted to others). This includes having a trained management, operations and maintenance staff in place <u>prior</u> to the start of the system acceptance testing.

System test results are recorded and reported via a formal system test report. Formal acceptance is subject to the agency's review and approval of the test report and should be so stated in the procurement specification.

The agency may wish to grant conditional acceptance for subsystems that have long-term burn-in periods or specific operational performance requirements in order to allow progress or partial payments to be made, but a sufficient holdback amount (that is relative to the risk being accepted by the acquiring agency), should be retained until all contractual requirements have been met. The amount to be withheld and the conditions for its release must be detailed in the procurement specification. The procurement specification should also include details of any monetary damages or penalties that accrue to the acquiring agency for contractor non-compliance to delivery, installation, and performance requirements.

Subsequent to acceptance testing and formal acceptance, the acquiring agency will own the subsystem or system and will be responsible for its operation and maintenance. The procurement contract should specify what and how (including payment provisions) hardware and software licensing agreements, extended warranties, operations and maintenance agreements, and spares provisioning, are to be handled before conditional or final acceptance is granted by the acquiring agency.

4.4.8 Regression Testing

Additional testing is required whenever new components and/or subsystems are incorporated into the system. These tests are necessary to assure that the added components comply with the procurement, installation, and functional specification requirements and perform as required within the integrated system without degrading the existing capability. A series of regression tests, which are a subset of existing tests, should be performed to re-test the affected functionality and system interface compatibility with the new component. The goal of regression testing is to **economically validate** that a system modification does not **adversely impact** the remainder of the system.

The regression tests assure that the "new" system continues to meet the system performance requirements and provides the required added capabilities and functionality. The regression tests are selected from the test set already conducted on the existing system for the affected interfaces and functionality with procedural adjustments made to accommodate the new components. It is important to regress to a level of tests and associated test procedures that include the entire hardware and software interface. In most cases, this will require some interface and integration testing at the unit testing level in addition to functional testing at the subsystem and total system levels.

The agency should participate in regression testing when a large number of components or a new subsystem is being added or for a major software release that is adding significant new capabilities. In these situations, system operations and performance are likely to be impacted.

Regressions tests are also appropriately applied to software modifications. Their purpose is to identify related functionality that might have been adversely affected by the software modification. In complex systems, a software modification can cause unintended consequences in related subsystems.[8] The regression tests attempt to identify changes to system functionality and behavior prior to moving the software modification into the production system. Regression testing is also important when there are software “upgrades” to the underlying operating system and COTS products. Again, such upgrades can have far reaching consequences if processor loading or memory usage is affected.

In regression testing, the keys are economics and the definition of the system boundaries. System changes can impact narrow or broad functionality and must be assessed on a case-by-case basis. For the regression testing to be economical, it must be tailored to the potential impacts of the system modification. For example a narrow change could result from increasing the maximum cycle length used by a traffic signal controller. Timing plan operation would be regression tested, as would reports involving the signal timing. In contrast, increasing a communications timeout value would require extensive testing due to its impact on the communications infrastructure. This timeout change has the potential to disrupt communications with every field device in the system and might not impact system operations until communications channels are fully loaded at some point in the future. Tailoring regression testing to the apparent risks is appropriate for managing the life-cycle costs of the system.

[8] Note that due to the complexity of today’s systems, the change may have unintended consequences in apparently unrelated portions of the software or system functionality. This can be caused by such problems as processor loading, event sequencing, and data/table overflows that are not apparent. Therefore, regression testing should be broader when the potential for such problems exists.

4.5 Test Identification and Requirements Traceability

The System Test Plan identifies and describes the tests to be performed at Levels 4 and 5 (subsystem and system respectively). Lower level tests (i.e., Level 1 – hardware and software unit/component, Level 2 – software build integration and hardware assembly, and Level 3 – hardware software integration tests) are typically developed and executed by the contractor/integrator providing the hardware and software subsystems.

The system specification VCRM (see appendix A) identifies specific tests and test methods to accomplish verification of the functional and operational requirements of the system. Each test covers a specific set of related and/or dependent requirements as well as partial requirements and should be designed to focus the attention of the test conductor, witnesses, observers and the operations and support personnel on a limited group of functions and operations, thereby minimizing the necessary test resources. Some requirements may require verification activities in more than one test. The detailed test descriptions delineate the requirements and partial requirements specifically covered by a particular verification test (or test case).

A requirement is fully verified by the accomplishment of all tests identified for it in the VCRM and is subject to the approval of test results submitted to the acquiring agency.

4.6 Test Descriptions

Each test description in the system test plan includes the following:

- A test objective.
- Test identification.
- Traceability to the requirements to be verified by the test.
- Test type.
- Level (e.g., demonstration at level 4).
- Test prerequisites and limitations.
- Identification of special test software or equipment needed to accomplish the test.
- Hardware/software test configuration.
- Test responsibilities (e.g., who performs and witnesses the test).
- Data requirements.
- Test scenario.

The test descriptions provide the framework and an outline for the more detailed descriptions that are included in the test procedures. Success criteria are provided on a step-by-step basis in the test procedures.

4.7 Test Requirements and Resources

The test description identifies specific requirements and resources necessary for conducting the test. These include who will conduct the testing and what their responsibilities are prior to, during, and following a test; what the test configuration is; what test equipment will be needed; what the test data requirements are; and what the testing schedule is.

4.7.1 Personnel and Responsibilities

The following paragraphs describe roles and responsibilities of the key system-level test personnel. Depending upon the size of the test program, several roles may be assigned to the same person (small program) or several team members may perform the same role on a dedicated basis (large program).

4.7.1.1. System Test Director

The agency's system test director approves the system test plan, the test procedures developed to accomplish the tests described by the plan, and the test reports provided to formally document the execution of the tests. Subject to a favorable test readiness review, the test director authorizes the use of system resources necessary to conduct the tests as defined by the test descriptions and procedures and has final approval of the test execution schedule.

4.7.1.2. Test Conductor

The test conduct is probably the most important member of the test team. The agency must be comfortable with and trust the individual selected to have the system knowledge and experience level necessary to represent it and fairly conduct the testing. The test conductor directs and oversees the execution of the system tests described by the system test plan. The test conductor conducts the test readiness review and obtains approval from the test director to schedule the test and commit the necessary resources to execute the test. The test conductor provides test briefings and approves any changes to the test configuration and test procedures. The test conductor assigns test witnesses to monitor the execution of the test procedures, record data, and complete checklists. The test conductor may assign limited duties to test observers, e.g., to affirm specific events did or did not occur. The test conductor compiles the test data and prepares the test report for submission to the test director for approval.

4.7.1.3. Test Witnesses

Test witnesses observe, verify, and attest to the execution of each step in their assigned test procedure(s) by completing the checklist(s) provided. Test witnesses also record test data (as required) and comments when appropriate. Test witnesses record the actions and observations reported to them by test observers designated by the test conductor to perform a specific test function. The agency should supply some if not most of the test witnesses. While not totally impartial, as ideal witnesses should be, agency personnel have a stake in assuring the system performs as it is intended to. Contactor personnel have different motivations: wanting to complete the test as quickly as possible and raising the fewest issues.

It is also recommended that there be more than one test witness; often transient behaviors and/or anomalies occur that should be observed but that may be missed by a single observer.

4.7.1.4. Test Observers

Test observers are allowed to observe test activities where their presence does not interfere with the test process. Test observers may also serve a limited role in the test to perform a specific action or to affirm that a specific event did or did not occur. When test observers are requested to perform a specific test activity, they report their actions and observations for the record to a test witness or directly to the test conductor as instructed prior to the test. The agency may wish to use managers as test observers. This affords them the opportunity to become familiar with the test process and take away a first hand knowledge of the details, issues, and time required to implement a TMS.

4.7.1.5. Test Support Personnel

Test support personnel, i.e., the test executors, perform the test steps as directed by the test conductor by operating the test equipment (if any), all system equipment in the test configuration, and the required user interfaces. Test support personnel may be contractor personnel or authorized acquiring or operating agency personnel who have completed qualification training.[9]

4.7.2 Configuration, Equipment, and Data Requirements

Each test procedure should carefully specify the test configuration,[10] test equipment, and data requirements that are necessary for a successful test. The test configuration (i.e., the specific system equipment and/or software under test) must be consistent with and operationally ready to support the functionality of the requirements being verified by the test procedure. Any special test equipment or test software required by the test procedure must have been pre-qualified and (where applicable) have current certifications or calibrations. Data requirements specified in the test procedure, such as database structures, tables, and specific data items, must be in place and complete. For example, in verifying the capability of a dynamic message sign (DMS) to display a message on command, test messages must have been generated and stored and be accessible to the DMS command and control software.

It is important that the test “environment” be reviewed and validated (certified) prior to the start of any testing. This is to ensure that the test environment can and will measure the anomalies and normal operation and that failures of the system under test will be apparent.

4.7.3 Schedule

Test scheduling and coordination may be the most important aspect of testing. It involves a specific knowledge of what components are ready to be tested, how they are to tested and in what environment, what resources (time, material, and personnel) are needed for the testing, and what impacts to other operations can be expected. Once a test has been scheduled, a specific set of resources is committed to be available for a specific period of time. The system test plan should provide an overall test schedule indicating which tests should be performed and in what order. It should also specify the calendar time frame and expected test durations. Actual test dates will be set and coordinated by the test director following a test readiness review.

[9] It is assumed that contractor personnel will have the requisite knowledge and experience to operate the test equipment and system equipment, and follow the test procedures. If agency operations personnel are used, they should have specific operations training on the system functions being tested, otherwise the test steps must be extremely detailed. They also need an understanding of what must be done and the rules of engagement, such as who can touch the equipment, what to record, what to observe, and who to report to.

[10] This should typically include a complete wiring diagram and perhaps a physical diagram where external devices or test stimuli are necessary. Such test configurations must be sufficiently documented that they show all normal connections and potential unintended connections.

4.8 Test Execution

The acquiring agency must formally approve each verification test and the associated technical test procedures prior to test execution. Test pre-planning for the specific test should commence at least 15 days[11] prior to the planned test start date. During this pre-planning period, the test conductor conducts a test readiness review. The readiness review evaluates the test description, current system configuration, test equipment, and training to determine if the required test conditions can be met. This review includes system problem/change requests (SPCR) and other configuration management data that may be applicable. The test conductor makes a determination of which test procedures (if any) must be modified to accommodate differences in the configuration as specified in the test description and the probable test configuration at the time of the test. Test procedures requiring modifications are redlined as appropriate and submitted for approval with the other pertinent test readiness review data. A summary of that readiness review, including any configuration differences, redlined procedures, and a formal test schedule and resources request, is submitted to the acquiring agency for approval. That approval sets the formal test start date and commits system resources and personnel to support the test.

It is recommended that this readiness review be taken seriously by all parties concerned. Testing is expensive for both the agency and the contractor; all too often the contractor “gambles” that they will be ready on a specific date, yet when the consultant and agency personnel arrive, the test environment may be flawed (incapable of performing some of the tests) and the contractor may never have actually attempted to perform the test. This can be a blueprint for disaster and a failed test. Such failures delay the project and inconvenience everyone and start the testing program on a sour note.

Prior to the execution of each test, the test conductor provides a test briefing (orally and/or in writing, as appropriate) to all test participants, including test witnesses, observers, and operations and support personnel. The briefing reviews the test description, including the requirement(s) to be verified, required test configuration, expected duration, and the test responsibilities of each participant. Test procedure checklists are provided to the test witnesses designated to perform a test recording function. Any test limitations, special circumstances, or test configuration and procedure changes are noted at this time. Unless specifically precluded in the test procedures or at the direction of the test conductor, once initiated, the test should be executed to planned completion, even if some individual steps cannot be completed (or fail). A decision to rerun the entire test, or a partial test to complete those steps skipped, will be made after the test conductor terminates the test. Test completion is determined from a review of the test report.

During the execution of the test, designated test witnesses record data and comments as appropriate, and complete each step of the test procedure(s) on the procedure checklists provided. Other data collected during the test (not recorded on the checklists) is identified in the checklist and marked with

[11] This is under ideal conditions. However, when such time frames must be abbreviated due to project schedules, equipment and personnel availability, etc. it is important that the test director and the agency remain comfortable with the test plan. The 15 days is generally necessary to ensure that everyone understands the test and that all resources are available to ensure that the test can be run without interruption.

the test identification number, date, and time collected. Completed checklists and data collected are submitted to the test conductor for inclusion in the test report.

Any conflicts that occur during the execution of a test should be resolved first by the test conductor and then, if necessary, elevated to the test director. Potential sources of conflict include impacts to ongoing operations or other testing; whether or not to allow a series of test steps to be repeated following a failure, if that failure can be attributed to missing a test step, incorrectly following a test step, or executing test steps out of order; and terminating a test in progress due to the failure of test equipment, significant change in test conditions or availability of test resources, or too many test steps having unsuccessful results. The test director, acting on behalf of the agency, is the final authority for resolving testing conflicts.

The test conductor should convene a test debriefing as soon as possible following the termination of the test. Test witnesses and observers are polled for their comments on test conditions, problems, and general observations. A preliminary test completion status is determined from this debriefing to allow for planning of future testing. The agency should have a presence at the test debriefing to get a heads-up on any problems discovered and potential schedule impacts.

If the testing will span multiple days, it is suggested that a test de-briefing be conducted at the conclusion of each day to review notes and identify potential problems and anomalies.

4.8.1 Test Procedures

Detailed technical test procedures are prepared for each system test (see Appendix B for a sample test procedure). A single test may involve multiple procedures, each of which includes a test procedure checklist. This checklist contains the test identification number, enumerated test steps with the expected outcome or response (success criteria) for each step, space (as appropriate) for comments or recording test data, and a place to initial the completion of the step. The original checklist with initials becomes a part of the formal test report. The agency has the responsibility to review and approve system test and acceptance procedures.

It may be convenient to construct an on-going test report using a three-ring binder to which various supporting documents can be added. Examples include strip chart records, calibration charts, schematics, and witness notes. Each page of the test procedure with the observed results and witness initials is also included in the binder as the test is completed. It is also recommended that the agency take pictures of the test environment and results to facilitate later reconstruction of reports. Digital photos are inexpensive to take and the record could be invaluable for later analysis.

4.8.2 Test Tools

Test software, test equipment, hardware/software simulators, data generators (if any), etc. must be pre-qualified,[12] have current certifications or calibrations, and be approved for a specific intended application before use during verification testing. Test procedures must refer to specific test software, if and where applicable, and must include the necessary steps to load, initialize, and operate the test

[12] This must include certification that the test tool or instrumentation can and will capture the specific measurement and that anomalies or failures will be detected or become visible during the test.

software. To insure the repeatability of the test conditions, the test software and operational data including specific "scripts" to be used during verification testing must be under configuration management control prior to test start.

4.8.3 Problems Occurring During Testing

System problem/change requests (SPCR) are written for hardware and/or software that malfunction or fail during the test. Where possible or necessary to continue a test, and at the direction of the test conductor, maintenance support personnel should attempt to restore hardware functionality. At the direction of the test conductor, the software developer or the agency's ITS consultant makes a determination whether or not to attempt a retry of the software function. No software corrections may be made during the test. Data problems (i.e., initial values, thresholds, limits, channel assignments, etc.) may be corrected if necessary to continue the test and meet the test objectives, only if and where provisions have been made to add, edit, or update these data using a standard user interface or data entry screen. Otherwise, the problem should remain uncorrected for the remainder of the test. Data problems must be noted for CM review. No software changes, data or code, should be made during the test using software-debugging tools.[13]

SPCRs written during the test are listed in the formal test report and copies provided as supplemental reports.

4.8.4 Test Reports

The test conductor is responsible for the preparation and submittal of test reports to the acquiring agency for approval. Test reports should be submitted within 15 days of the test execution. A determination of test completion is made by the acquiring agency from a review of the test report.

Formal test reports are prepared for each functional element or system test. As a minimum, the test report must:

- Identify the test being reported by test title and unique test identification number.
- Provide a summary of test activities, including date and time of the test, test witnesses and observers present.
- Provide a brief discussion of any exceptions or anomalies noted during the test.
- Include the original copy of the completed test procedure checklists, and data collected and recorded during the test.

SPCRs written during the test and any supporting analyses conducted on test data collected and recorded should be documented separately and referenced in the test report. Copies of the SPCRs and a copy of each supporting analysis should be provided with the test report.

[13] Note that software debugging tools tend to perturb the operation of the software by altering the timing and machine utilization. As a result, the results may not be representative of normal system operation. Often times, the activation of software debugging tools will have the effect of masking or hiding the problem where timing is suspected.

4.9 Summary

This chapter presented a testing tutorial that included many of the testing concepts and the terminology that you are likely to encounter in your testing program. The five basic verification methods (i.e., inspection, certificate of compliance, analysis, demonstration, and formal test) were defined and applications explained. A multi-level, building block testing approach was introduced that delineated the types of testing that will be conducted and what organization(s) have the primary test responsibility for which tests at each level. Next, each test type was described and linked to its test level. Then the basic elements of the test procedures including test identification and requirements traceability, test descriptions, test requirements, and resources were presented. Finally, what and who is involved in the execution of a test procedure was discussed along with the content of the final test report.

5. Planning a Project Test Program

5.1 Overview

This chapter presents some of the considerations necessary for planning a project test program. As stated previously, several levels of testing are necessary to verify the compliance to all of the system requirements. In the next section, the building block approach to testing is shown and each of the testing levels discussed in detail. Then, the concept of product maturity is introduced, with emphasis on how the maturity of ITS products can affect the level and complexity of testing needed. Lastly, a discussion of how the choice of custom or new products affects the risks to project schedule and costs.

5.2 A Building Block Approach

Figure 5-1 shows the building block approach to testing. At the lowest level (1), hardware and computer software components are verified independently. Receiving inspections, burn-in,[14] and functional checkouts are performed on individual hardware components by the respective vendors and installation contractors as appropriate. A functional checkout is performed for all commercial-off-the-shelf (COTS) software and hardware components, and pre-existing systems or equipment (if any) to be incorporated into the system. This testing is to ensure that the item is operational and performs in accordance with its specifications.

At the next level, software components are combined and integrated into deliverable computer software configuration items that are defined by the contract specification. A software build, typically consisting of multiple configuration items, is loaded and tested together on the development computer system. Hardware components are combined and integrated into deliverable hardware configuration items, also defined by the contract specification. Hardware acceptance tests, which include physical configuration audits and interface checkouts, are performed on the hardware configuration items at

[14] Burn-in is a procedure used to detect and reject products that fail early in their operational life due to otherwise undetected manufacturing process or component defects. The procedure involves exposing the product to the full range of expected operational and environmental conditions for a continuous period of time that exceeds its early failure rate (also known as infant mortality rate). Products that continue to function properly following the burn-in period are accepted for installation and use; those that fail are rejected as unsuitable. The product manufacturer typically conducts the burn-in procedure prior to shipment. However, a large assembly of different components from different manufacturers such as a DMS may require burn-in after site-installation.

the factory, as they are staged at an integration facility,[15] and/or as installed and integrated at their operational locations.

At level 3, the system computer software configuration items and the hardware configuration items are integrated into functional chains and subsystems. The integrated chains and subsystems provide the first opportunity to exercise and test hardware/software interfaces and verify operational functionality in accordance with the specifications. Subsystem integration and subsystem-to-subsystem interface testing is conducted at level 4 to verify that the physical and performance requirements, as stated in the contract specifications, have been met for the complete hardware and software subsystems comprising a functional element.

System integration and test is performed at level 5 to verify interface compatibility, design compliance of the complete system, and to demonstrate the operational readiness of the system. System-level integration and test is the responsibility of the acquiring or operations agency. Final system acceptance is based on review and approval of the test results and formal test reports submitted to the acquiring or operations agency.

The system tests at levels 4 and 5 should be accomplished following successful completion of all relevant lower level testing. Level 4 of the system test follows the successful completion of level 3 tests associated with system configuration; startup and initial operations; and communications network connectivity between the field devices, communication hubs, and the TMC(s). System software functions, relational data base management (RDBMS) functions, and those system security functions associated with access authorization, privileges, and controls are also verified at level 3. Level 4 testing comprises integration and functional tests of the major subsystems and the connectivity between multiple TMCs (if applicable) and associated field equipment to include the functions of the interface and video switching, routing, recording, retrieval, playback, display, and device control (as appropriate). Level 5 testing should include an operational readiness test that will verify the total system operation and the adequacy of the operations and maintenance support training over a continuous period of multiple days. The operational readiness test should include functional tests of the inter-operability with multiple TMCs (if applicable), failover of operations from one TMC to another, and other higher level functionality such as congestion and incident detection and management functions. At each level, system expansion performance should also be verified with the appropriate simulators to ensure that the level of testing is complete and that the system will meet its requirements under fully loaded conditions.

[15] An integration facility typically incorporates a software development environment as well as an area for representative system hardware and software components that have successfully passed unit testing to be integrated and tested prior to site installation.

It should also be noted that the testing program must include infrastructure-induced anomalies and failures such as power interruptions, connection disruption, noisy communications (lost packets, poor throughput), improper operator interactions (which might actually occur with inexperienced users or new users), device failures, etc. These “anomalies” do and will occur over time, and it is important that the system responds as specified (based on the procurement requirements).

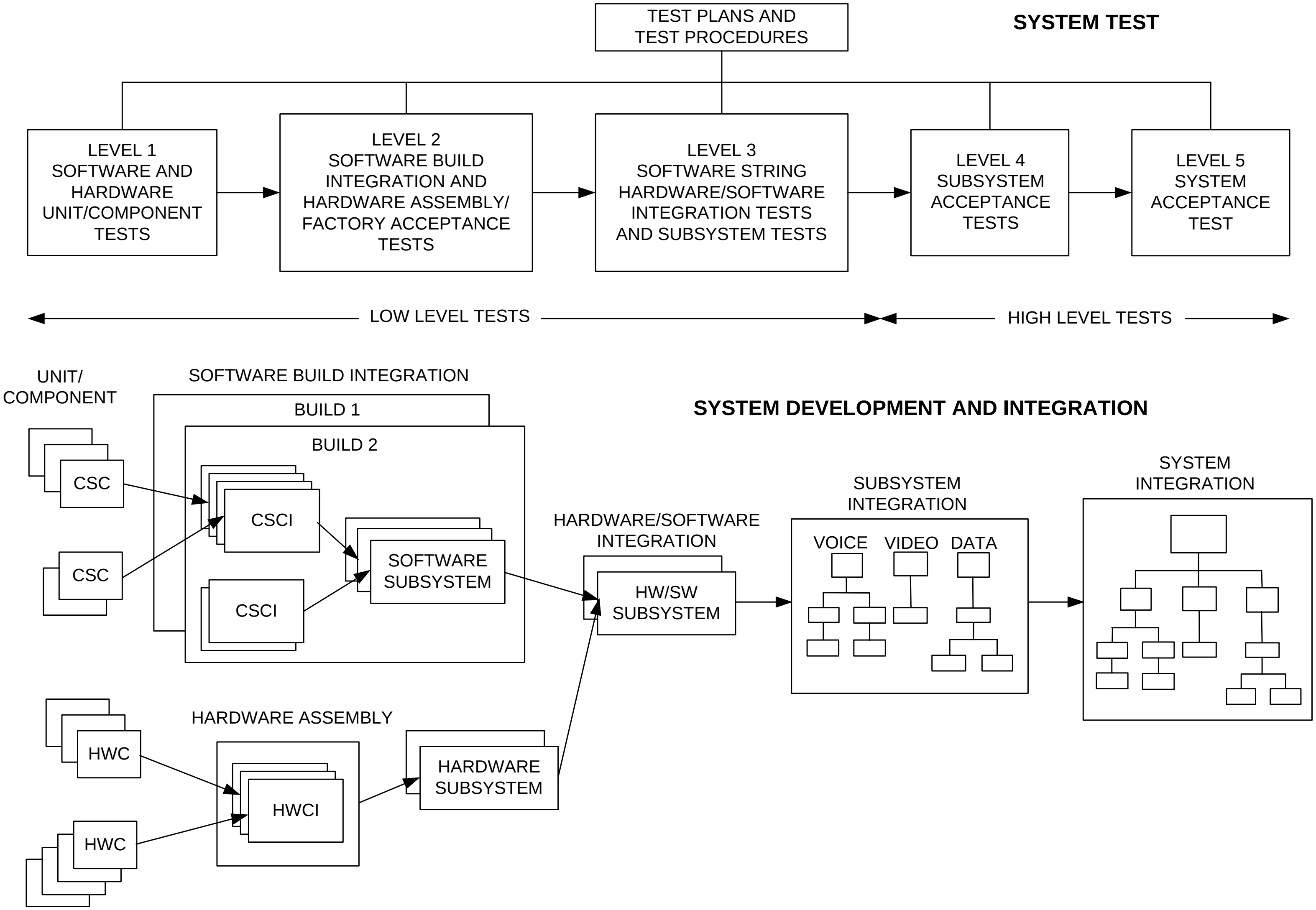

TEST PLANS AND TEST PROCEDURES
SYSTEM TEST
LEVEL 1 SOFTWARE AND HARDWARE UNIT/COMPONENT TESTS
LEVEL 2 SOFTWARE BUILD INTEGRATION AND HARDWARE ASSEMBLY/ FACTORY ACCEPTANCE TESTS
LEVEL 3 SOFTWARE STRING HARDWARE/SOFTWARE INTEGRATION TESTS AND SUBSYSTEM TESTS
LEVEL 4 SUBSYSTEM ACCEPTANCE TESTS
LEVEL 5 SYSTEM ACCEPTANCE TEST
LOW LEVEL TESTS
HIGH LEVEL TESTS
UNIT/ COMPONENT
SOFTWARE BUILD INTEGRATION
BUILD 1
BUILD 2
SYSTEM DEVELOPMENT AND INTEGRATION
CSC
CSC
CSCI
CSCI
SOFTWARE SUBSYSTEM
HARDWARE/SOFTWARE INTEGRATION
HW/SW SUBSYSTEM
SUBSYSTEM INTEGRATION
VOICE
VIDEO
DATA
SYSTEM INTEGRATION
HWC
HWC
HARDWARE ASSEMBLY
HWCI
HARDWARE SUBSYSTEM

Figure 5-1. Building Block Approach to Testing

5.3 The Product Maturity Concept

The maturity of the software and hardware ITS products selected for deployment in your TMS will affect the type and complexity, and therefore the cost, of the testing necessary to verify compliance with your specific project requirements.

The following will establish a definition for the maturity of the ITS product and provide some guidance as to when it should be classified as *standard, modified, or new/custom*. Even though these classifications are very similar as applied to hardware and software products, they are presented separately because of the specific examples given.

5.3.1 Hardware

5.3.1.1. Standard Product

This type of device uses an existing design and the device is already operational in several locations around the country. It is assumed that the hardware design is proven and stable; that is, that general industry experience is positive. Proven, stable hardware design means other users of this same product report that it is generally reliable[16] and meets the functionality and performance listed in the published literature (note that it is recommended that product references be contacted to verify manufacturer claims of proven reliability).

This device may be from the manufacturer's standard product line or it may be a standard product that has previously been customized for the same application. The key factor is that the device has been fully designed, constructed, deployed, and proven elsewhere prior to being acquired for this specific project. The product will be used "as-is" and without any further modification. It is assumed that the user is able to "touch," "use," or "see" a sample of the device and is able to contact current users of the device for feedback on the operation and reliability of the device.

It is important that the environmental and operational use of the device is similar in nature to the intended operation; devices with experience in North Dakota may have problems when deployed in Arizona, or devices generally installed on leased, analog telephone lines may experience a different result when connected to a wireless or other IP network. It is also assumed that the device is being purchased or deployed without design or functional changes to its operation (i.e., no firmware changes). Examples include typical traffic controllers, loop detectors, CCTV equipment, video projectors, workstations, monitors, and telephone systems. Such devices have typically been subjected to NEMA TS2 (or the older TS1) testing specifications or have been placed on a Qualified Products List (QPL) for some other State (e.g., CALTRANS, Florida DOT). In general, the acquiring agency should be willing to accept the previous unit test results, providing they were properly done and are representative of expected operating conditions. The agency can then concentrate on delivery inspection and functional tests and installation testing of these standard products. If the

[16] This is not quantified and it is difficult to put a number on "reliability" for a typical ITS device. For traffic controllers, the general history has been a 1-2 percent DOA rate for initial installation and typically between 4 percent and 8 percent failure rate per year. While these are non-scientific numbers, this has been the general experience for these types of devices.

agency is in doubt, it is recommended that the test results from previous tests be requested and reviewed to verify the above.

5.3.1.2. Modified Standard Product

This type of device is based on an existing device that is being modified to meet specific functional and/or environmental or mechanical requirements for this procurement. It is assumed that the modifications are relatively *minor* in nature, although that is a judgment call and difficult to quantify.

Several examples of minor changes and their potential implications should clarify why these changes are difficult to quantify. A vendor may increase the number of pixel columns in a dynamic message sign. This change could impact overall sign weight, power consumption, and reliability, therefore affecting other sign systems. A DMS vendor may change the physical attributes of the design — perhaps the module mounting — but keep the electronics and other components the same. This change can also impact sign weight and mounting requirements. A DMS vendor may install the same electronics onto a new circuit board layout. This change may impact power consumption, heat generation, and electrical “noise” within the DMS. In all cases, in order to be classified as a *modified* standard product, the base product being modified meets the criteria for a *standard* product as described above. Although the changes are minor, they may create “ripple effects” that impact other design characteristics.

The type of testing that the unit should be subjected to will vary depending on the nature of the modifications. Mechanical changes may necessitate a repeat of the vibration and shock testing; electrical or electronic changes may require a complete repeat of the environmental and voltage testing; and, functional changes such as new display features (e.g., support for graphics on a DMS) may necessitate a complete repeat of the functional testing (but not a repeat of the environmental testing). Members of the design team should work closely with the test team to build an appropriate test suite to economically address the risk associated with the design modifications.

5.3.1.3. New or Custom Product

This type of device is likely to be developed and fabricated to meet the specific project requirements. Note that “new” may also mean a first-time product offering by a well-known vendor, or that a vendor is entering the market for the first time.

The device’s design is likely based on a pre-existing design or technology base, but its physical, electrical, electronic, and functional characteristics have been *significantly* altered to meet the specific project requirements. Examples might include custom ramp controllers, special telemetry adapters, and traffic controllers in an all new cabinet or configuration. It is assumed that the design has ***not*** been installed and proven in other installations or that this may be the first deployment for this product.

5.3.2 Software

5.3.2.1. Standard Product (COTS)

Standard product or commercial-off-the shelf (COTS) software has a design that is proven and stable. The general deployment history and industry experience has been positive and those references contacted that are currently using the same product in the same or a similar environment find that it is generally reliable and has met the functionality and performance listed in the published literature.

It may be a custom, semi-custom or a developer's standard software product, but it has been fully designed, coded, deployed, and proven elsewhere prior to being proposed for this specific project. It is assumed that the user is able to "use" or "see" a demonstration of the product and is able to contact current users of the product for feedback on its operation and reliability.

Operating systems, relational databases, and geographical information systems software are examples of standard products. ATMS software from some vendors may also fall under this category as well.

5.3.2.2. Modified Standard Product

This type of software is based on an existing standard product that is being modified to meet specific functional and/or operational requirements for this procurement. It is assumed that the modifications are relatively minor in nature, although all differences should be carefully reviewed to determine the extent of the differences and whether the basic design or architecture was affected.

Examples of minor changes include modifying the color and text font style on a GUI control screen or changing the description of an event printed by the system. Sometimes the nature of the change can appear minor but have significant effects on the entire system. For example, consider a change to the maximum number of DMS messages that can be stored in the database. Increasing that message limit may be considered a major change since it could impact the message database design and structure, message storage allocation, and performance times for retrieving messages. On the other hand, reducing that limit may only affect the limit test on the message index. In all of these cases, in order to be classified as a *modified* standard product, the base product must meet the criteria for a *standard* product as described above.

The type of testing that this class of products should be subjected to will vary depending on the nature of the modifications. Changes to communication channel configurations to add a new channel type or a device on an existing channel may necessitate a repeat of the communication interface testing; functional changes such as new GUI display features (e.g., support for graphics characters on a DMS) may necessitate a repeat of the functional testing for the DMS command and control GUI. Members of the design team should work closely with the test team to build an appropriate test suite to economically address the risk associated with the design modifications.

5.3.2.3. New or Custom Product

This class of software is a new design or a highly customized standard product developed to meet the specific project requirements. Note that this may also be a new product offering by a well-known applications software developer or ITS device vendor, or a new vendor who decides to supply this software product for the first time. The vendor is likely to work from an existing design or technology base, but will be *significantly* altering the functional design and operation features to meet the specific project requirements. Examples might include: custom map display software to show traffic management devices and incident locations that the user can interactively select to manage, special routing, porting to a new platform, large screen display and distribution of graphics and surveillance video, and incident and congestion detection and management algorithms. It is assumed that the design has ***not*** been developed or deployed and proven in other installations or that this may be the first deployment for a new product.

For software, agencies are cautioned that the distinction between "modified" and "new" can be cloudy. Many integrators have claimed that this is a "simple port" to re-implementation with a few changes, only to find that man years had to be devoted to the changes and testing program that resulted in the modified software.

5.4 Risk

The risk to project schedule and cost increases when utilizing custom or new products in a system. There are more “unknowns” associated with custom and new products simply because they have not been proven in an actual application or operational environment, or there is no extensive track record to indicate either a positive or negative outcome for their use or effectiveness. What’s unknown is the product’s performance under the full range of expected operating and environmental conditions and its reliability and maintainability. While testing can reduce these unknowns to acceptable levels, that testing may prove to be prohibitively expensive or disproportionately time consuming. Accepting a custom or new product without appropriate testing can also result in added project costs and implementation delays when these products must be re-designed or replaced. These unknowns are real and translate directly into project cost and schedule risk. It is imperative that the design and test teams work together to produce test suites that economically address the risk associated with the new or custom designs.

When planning the project, risk areas should be identified early and contingency plans should be formulated to address the risks proportionately. Risk assessments should be performed that qualitatively assess the probability of the risk and the costs associated with implementing remediation measures. The remediation measures can include a wide range of measures including increased testing cycles, additional oversight and review meetings, more frequent coordination meetings, etc.

After identifying the risks and remediation costs, the project funding and schedules can be adjusted to provide a ***reasonable*** level of contingency for the risks. Risk is managed and not eliminated by shifting it to another organization. Costs will follow the shifted risks; however, some level of management costs will remain with the acquiring agency.

5.5 Summary

This chapter has described the building block approach to testing and the testing level in a way that should help in planning a project test plan. It introduced the product maturity concept and how the maturity of products selected for use in your TMS affect what and how testing must be done. These concepts directly impact test planning. Lastly, this section considered how the risk to project schedule and cost are affected by the maturity of the products and their testing program requirements.

6. Hardware Testing

6.1 Overview

This chapter focuses on the testing of hardware or physical devices including traffic controllers, detection systems, ramp controllers, and dynamic message signs and TMC devices such as workstations, video projectors, and communications equipment. While *software* is usually part of these devices since most include computers (microprocessors), it is typically "embedded" (classified as *firmware*) or an integral part of the device. The hardware test program is intended to cover the device testing from the device testing from prototype to final deployment.

Throughout this chapter, the various phases of testing are presented, from *prototype* testing during early phases of the product development through *site testing,* which occurs once the equipment is installed at its final location. The degree of testing required will depend on the maturity and track record or installation history of the device (product), the number of devices purchased, the cost of the testing, and the risk of system failure caused by problems with the device. For general classification purposes, the maturity of the device will be categorized based on its history, which can vary from *standard* devices (typically standard product), to *modified* devices, to *new or custom* devices developed for a specific deployment.

The following sections will discuss the device testing in phases starting with the development of a prototype to the final acceptance testing. In the installation phases and beyond, the test activities are the same regardless of the maturity of the product (new, existing, or modified).

6.2 What Types of Testing Should Be Considered?

The testing for a device or product can be broken down into the following general categories:

- Design verification.
- Functionality.
- Mechanical and construction.
- Standards compliance (NTCIP and others).
- Environmental.
- Serviceability.

Each of these will be discussed to gain a better understanding of what is meant and what is required for each.

The following sections describe the elements of a complete testing program based on the assumption that the device being offered is a new design or custom product, and hence the device should be subjected to all aspects of requirements verification. After this initial discussion of the worst case testing program, this guide will consider what steps can probably be eliminated or minimized for *standard* products and *modified* products as described above.

6.2.1 Design Verification

Most procurement specifications will include design requirements for the ITS devices. If these requirements are not explicitly included in the procurement specifications, they may be invoked through referenced standards such as the CALTRANS Traffic Engineering Electrical Specification (TEES) or the equivalent NY State Standards.[17] These requirements typically include such physical issues as choice of hardware (see mechanical and construction below), choice of materials, voltage margins, visibility of indicators, speed of devices, and component thermal restrictions. These design requirements may also include limitations on the mounting of electronic components, insertion and removal force, connector plating, labeling of electronic components, printed circuit board layout markings, and custom components. The agency is cautioned that re-use of "existing" procurement specifications can often lead to references that may be outdated or obsolete, such as retired military, or "MIL," standards. In fact, if you use MIL standards, the question of whether your agency is capable of actually performing tests to verify conformance to MIL standards must be asked. If you can't or don't intend to actually test for compliance with these standards, don't put them in the specifications unless you are willing to accept a certificate of compliance from the vendor for these tests.

There may also be outdated restrictions on electronic construction techniques, such as a prohibition of multi-layer printed circuit boards, and requirements for integrated circuit sockets that are no longer valid and would prohibit many of today's newer technologies. It is important that the procurement specifications be reviewed and updated by a person who is knowledgeable of current electronic designs and construction methods to ensure that all external references are valid and current and that manufacturing technology has likewise been updated to reflect current techniques. Because of the specialized skills required, most agencies and many consulting firms will need to supplement their staff by outsourcing this work. When an agency engages a consultant to prepare their procurement specification, how and by whom (e.g., sub-consultant or on-staff engineer) this expertise will be provided should be well defined.

As a case in point, the following are "design requirements" typically found in ITS device procurement specifications and therefore must be verified for product acceptance by either the agency or its consultant. Note in example requirement 1.3.2.3 below, it may be difficult or impossible to read the manufacturing dates on all the PC board components unless they are physically removed and inspected under a magnifying glass or microscope. However, this requirement could be verified during the manufacturing process, before the components are inserted and wave soldered on the PC boards.

[17] Many states other than California and New York have developed or adopted similar standards. These, however, are the ones most often cited in the ITS industry.

REAL WORLD EXAMPLE: (taken from CALTRANS TEES August 2002[18]).

1.3.2.3

No component shall be provided where the manufactured date is 3 years older than the contract award date. The design life of all components, operating for 24 hours a day and operating in their circuit application, shall be 10 years or longer.[19]

It is recommended that these types of design requirements be validated with respect to the rationale behind why they have been included in the specification (e.g., what value do they add to the product's reliability, electromagnetic compatibility, etc.) and what methods will be used to verify them. Requiring vendor certification that the design meets these requirements and a performance bond with an extended full replacement warranty on the entire device might accomplish the same objective without the agency bearing the cost of extensive design verification testing. For new or custom devices, the device design must be reviewed for conformance to these types of requirements and this type of inspection will require specialized technical expertise to review design drawings, data sheets, etc.

The goal of the design review is to examine areas of the design that may be subjected to electrical, mechanical, or thermal stress. Several areas of a device's electronic design typically warrant close design review; these include the power supply design and the external interfaces for voltage and power dissipation. These areas of a design are often subjected to the most stress due to AC power line regulation and the characteristics of the external devices. Design short cuts in these circuits can affect the long-term reliability of the device. It is also necessary to review the manufacturer's data sheets to ensure that the components being provided are truly rated for operation over the NEMA or TEES temperature ranges; this is often the difference between "commercial" and "industrial" rated components. It is common to include a requirement that non component shall be used in a manner which is inconsistent with the manufacturer's recommendations without explicit written information from the manufacturer stating that the vendor's use is acceptable.

As with the environmental testing (below), it is important that the specifications identify the design requirements and that the test (and inspection) procedure include verification of these requirements. The vendor should be required to assemble a set of manufacturer data sheets for all components and have those included in the reference material provided as part of the test procedure. Often the best approach to this aspect of the "testing" (inspection) of the product is to require that the vendor provide engineering work sheets that show the thermal characteristics of the device, demonstrate how the

[18] The California Department of Transportation (CALTRANS) published a standard for Transportation Equipment Electrical Specifications (TEES); this is generally available on the CALTRANS web site: http://www.dot.ca.gov/hq/esc/ttsb/electrical/2070c.htm for the 2070. It is likely that this web link may change over time; it is suggested that a search engine be used with the search criteria of "TEES" "CALTRANS" and that the current version be located in this manner.

[19] This may be very difficult to verify; one needs to review the expected life of the components and ensure that all devices used are rated for continuous 24 hours per day 7 days a week operation over a minimum of 10 years without replacement. Components such as fans – which may be extensively used in a dynamic message sign, must be rated for such continuous operation; typically the vendor is required to show that they have met this requirement by presenting component information to verify the expected life.

components are maintained within their junction temperatures for all ambient temperatures required, and how the voltage ratings of the interface devices were determined.

Experience has shown that when vendors are aware that the device will be inspected for conformance to all of the design requirements, they will revisit their design decisions and/or start the process of submitting a request for an exception. Often, by requiring that the vendor develop the test procedures and inspection "check sheets," they will address potential problems before the testing.

> EXAMPLE: During the development of custom telemetry equipment for the Manhattan Signal System, the vendor was required to develop a test procedure that included full and complete verification of all of the design requirements in addition to all of the functional and performance requirements listed in the specifications. Initial test procedures delivered to the agency did not include inspection for the design requirements. When the agency insisted that "all aspects of the specification requirements be demonstrated," the vendor conducted their internal review, which revealed that their power supply design and interface circuitry would not meet the specified requirements. The vendor modified the design and submitted proof that the design met the requirements, and the units have provided long-term (> 15 years) reliable operation.

> EXAMPLE TEST STEPS TAKEN FROM A DMS FACTORY INSPECTION TEST PROCEDURE: This test procedure included a complete matrix of all of specification requirements and a method used to verify all such requirements. Only two of the requirements are shown here, but the test procedure included design review of the power supplies, driver circuits, and such details as component mounting and printed circuit board labeling.

Specification Section	**Requirement**	**Verification method**
4.1.10	Encapsulation of two or more discrete components into circuit modules is prohibited, except for opto-isolators, silicon controlled rectifiers, transient suppression, circuits, resistor networks, diode arrays, and transistor arrays. Encapsulated assemblies shall be second sourced standard items.	Process Visual Inspection
4.1.11	Except as specified above, all discrete components, such as resistors, capacitors, diodes, transistors, optical isolators, triacs, and integrated circuits shall be individually replaceable.	Process Visual Inspection

> Note: possible verification methods include - inspection, certificate of compliance, analysis, demonstration, and test.

6.2.2 Electrical, Mechanical and Construction Verification

Different devices will have different mechanical and construction requirements. This aspect of the testing program should include conformance to mechanical requirements such as weight, height, depth, and width, where specified. For example, for dynamic message signs, it is important that the vendor's design adhere to the specifications for these parameters because they directly affect the design of the structure and its installation (assuming that the structure was properly designed). For other ITS devices such as traffic controllers or field cabinets, it may be essential that they match an existing foundation size, mounting footprint, or existing cabinet. For some devices such as detectors, communications equipment, and controllers, it may be shelf limitations, rack height and depth, or printed circuit card profile.

Aspects of the mechanical design and construction must also be inspected or tested to ensure that the welding is proper, that the paint adhesion, thickness, harness, and color are proper, and that the material was properly treated before assembly. Some agencies have developed specific test procedures for parameters such as paint hardness (e.g., writing with a HB pencil which must not leave a mark) and paint color (the agency's specified color samples are compared to the painted cabinet). Some parameters (e.g., paint thickness) require that the thickness of the paint be measured on a cross section of a sample. In summary, although some of these requirements can be observed, many may require certification by the vendor and an inspection or analysis by a third party laboratory.

Other construction requirements may require inspecting for all stainless steel hardware, prohibitions on sheet metal screws and pop-rivets, and specific requirements for wire protection against chaffing and abrasion, and the use of wire harnessing, terminal block types, wire terminations, wire labeling, and wire colors and gauges. A design and construction checklist developed from these specification requirements (and other invoked standards) should be used during this aspect of testing and inspection for compliance verification.

The procurement specifications should specify which party develops the checklist. Typically, the vendor creates this checklist for review and approval by the agency. The agency in turn, must verify that all of the requirements identified in the procurement specifications (and invoked standards[20]) are included in the checklist. The advantage of requiring that the vendor develop the checklist is that as the vendor develops the list, they are forced to review the details of the specifications and address potential areas of non-conformance before the formal factory testing.   There have been instances where vendors have discovered areas of their own non-compliance during the development of this procedure; they can then work with the agency to accept the deviation or alter their design without the impending failure of a factory acceptance test. Sometimes, such deviations are inconsequential in nature and may reflect improvements in the overall product design. By reviewing the issues before the formal test, both the vendor and the agency are able to identify a workable solution without the pressures often associated with a factory acceptance test.

For design requirements such as cabinet doors, one needs to inspect the latches and locks, the gasket material, and the adhesion of same. If the specifications for the cabinet include specific airflow requirements, the vendor should be required to show that the design of the fan, filter, or louver systems are sufficient to provide the required airflow. This must generally be done through airflow calculations (analysis) based on the openings, filter, and fan characteristics. Associated components and design characteristics such as the thermostat, fan bearings, fastener and filter types used, component locations and accessibility for replacement and maintenance, and component labeling should also be inspected.

[20] Note that most procurement specifications will also invoke NEMA TS2-2004, TS4, CALTRANS TEES or other recognized standards for the ITS device. When developing a checklist or inspecting the device for conformance, these standards must also be considered and the various requirements of those standards must be included in the checklists used for the inspections/testing.

Verification of water leakage (entry) requirements will generally be a combination of inspection and actual water testing. To test for water leakage, the entire cabinet (e.g., controller, DMS, etc.) should be subjected to water flow based on the specification requirements that reflect the expected environmental conditions and maintenance activities at the installation site. This should include driving rain on all surfaces as a minimum and high pressure washing of sign faces. While not quantitative in nature, some agencies simply subject the cabinet (or sign) to the water spray from a typical garden hose held at a 45-degree angle from the ground. This test practice may not be sufficient and does not reflect the real world environment. A specific test designed to verify the expected environmental conditions and maintenance activities should be used. For example, cabinets supplied to meet CALTRANS TEES specification requirements are tested by subjecting them to the spray from an irrigation sprinkler of a specific type with a specific volume of water. When performing such a test, white newspaper (the type used for packing) can be placed into the cabinet and then inspected for signs of water after the test. Cabinet inspections should include design characteristics such as how various joints are constructed and sealed. The inspection should consider what happens as gaskets age or if the gasket material is damaged. Examine the cabinet design to determine what happens when (not if) water does enter around the gasket area or through the louvers. A good design will anticipate this life cycle problem and will ensure that the mechanical design is such that any water entering the cabinet is safely managed so that it does not damage any of the components or compromise the integrity, operation, or utility of the device.

As noted earlier, the testing procedure can only verify the requirements documented in the specification. In order to require that the vendor conduct these tests and inspections, the procurement specification must address all of these issues in either an explicit manner (e.g., “all hardware shall be stainless steel”) or in a functional manner (e.g., “the device shall ensure that as the gasket ages and the product is serviced, water cannot cause damage or improper operation to the device or its components”). The requirements should be explicit and quantifiable such that verification by one of the test methods (inspection, certificate of compliance, demonstration or test) is not subject to interpretation. In the above example the requirement phrase “water cannot cause damage or improper operation” is subjective and not easily verified – in general negative statements in requirements should be avoided, they are difficult or impossible to verify. This example’s requirements language should be replace by a positive statement like “cabinet drainage holes shall be provided to allow for water that intrudes into the interior of the cabinet to self drain; interior components shall be mounted at least 2 inches above the bottom of the cabinet; and all electrical connections shall be covered with water spray shield.”

While you can’t make all of your tests totally objective, you have to be careful how you deal with things that could be judged as subjective. In particular, there should be a stated methodology in the procurement specification for how the agency intends to resolve such conflicts. For example, “should a conflict arise with respect to satisfaction of any requirement that may be subject to interpretation, the agency shall have the right to accept or reject the vendor’s interpretation and test results offered as proof of compliance, and shall provide a requirement clarification and/or set specific criteria for compliance for a re-test.” This type of statement in a procurement specification would serve notice to vendors that they need to thoroughly review the specification requirements and ensure that the requirements are clear, unambiguous, and not subject to interpretation. Any requirements that don’t meet this standard should be questioned and clarified in the final procurement documents. This is an area where careful document review before the bid will lead to pre-bid questions for clarification. Then all bidders will understand the intent and intended testing that will be performed.

Some of the requirements may need more explicit mechanical design review to determine if the structural integrity of the product is sufficient (e.g., design of a large walk-in dynamic message sign) and that the welding meets American Welding Society (AWS) standards. This may require that a welding inspection firm be hired to x-ray and verify the welds. For a less costly approach, the agency could require that the vendor provide certification that the welders, who actually constructed the device, have been properly trained and certified by the AWS and that such certification is up to date. An outside inspection firm could be hired to inspect some of the more critical welds and those made at the installation site (if any) to provide added confidence. For installation of an over-the-road device, the agency should request that a State licensed structural engineer be responsible for and oversee the design and seal all structural documents.

6.2.3 Environmental

Environmental testing verifies that the product operates properly under the field conditions of the expected installation site and typically includes temperature, humidity, vibration, shock, and electrical variations. This aspect of testing is usually the most extensive and complex required for any product.

There are a number of industry-accepted references for the environmental and electrical requirements; these include (as examples) NEMA TS2 (and TS4 for DMS), the CALTRANS TEES document and the NY State controller specifications.

All of these documents provide guidelines for temperature and humidity, vibration and shock, and power interruptions, voltage transients, and power line voltages during which the equipment must operate properly. Vendors typically develop a test plan that includes testing performed by an independent test lab based on the NEMA testing profile.

A portion of the test profile in the NEMA TS2 Standard[21] (see figures 6-1 & 6-2) includes a temperature and humidity time profile that lowers

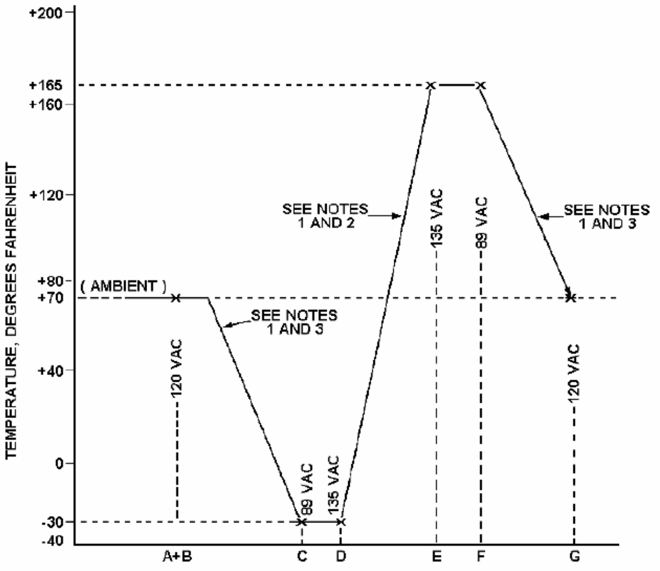


Figure 2-1
TEST PROFILE

NOTES:

1. The rate of change in temperature shall not exceed 17°C (30°F) per hour.
2. Humidity controls shall be set in conformance with the humidities given in Table 2-1 during the temperature change between Test D and Test E.
3. If a change in both voltage and temperature are required for the next test, the voltage shall be selected prior to the temperature change.

Figure 6-1 NEMA Temperature Profile

[21] Table 2-1 and figure 2-1 shown here are taken with permission from the NEMA TS2-2003 standard for Traffic Controller Assemblies with NTCIP Requirements, Version 02.06 – contract www.nema.org to purchase a full copy of this standard.

the temperature to –30° F and then raises the temperature to +165° F.

Detailed operational testing is performed at room temperature, low temperature, and high temperature with varying line voltage and power interruptions. Vendors typically subject only a single unit to the testing profile, and the unit is often not continuously monitored; as a result, failures caused by thermal transients during temperature transitions can go undetected. Further, the shock and vibration testing should be performed before the functional and environmental testing. Performing the environmental testing after the shock and vibration testing should reveal any previously undetected problems due to intermittent connections or components that may have suffered damaged as a resulted of the mechanical testing.

For the environmental testing, it is recommended that the procurement specification require the vendor to develop the test plan with references to the specific environmental requirements to be verified and submit this test plan to the agency for review and approval. The rational for this approach is that the vendor can then develop the test plan based on their available instrumentation and resources. Review of the test plan and associated test procedures is extremely important. A proper review of the test plan requires that the agency (or their representative with the appropriate technical expertise) compare the specifications and additional standards with the proposed test procedures to ensure that all of the requirements are verified. Such a test plan should be submitted well in advance of the planned testing date, and it is recommended that agency personnel and/or their representatives observe the testing program.

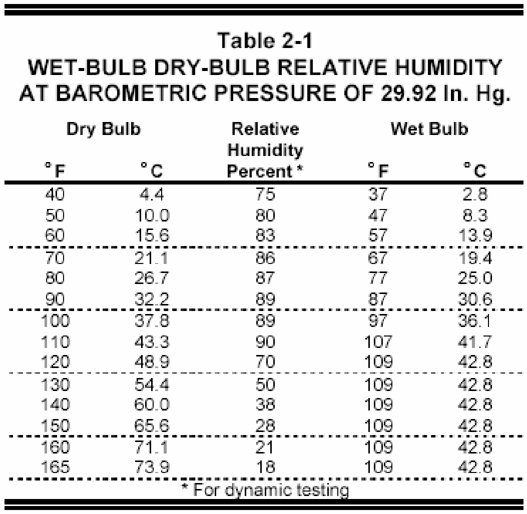


Table 2-1
WET-BULB DRY-BULB RELATIVE HUMIDITY
AT BAROMETRIC PRESSURE OF 29.92 In. Hg.

Dry Bulb		Relative Humidity	Wet Bulb	
°F	°C	Percent *	°F	°C
40	4.4	75	37	2.8
50	10.0	80	47	8.3
60	15.6	83	57	13.9
70	21.1	86	67	19.4
80	26.7	87	77	25.0
90	32.2	89	87	30.6
100	37.8	89	97	36.1
110	43.3	90	107	41.7
120	48.9	70	109	42.8
130	54.4	50	109	42.8
140	60.0	38	109	42.8
150	65.6	28	109	42.8
160	71.1	21	109	42.8
165	73.9	18	109	42.8

* For dynamic testing

Figure 6-2. NEMA Relative Humidity Profile

The environmental test configuration should include a means to continuously monitor and record the operation of the device under test (DUT). The vendor should be required to develop simulators and monitoring interfaces that will continuously exercise the unit's inputs and monitor all of the unit's outputs. For devices such as traffic controllers, ramp meters, and detector monitoring stations, it is essential that strip chart recorders or similar devices be used to continuously record the operation and that inputs are "stimulated" in a known manner to verify monitoring and data collection calculations. For devices such as DMS, new messages need to be sent to the sign, and pixel testing should be periodically requested. All results should be logged. For all ITS devices, a simulated central computer system needs to continuously (at least once per minute) interrogate the device and verify the proper responses. If the device is intended to support once-per-second communications, then the central simulator should interrogate the device at that rate. All of the device inputs and outputs (e.g., auxiliary functions) must be included in the testing; where measurements are required (e.g., speed traps), the simulator must be able to provide precise inputs to the DUT to verify proper interface timing and calculations.

The test plan must include provisions to verify the test configuration before starting the test. To accomplish this, the test procedure must be reviewed to determine whether all the test conditions can be met and that the appropriate test tools (including software), test equipment, and other resources that will be used to accomplish the test are available and ready to support the test. If the test configuration cannot support the testing requirements for observing, measuring and/or recording expected results as detailed in the test procedure, then the test configuration cannot be verified and the test should not be attempted.

Constructing and configuring a test environment that establishes the appropriate set of test conditions, test stimulus, and measuring and recording equipment while remaining unaffected by the DUT can be difficult. For example, a rapid power interruption and restoration test procedure may cause the DUT to fail, shorting out its power supply and blowing a fuse on the power source side. If the test environment and DUT are fed from the same fused source, the test instrumentation and simulation equipment will also lose power and can't record the event or subsequent effects. Independent power feeds would prevent this problem and expedite testing. When reviewing test procedures, the accepting agency should pay careful attention to what is defined for the test environment and how it is isolated from the DUT.

The environmental test plan must include, as a minimum, the following elements:

- A complete description of the test environment including a diagram showing all wiring and instrumentation, the location of all equipment, etc.

- A detailed description of the techniques that will be used to measure the performance of the DUT. The test procedure should also include verification of calibration certificates for all test equipment used to measure or control temperature, voltage, vibration, shock, and timing (frequency, spectrum, etc.).

- A complete step-by-step procedure (test scenarios) showing how each requirement listed in the specifications (and auxiliary standards which may be invoked) will be verified. For any measurement or printed result, the test procedure should indicate the expected (correct) result; any other result is classified as an error.

There are other requirements for the test procedure that will be discussed later; what the reader should understand is that a simple "follow the NEMA testing profile" is not sufficient. It is up to the vendor to determine how to demonstrate proper operation and how the test will be conducted and to show each step that will be taken to verify the requirement. It is up to the agency to review this material to ensure that the testing is thorough, fully verifies the requirements of the specifications, and, at a minimum, is representative of the extreme field conditions expected. Note that it was the responsibility of the specification writer to ensure that the requirements stated in the procurement specifications (and invoked standards) are representative of the field conditions; if the temperature is expected to be colder than –30° F, but the specification only mandated operation to -30° F, it is not reasonable to require that the vendor test to –50° F. If this is a real requirement, it should have been included in the specifications.

> REAL WORLD EXAMPLE: Note the following are examples of environmental requirements that exceed NEMA requirements; if the vendor only tested to the NEMA standard, it is likely that the product was not subjected to testing for these requirements; therefore, additional testing will be required even if the product has been previously tested to the NEMA standard. Examples of additional requirements include:
>
> - "All equipment shall be capable of normal operation following rapid opening and closing of electromechanical contacts in series with the applied line voltage for any

number of occurrences. Line voltage shall mean any line voltage over which the unit is required to function properly."

- "… moisture shall be caused to condense on the EQUIPMENT by allowing it to warm up to room temperature [from –30° F] in an atmosphere having relative humidity of at least 40 percent. The equipment shall be satisfactorily operated for two hours under this condition. Operation shall be successfully demonstrated at the nominal voltage of 115 volts and at the voltage extremes …"
- "The LCD display shall be fully operable over the temperature range of –10°F to +165°F. Fully operable shall be defined to mean that the display shall respond fast enough to allow clear readability of data changing at a rate of once per second."[22]

In this case, the test procedure must be expanded to show that the equipment will meet these requirements.

6.2.4 Functionality

Functionality testing verifies that the device performs all of the specified operations listed in the requirements. Examples of *operational* requirements include the number of plans supported by a ramp controller, the number of events in a scheduler, the number of messages for a DMS, the number of fonts for a DMS, accuracy of speed detection algorithms, etc. *Functionality* includes such capabilities as display a message, change a timing plan, detect a failure, calculate ramp metering rates, and collect data.

Functionality testing will be extensive, and it is likely to be incomplete when one examines the complete "tree" of all possible combinations of operational situations. As an example, it is virtually impossible to test for all combinations of timing plan parameters (e.g., cycle, splits, offset), output configurations, phase sequences, communications anomalies, and preemption conditions for a traffic controller. Likewise for a DMS, it is very time consuming to test for all possibilities of animation, fonts, text, special characters, graphics, communications anomalies, message sequencing, and timing. Under these circumstances, one must weigh and manage the risk of having an undiscovered "bug" with the time and budget available and the likelihood that the specific combination will ever be experienced during operation of the device.

When attempting to determine what testing is important – one might consider some of the following:

1. What happens when the communications is disrupted and restored?
2. What happens under a fully loaded Ethernet connection? [Warning, 2070 traffic controllers have a problem with this.]

[22] Note that most standard ITS devices with LCD displays do not meet this requirement since the basic standards do require support for these extremes; this requirement means that the vendor must add heaters and heater control circuitry to their product. However, if field maintenance under these conditions is expected, then such a requirement should be considered.

3. How does the device recover from power outages of various types?

4. Does it support the correct number of plans, events, fonts, etc.? This is generally checked at the limit conditions (i.e., plan 31, message 63, etc.) and should also be checked to see if it rejects a request for something outside the limits (e.g., 0 and limit +1).

5. Does the device keep proper time; i.e., does it meet the timing accuracy and drift requirements of the specifications (see 6.2.5). Does it properly deal with the daylight savings time transitions?

6. For a dynamic message sign, check basic text rendering, justification, character sizes, flashing timing, multi-phase message timing, scheduler operation, status monitoring, error detection, communications response times (assuming they were specified), and error handling for messages that are too long, improperly formulated, etc.

7. For a traffic controller, check for basic operation, phase sequencing, plan transitions, event scheduling, preemption, and detector processing. For a traffic controller, it is likely that the agency has a specific subset of the overall functionality that is critical to its operation; the testing should be structured to test for those specific combinations of operation and features.

8. There are a number of deployment issues that need to be addressed such as: Will there be three thousand of the devices deployed in the field or a dozen? Are the devices easy to access (e.g., 10 intersections along an arterial) or difficult (e.g., a dozen DMS spread over 200 miles of highway)? Because of the great number to be deployed or the difficulty of accessing geographically dispersed devices, testing must be more intense and robust to reduce the potential for site visits after deployment. After the device is deployed, any hardware modifications become expensive and easily offset testing expenses.

For the device functionality, each operational requirement needs to be addressed with a specific test case. The test cases are likely to be complex; after all it takes a considerable amount of setup to be able to test some of the functions.

Consider, for example, testing for correct transitions between daylight savings time and standard time. This requires that the controller clock be set to a date (last Sunday in October[23] or first Sunday in April) and time prior to the 2:00 a.m. changeover time. The controller is then allowed to select and transition to the appropriate timing plan for that time and let the clock time advance to the 2:00 a.m. daylight savings time to standard time change point. At this point in the test, a check is made to determine whether the controller’s clock time was either set back to 1:00 a.m. for the fall changeover or advanced to 3:00 a.m. for the spring changeover. The check should also confirm that appropriate plan for the new time was selected and that the new cycle length, phases, and that the transition is made to the correct offsets. Note for the fall change, when the time is set back, it is important to allow the controller’s clock time to advance to (or be reset to near) 2:00 a.m. again and continue advancing

[23] Congress has recently changed the law adding 2 weeks to daylight savings time, so in the fall of 2007 the return to standard time will occur at 2:00 a.m. on the 2nd Sunday in November.

without repeating the set back to 1:00 a.m. and the selection of a new timing plan. If the agency routinely includes 2 AM plan changes, then this needs to be factored into the test procedures.

It is also critical that the test environment provide simulation tools (hardware and software) to fully verify the required operations and that those tools be verified for accuracy. For devices such as detector monitoring stations and actuated controllers, it is essential that detector simulators be available to accurately simulate volumes, occupancies, speeds, vehicle length, etc. If a ramp meter is supposed to change plans based on specific traffic conditions, then the test environment must provide a method to accurately simulate the value of the traffic condition (volume, occupancy, etc.) input parameters that are specified to cause a plan change, and it must be verified that the change was to the correct plan. For incident detection algorithms, the simulation environment must be able to present a profile of input data to replicate the anticipated conditions. To simulate detector activations at a given frequency (vehicles per hour), and occupancy based on vehicle length, or to provide trap contact closures to simulate various speeds and vehicle lengths. As an example, detector inputs to a controller/ramp (contact closures) can be simulated using a Universal Serial Bus (USB) relay device interfaced to a PC running simple detector simulation test software.

As noted above, the vendor usually provides the test plan, but the agency or its representative needs to work with the vendor to make sure that the test plan is representative of the operation and complexities required for their specific application and as documented in the procurement specification. The procurement specification should provide specific guidance with respect to what features, configurations, and operations must be demonstrated during operational testing and therefore included in the test procedure. The agency has a responsibility to review the test procedure and assure that the test cases proposed cover all of the operational test requirements and are structured to be representative of both planned and future operations as detailed in the specification.

The operational test procedure should subject the DUT to bad data, improperly formed packets, and other communications errors (e.g., interruptions, packet loss) to make sure that it handles the situation in a safe and orderly manner.

If NTCIP standards are invoked in the specification and include requirements for default configuration parameters for such anomalies as power and communications outages, the test procedure should include checking that these default configuration parameters have been successfully implemented following the respective outages. Such errors should not cause the DUT to reset to an unknown configuration, halt, or perform unsafe operations. As an example, communications errors should not cause an intersection to go to a flashing condition, and attempts to store parameters that are “out of range” should return an error to the management station rather than store the bad values. Where these conditions are not addressed in the invoked standards, it may be necessary to include some additional provisions in the procurement specifications. As an example, should a traffic controller check the validity of a timing plan when it is being stored in the database or when it is “run”? If the former, the management station is alerted to the bad data and can correct the situation before it affects operation. However, if the data is not verified until it is “run,” the traffic controller will not discover the error until it attempts to run the plan, causing disruption to the traffic flow. It is the responsibility of the agency to review the device standards (e.g., NEMA, CALTRANS TEES, IEEE) and the NTCIP standards (e.g., 1201, 1202, etc.) and determine if they meet their operational needs. If not, then the procurement specification must be augmented to address the differences.

The agency needs to work with the vendor to establish a test environment and test cases that will be representative of the proposed operation, including limit conditions (e.g., number of units on a line, maximum number of messages) and error conditions (power interruptions and communications disruptions).

6.2.5 Performance

Performance testing verifies that the device meets requirements that specifically relate to quantitative criteria (i.e. measurable) and apply under specific environmental conditions. *Performance* typically deals with timing accuracy, brightness, visibility, and the accuracy of the measurements (e.g., speed, volumes, temperature, RF levels).

The following is an example of performance testing that will test the resolve of both the agency and the vendor to accomplish, but is extremely important to assuring that the implementation performs correctly and will serve the intended purpose.

Verifying the accuracy of a traffic controller's internal clock and interval timing is one of the more difficult performance tests to perform. It is important that clock drift, clock accuracy (time-of-day) and the consistency of interval timing be verified to be compliant to the specification requirements. Controller clocks are typically synchronized (track) to the local AC power cycle zero (power line voltage) crossings and derive their basic once-a-second clock cycle from counting 60 successive zero crossings. The power company maintains the long-term accuracy of the 60-cycle frequency to within a few seconds, making it a very good clock synchronization reference. The power grid manages the addition and subtraction of cycles in a manner that ensures that there is no long-term clock drift; although the clock may wander within limits (up to 22 seconds has been observed), it will not drift beyond those limits. Testing for clock drift in the presence of short-term power interruptions requires very accurate measurements. For example, if the controller's internal clock maintenance software were to "loose" or "gain" even a single 60th of a second with each short-term power interruption (<500 milliseconds), over time the controller's clock will gradually drift from neighboring controllers that may have had a different short-term power interruption history. The resulting error or clock drift will be reflected as a timing plan offset error between adjacent signals which will compromise the green band.[24] This type of error can cause serious damage to arterial progression depending on intersection spacing and speeds.

Testing for controller timing accuracy is far more difficult than simply looking for clock drift over a 24-hour period. It requires an accurate recording device that allows the comparison between the output timing of the DUT and a time reference standard that is accurate to within a few milliseconds. Testing a unit for the accuracy of its internal clock (typically specified as ±0.005 percent) when power is not applied requires a reference to a national standard such as WWV or GPS. Because the AC power line can "wander" several seconds[25] during any test period, it is important to account for this effect to

[24] The green-band is determined by the time offset between the start of the green interval at successive intersections, the duration of the green intervals at successive intersections, and the desired traffic progression speed.

[25] The three continental United States power grids (Eastern, Western and Texas) are controlled such that there is no long-term "drift" for devices using the AC power line for time keeping purposes within a grid network. However, the AC power line does wander short-term by several seconds depending on the loading on the network, network disturbances, and switching. The instantaneous offset or deviation of a power grid from a

ensure the accuracy of the clock drift measurements. Conversely, when monitoring the timing of a device connected to the AC power line, it is important that the reference used for such measurements be calibrated with or linked to the AC power line being provided to the DUT.

Again, the agency needs to work with the vendor to establish an appropriate test environment, specific test measurement and recording equipment, and test cases with well understood and agreed on pass/fail criteria that will verify the quantitative measures specified in the performance requirements.

6.2.6 Standards Conformance

TMS hardware procurement specifications typically reference a number of different device and communication protocol standards and require conformance to them. Representative standards include the NEMA TS2 and CALTRANS TEES traffic controller standards, the advanced transportation controller (ATC) standard, the NEMA TS4 standard for dynamic message signs, and the NTCIP communications standards. The device standards generally describe functionality and some physical features of the device. The NTCIP communication standards define the meaning and format of the data exchanged between the device and a management station (e.g., closed loop master controller, TMC, etc.) and for the most part do not describe the device's functionality. However, the test plan for a TMS that includes these devices and standard references that must test the data exchanged, device functionality, and physical features actually delivered. Where a delivered function, feature or data exchange is required in the procurement specification to conform to a particular standard, the verification test must include steps to confirm that conformance.

For example consider the NTICP standards. It is important that the procurement specifications include a complete description of what specific parts of NTCIP standards apply to this procurement and for what devices. Specifications that simply require that the device “shall be NTCIP compliant” are meaningless without an entire series of clarifying statements. It is important to identify the communication media (e.g., analog telephone, Ethernet, EIA-232), the transport to be supported (e.g., point-to-point or point-to-multi-point), and whether the exchanges will be handled on an IP network. In addition, the procurement specifications must identify the application level protocols to be exchanged (reference NTCIP 1103) such as simple network management protocol (SNMP), simple fixed message protocol (SFMP), and simple transportation management protocol (STMP – also described as “dynamic objects”). Finally, the procurement specifications must identify the value ranges for certain variables (e.g., the number of messages to be supported), and which (if any) optional objects are to be supported. These details are representative of the complexity of developing a proper device procurement specification that invokes the NTCIP standards. In order to ensure the interchangeability of the field devices, the agency procurement specification must fully identify all of the NTCIP objects to be supported, the value ranges, any specific optional objects, and how special or proprietary functions are defined in terms of both functionality and communications protocols. The NTCIP standards are a powerful tool for the agency and can ensure interchangeability of field devices, but only if the agency takes the time to fully identify both the functionality and the objects to support that functionality. For those about to develop their first NTCIP-oriented device procurement,

WWV reference can be determined by contacting a regional WWV reliability coordinator. http://tf.nist.gov/timefreq/stations/wwv.html it is recommended that they review NTCIP 9001, which is freely available on the NTCIP web site at www.ntcip.org.

There are a number of communications testers and software applications[26] that can be used to exchange NTCIP objects with an ITS device, but there is no “generally accepted test procedure” for verifying NTCIP compliance to specific requirements. The Testing and Conformity Assessment Working Group under the NTCIP Joint Committee has produced a document, NTCIP 8007, “Testing and Conformity Assessment Documentation within NTCIP Standards Publications,” to assist the NTCIP working groups in developing test plans to be included in the various NTCIP standards. However, there is no assurance that this section will be added to the standards.

There are two different views of NTCIP testing that need to be understood. Since the NTCIP standards generally only define communications objects (parameters), one group feels that NTCIP compliance testing can be performed by verifying that data packets and parameters sent to the device are properly stored and available for retrieval. The second group wants to verify full device functionality based on the exchange of the NTCIP objects. Their claim is that the only way to be sure that the device will perform as expected is to combine both requirements into the NTCIP test plan. Hence any NTCIP test plan must verify both the data exchanges and the device functionality. The latter requirement is probably the most important for any ITS deployment and should be included in any device testing program.

NTCIP compliance testing typically consists of “walking the MIB”[27] to verify that the device supports all of the required data objects (and value ranges) of the NTCIP standards referenced in the procurement specification. Testing then uses individual SNMP *SET* and *GET* operations to verify that each of the parameters can be stored and retrieved, and that out of range data is rejected and the proper response occurs when it is out of range. If the device has a display, then that display should be used to verify that the parameter sent to the unit is what the unit displays on its front panel; if the unit allows the operator to store parameters, then the SNMP *GET* operations should be performed to verify that the data can be properly retrieved. Any errors noted while executing either of these processes means that the device does not conform to the NTCIP standard. There are a number of issues with NTCIP implementation that make this aspect of device testing very time consuming. First, while there are testers for SNMP, most of the available test devices do not handle STMP (dynamic objects), which are typically critical to low-speed communications to actuated signal controllers.[28] As a result, the test environment may need to use a sample central system or extend the testers with scripts to verify these objects. Secondly, many of the vendors have created custom objects and block objects to improve the efficiency of the data transfers. Where they are used, the vendor will typically have a means for verifying them. While the new versions of the standards (e.g., 1202V2) have standardized these blocks, not all vendors will support all versions of the standards. Further, the NTCIP standard only deals with the NEMA TS2 described functionality. When the vendor adds features and functions

[26] See NTCIP 9012.

[27] Management information base that lists all of the SNMP objects supported by the device. This is a series of *GET NEXT* commands until all objects have been retrieved.

[28] STMP is not generally supported by most ITS devices; STMP allows the efficient exchange of dynamically configured data which is essential to supporting once per second status monitoring. Devices such as DMS and ESS generally don’t need this type of high-speed communications and therefore may not support STMP.

beyond the basic NEMA standards, then the NTCIP testing must also be extended to verify the additional capabilities. With this type of “extension” (see section 7.2.1.3) comes a potential for conflicts between the standard NTCIP objects, the custom objects, and the block objects. Therefore, it is critical that the NTCIP testing verify the data exchanges using STMP, block objects, single objects, and the custom objects over all of the value ranges identified in the specifications. The vendor should create a test procedure with test cases to verify all of these issues and to demonstrate all functionality in the requirements.

In addition to simply verifying that the NTCIP objects can be exchanged with the device and that the device performs the proper “function” or reports the proper “status,” there is a need to verify the performance on the communications channel. If the agency plans to use low speed communications, then there may be timing requirements for the communications response that should be added to the specifications. Such considerations may be part of the overall system design and not part of the NTCIP standards. However, these must be verified as part of the NTCIP testing. Such timing can be critical to a system deployment and will affect the number of devices attached to a communications channel.

6.2.7 Maintenance and Serviceability

For maintenance activities to be carried out effectively and efficiently, it is important that the procurement specifications include some serviceability requirements. For example, a specification might require that a technician be able to repair or replace any field-repairable (replaceable) subassembly in 10 minutes without risk of personal injury or damage to the device with the use of common hand tools only. Such a requirement is somewhat ambiguous because, for example, the definition of *common hand tools* must be established, expected field conditions (e.g., weather, traffic, etc.) must be defined, and even the type of training the maintenance technician must have. Once these clarifications have been established, the agency simply inspects the device and goes through the process of replacing any device that looks difficult to service. The procurement specification should specify that the serviceability tests will be done by agency technicians attempting to perform the maintenance activity following only the maintenance instructions in the vendor’s written documentation. If specific training will be required to perform certain maintenance activities, these activities and training courses should be required deliverables defined in the procurement specification. Serviceability tests are an opportunity to verify required maintenance training and both the documentation (which is really part of the serviceability requirements) and the product’s compliance with the serviceability requirements. Inadequate training and/or poor documentation will most likely result in the failure of serviceability testing. This can be educational for both the vendor and the agency. It may point out the need to alter the mechanical design; for example, the addition of mounting rails to support heavy rack-mounted equipment.

The fragile nature of the components should be addressed in the maintenance procedures. Examples of these considerations might include anti-static straps, module carriers, and special packaging for the individual subassemblies or replaced components. Other examples of problems that might be discovered during the serviceability testing include tolerances on mechanical assemblies and the complexity of the disassembly and re-assembly process to change a component. It is possible that the mechanical design may need to be altered with such additions as mounting rails, locating studs, or alternate fasteners due to problems with threaded mounts.

The agency technician should go through the complete replacement operation for such commonly replaceable components as matrix panels for a DMS display, power supplies, power distribution assemblies, rack assemblies, fan assemblies, filters, load switches, flashers, and shelf mounted devices.

6.3 When Should Testing Occur?

The previous section dealt with the types of testing that may be part of the hardware test program for ITS devices. This section discusses the chronology of a typical ITS hardware testing for the procurement of ITS devices. The testing program described herein is based on the procurement of a **custom** or **new** device, not previously provided to the agency, and without an extensive deployment history. Depending on the maturity of the product, not all of the test phases described below will be required. Each of the following discussions will provide some guidance on the testing required based on the maturity of the device.

6.3.1 Acceptance of Previous Tests

Although the project specifications may require a full testing program to verify compliance with the procurement specifications, the vendor (contractor) may offer to submit the results of tests performed on the device for a previous project or as part of their product development process in lieu of performing all or part of the specified tests. This typically occurs when the device specified is a *standard* product or has been proven on previous projects. When developing the procurement specifications, this request should be expected and special provisions should be included that allow the agency to accept or reject the previous test results based on a well-defined criteria. This criterion should state that the testing must be on the *identical* product and must encompass all of the testing required in the project specifications. If a vendor can truly show this to be the case, then the agency should require that the vendor submit the previous test results, all data taken, details of the testing configuration, and the details of the test plan. The agency should also insist on inspecting the actual device that was submitted to the previous testing program to verify that the design is truly the same (trust but verify).

To determine whether the previous results are relevant, one needs to ensure that the current product design is identical to the unit that was previously tested. The determination of "identical," however, can be subjective. Often vendors will modify their design or are forced to change their design to manage component obsolescence, reduce costs, simplify construction, or to meet special project requirements; however, they will still claim that the previous test results are valid. When determining if a re-test is required, one needs to determine if and how the electrical or electronic design has changed, and whether the design changes could adversely affect the characteristics of the device. This includes the following evaluations:

- Thermal characteristics of the device in terms of internal case temperature, adjacent component mounting, and location of ventilation. Could the change affect the temperature of operation of the device?

- Mechanical characteristics to determine if the changes could affect the shock and vibration response of the unit. Examples include location of connectors, component mounting, hardware changes that might cause a change in how the unit handles the mechanical stress.

- Electrical and electronic characteristics to determine if any of the changes could affect the operation of the unit under transient and power interruptions. One also needs to determine if component substitutions compromise the margin requirements contained within the project specifications.

Changes that appear to be relatively harmless could have significant consequences, and often require analysis by an experienced electrical and/or mechanical engineer to determine whether there are issues that would warrant re-testing vs. acceptance of previous test results.

In addition, the agency should not automatically accept the unit because it is on the qualified products list of another agency or State. The procuring agency needs to request and carefully review the full test procedure that was followed, the data that was collected, and the results that were observed. Examples of issues that should raise doubt as to the acceptability of the test results include:

- Anomalies that may have been observed but ignored because they only occurred “once.”

- The test environment did not allow the device to be continuously monitored during all temperature transients and transitions.

- All of the tests were not preformed at all temperatures and line voltages. In other words, is there a question as to whether the testing was as thorough as your procurement specifications require.

There have also been instances where anomalies were observed during the testing, but a repeat of the specific test step did not repeat the anomaly so the device “passed” the test and was accepted. The agency should expect to review the cause of the failure (because this was a genuine failure) and determine if this is acceptable operation. Examples of areas which merit close scrutiny include power interruption testing, slowly varying line voltages, timing accuracy, and power line transients. Most ITS devices are expected to operate 24 hours per day 7 days a week without operator intervention; even very infrequent anomalies that require a machine reset or power to be cycled can create operational problems for both the agency and the public depending on the number of units deployed and the frequency of such disturbances.

> REAL WORLD EXAMPLE: In a recent review of the test results provided by a vendor to show compliance with the NEMA TS2 environmental requirements, the test report declared successful operation. However, upon close review of the detailed test report and test results, it was evident that one of the tests had failed during the first attempt and then did not fail during a repeat of the test step. In this case, the failure occurred in the monitoring electronics causing the device to report a non-existent failure. Such intermittent problems can be difficult to track and might cause an agency to “disable” the monitoring because of false errors. If a single sample product exhibits a few anomalies during a carefully controlled test, what can be expected when a large number are deployed? Such a situation should require that the vendor conduct a full review of the design of the monitoring circuitry to determine what happened and why and modify the design to avoid such false readings in the future.

6.3.2 Incoming Unit Testing

This is the lowest level of testing for hardware components delivered for installation. It involves a receiving inspection to verify compliance with the Contract Requirements Deliverables List and/or purchasing documentation, completeness of products and supporting documentation (operations and maintenance manuals, installation drawings and checklist were applicable), workmanship[29]; damage due to shipping; and stand-alone functionality testing if applicable. The functionality testing may include mechanical and interface testing as described below. Products found to be incomplete, of poor quality workmanship (as defined in the procurement specification) or damaged in shipment, or that did not pass applicable stand-alone functional testing should not be accepted. The agency should have the right to waive any minor receiving irregularities, such as missing documentation, mounting hardware or cabling and grant conditional delivery acceptance (if it is in the interest of the agency to do so), with final delivery acceptance subject to correction of the irregularities and completion of the unit testing within a negotiated time interval.

The agency should prepare a receiving inspection and unit test report and advise the vendor of the delivery acceptance. Hardware components that fail unit testing should be repackaged, with all other materials received, in their original (undamaged) shipping containers and returned to the vendor. If the shipping container has been torn open or shows extensive damage (crushed, water stains, etc.) it should not be accepted from the shipping agent. Note: completion of the unit testing and delivery acceptance will typically trigger the payment provisions of the procurement specification or purchase order. Once the unit has successfully passed unit testing and has achieved delivery acceptance status, it should be formally entered into the TMS equipment inventory and placed under configuration management control.

6.3.3 Interface Testing

Two types of interface testing are necessary: mechanical and electrical.

6.3.3.1. Mechanical

Mechanical interface testing involves inspection and test to ensure that the hardware component fits within specified space in an enclosure, equipment rack, etc. or on a required mounting bracket and has the required mounting points. It checks for component clearances, especially where the component moves on a mount such as a CCTV camera. It also includes checking to see that all required electrical and communications connectors are accessible, compatible with, and lineup with (or are properly keyed to) mating connectors before attempting mate testing. Mate testing ensures that connectors mate and de-mate with ease and do not have to be forced. This is not a functional interface test; it should not be performed with powered electrical or communications connections.

[29] Judging quality workmanship is very subjective particularly if no definition or qualifiers are included. Specific criteria for workmanship such as no burrs or sharp edges; freedom from defects, and foreign mater; and product uniformity, and general appearance should be included. They are applicable when the skill of the craftsman or manufacturing technique is an important aspect of the product and its suitability for the intended use. However, since there are no definite tests for these criteria, verification is by visual inspection.

6.3.3.2. Electrical

Electrical interface testing is performed subsequent to successfully passing mechanical interface testing. Electrical interface testing can be initially performed in a test environment without using an actual enclosure, a rack, or the mounting mechanical interfaces. However, electrical interface testing must ultimately be completed on components installed at their operational sites. It includes applying power and exercising the communications interfaces. Testing is performed to determine required compliance with at least some level of operational functionality; i.e., power on/off switches, displays, and keypads are functional and communications can be established with the device.

6.4 Hardware Test Phases

When procuring “new” or “custom” ITS devices that have not been deployed before, it is best to require a comprehensive hardware test program to verify the design and operation of the device from conception to final site operation. Further, by participating in the testing of the device from its design through deployment, the agency becomes familiar with the design, the operation of the device, the complexity of the testing, and the complexity of the device. Depending on the contract, the agency may also be able to work with the vendor during the early phases of the project (i.e., during the submittal phase and the prototype testing) to suggest or support changes to improve the overall cost, utility, and reliability of the product.

In general, the hardware test program can be broken into six phases as described below.

1. **Prototype testing** - generally required for “new” and custom product development but may also apply to modified product depending on the nature and complexity of the modifications. This tests the electrical, electronic, and operational conformance during the early stages of product design.

2. **Design approval testing** (DAT) –generally required for final pre-production product testing and occurs after the prototype testing. The DAT should fully demonstrate that the ITS device conforms to all of the requirements of the specifications.

3. **Factory acceptance testing** (FAT) –generally performs the final factory inspection and testing for an ITS device prior to shipment to the project location.

4. **Site testing** –includes pre-installation testing, initial site acceptance testing and site integration testing. This tests for damage that may have occurred during shipment, demonstrates that the device has been properly installed and that all mechanical and electrical interfaces comply with requirements and other installed equipment at the location, and verifies the device has been integrated with the overall central system.

5. **Burn-in and observation period testing** –generally performed for all devices. A burn-in is normally a 30 to 60 day period that a new devise is operated and monitored for proper operation. An observation period test normally begins after successful completion of the final (acceptance) test and is similar to the burn-in test except it applies to the entire system.

6. **Final acceptance testing** –verification that all of the purchased units are functioning according to the procurement specifications after an extended period of operation. The procurement specifications should describe the time frames and requirements for final

acceptance. In general, final acceptance requires that all devices be fully operational and that all deliverables (e.g., documentation, training) have been completed.

The following sections will explain the distinction between these phases and the product development cycle.

6.4.1 Prototype Testing

Prototype testing is intended to be a thorough test of the design of the device, but mechanically less stressful than the design approval testing (DAT*)* because it does not include the vibration and shock test procedures. The prototype testing should include full electrical and environmental testing to verify both the hardware and software design. It should also include inspection and verification of the serviceability, design requirements, and NTCIP compliance. If the prototype testing is robust and full featured, it is more likely that the DAT will be successful. Further, the prototype testing is the opportunity to verify both the ITS device and the DAT test environment.

Prototype testing is usually carried out on a single unit of each device type and takes place at either the vendor's facility or an independent testing laboratory if the vendor does not have the resources necessary for the testing at their facility. Even if they do have the resources, there are situations when an agency may require the use of an independent lab; for example, to accelerate the test schedule or when there is an overriding product safety issue.

Such a third party testing laboratory generally provides the measurement instrumentation, test chambers (temperature and humidity) and vibration and shock testing equipment that is expensive to acquire, operate and maintain. An independent testing laboratory will also have the technical expertise and experience necessary to conduct and monitor the testing, track access to the DUT, analyze the test results, produce a detailed test report, and certify the test operations that they are given the responsibility for performing. However, the laboratory is not likely to include domain expertise; hence, they will simply run the tests according to the documented test procedures and make the necessary measurements and observations. Thus, the robustness of the test procedure will determine the utility of the laboratory testing. More often, the laboratory simply provides the environmental and measurement equipment while the actual device testing is performed by the vendor.

During prototype testing (for a new product) it is assumed that the vendor may have included some "cuts and paste" modifications[30] to their circuit design to correct defects or problems discovered

[30] "Cut and paste" refer to a practice used by vendors to modify an existing circuit board's electronic design and layout by bypassing existing copper lands used to connect the board's components or external connectors. Under proper conditions each component is mounted directly to the circuit board, and all circuit lands are properly routed to the components and connectors. However, when a design problem is discovered, the vendor may correct the problem by simply gluing a new component to the circuit board, cutting existing leads, and running small wires (jumpers) to connect the new component to the proper points in the circuit. When such "modifications" are complete, the circuit reflects the final design, but the construction practices are generally considered unacceptable (particularly for a production product). If the circuit modifications are for relatively low-speed sections of the design, such cuts and pastes are not likely to affect operation; however, for high-speed designs (e.g., 100 Mbps Ethernet) such modifications could compromise the operation of the circuit. Such repairs are typically allowed at the discretion of the accepting agency for a prototype, or for very low number

during the early design testing. Thus, although the prototype phase must include the proposed final mechanical packaging, it is generally not required that the prototype meet all of the design and construction requirements identified in the procurement specifications such as hardware, paint color, component mounting, and printed circuit construction practices. Some of the “modifications” can have a tendency to compromise the structural integrity of the unit, hence most prototype testing does not mandate the full vibration and shock.

Under some circumstances, the agency may require that the vendor deliver a “mockup” of the proposed mechanical design to allow the agency to evaluate conformance to the serviceability requirements. An example that might warrant an early mockup is a custom traffic controller cabinet design to evaluate clearances and serviceability for equipment that will be housed in the proposed cabinet design. While such a mockup is not part of the prototype testing, it should be included in the procurement specifications if the agency feels it necessary to examine a representative product at an early stage of the design approval process.

While it has been stressed that the prototype testing should include full functional testing of the device, it is not unusual for the vendor to request that the hardware be tested with limited software functionality due to the prolonged software development cycle, changing operational requirements, and the pressures of the project schedule. Under these circumstances, the agency should proceed very cautiously as long as the hardware aspects of the product can truly be separated from the total product. There is a risk that latent problems will necessitate a change to the hardware design – which would require a complete repeat of the testing. Under these circumstances the vendor and the agency must weigh the risk of further delays to the project against the benefits of allowing the vendor to complete the hardware testing and move on toward the DAT. The risk is that in some circumstances, such as the traffic controller timing issues discussed earlier, the hardware design and the software design may be more closely coupled than is apparent. As a result, successful completion of the prototype testing (with limited software functionality) is no assurance of proper operation of the final product or completion of the DAT.   It is important that such risks be understood by all involved and that the agency does not assume liability for any of the risks if design changes are necessary. Such risks could include increased cost of the testing program and delays in the completion of the project.

As an example, during the initial field deployment of a new traffic controller design, it was observed that the clocks were drifting by minutes per day; after extensive investigation it took a combination of a hardware change (in this case re-programming a logic array) and a software change to correct the situation. While the software changes were easy to effect (reload the firmware using a USB device), the changes to the logic array were more time consuming and required that each field device be disassembled to access the programming pins. These issues were not discovered during the DAT because the full controller functionality had not been complete at that time – and the decision was made to allow the vendor to progress even though not all of the required functionality had been completed.

(pilot) production runs and with the understanding that the next production run will not include such modifications and previously accepted products will be replaced at the vendors cost with new products from that production run.

Another related situation arises when the vendor requests to *skip* the prototype testing and go directly to the DAT. This can work to both the agency's and the vendor's advantage if the vendor is confident in their design and has a well developed and acceptable test environment. However, if prototype testing was required by procurement specification, a contract modification will be required to eliminate it, and the agency should carefully assess benefits it should receive in return, i.e., reduced schedule and testing costs against the risk of skipping prototype testing. When prototype testing is skipped, there is a heightened risk that a design defect will be found in the subsequent DAT, necessitating a change to the circuit design that violates one or more aspects of the specifications for construction and materials. Further, the DAT should be performed on several devices and the prototype testing is typically only performed on a single device. In general, the testing program is structured to complete the prototype first because it offers a lower risk to the vendor by allowing construction practices that are not acceptable at the DAT phase. In addition, it provides an opportunity for the agency to evaluate the testing plan and environment.

It is also possible that the vendor may fail (what the vendor considers to be) some minor aspect of the prototype testing and requests permission to go directly to the DAT phase of the project. As long as the agency understands the risk and that the risk lies with the vendor and not the agency, such requests should be considered if there is a benefit to the agency (i.e., schedule, cost) to do so and the request is contractually acceptable. Procurement specifications should limit the test period and the number of unsuccessful vendor test attempts allowed without incurring significant penalties, such as withholding progress payments, liquated damages, and canceling the contract for non-performance. The limited test period and limited number of test attempts helps to contain the agency's costs of participate in testing. Note: if the vendor is allowed to skip prototype testing or the prototype is accepted without satisfying all aspects of the required testing, any contractual remedies the agency may have included in the procurement specification to cover prototype test problems are no longer available.

6.4.2 Design Approval Testing

The design approval testing is the next stage of the device testing and is intended to verify the complete product (in final production form), including packaging is in full compliance with the specifications. Typically, if the vendor "got it right" during the prototype testing and there were no changes to the mechanical design or construction, then the DAT should be a routine exercise. The only additional testing performed at the DAT is for vibration and shock following either the NEMA or the CALTRANS procedures – depending on which was designated in the procurement specification.

For the DAT, the testing should be performed on multiple units that are randomly selected by the agency from the initial (pilot) production run of the devices. All of the testing and inspection is the same as that for the prototype, except for the addition of the vibration and shock tests that should be conducted <u>before</u> the start of the environmental and functional testing. Depending on the number of units being purchased, the agency may wish to insist that a minimum of two (2) and up to five (5) units be subjected to the full testing suite. For a typical procurement of 200 or more, it is recommended that a least 5 units should be tested.

The procurement specifications should reserve the right to require a repeat of the DAT if there is a change to the design or components. If changes are required, then the agency (or its consultant) needs to analyze the nature of the changes and determine if the full battery of testing is necessary or whether a subset of the original testing can be performed to verify the effects of the change. In some cases, no further retesting may be necessary.

Another issue to be addressed during the DAT is the overall integration testing and the availability of the other system components. If the field device is to be connected to a large-scale central system, then it is best to bring a portion of the central system to the DAT (or extend a communications link to the test facility) to verify the communications aspects of its operation, such as object encoding, protocol support, and performance timing. Where this is not feasible, the vendor should be required to develop and demonstrate a central system simulator that provides the data streams specified and measures the performance and response from the device.

For certain devices such as dynamic message signs, the prototype and DAT may be waived or required only for a sample system (e.g., the controller and one or two panels) because of the expense and availability of test chambers large enough to test a complete sign. This will depend on the maturity of the device, the number being purchased, and the thoroughness of the vendor's previous testing program. However, when the formal environmental testing is waived for devices such as DMS, it is recommended that the power interruption testing, transient testing, and voltage variation testing be performed for the complete DMS as part of the factory acceptance test.

Note that we have continued to stress the need to verify operation under transient power conditions. Since all of the ITS devices are expected to operate 24x7x365 without operator intervention, the goal is to ensure that typical power line anomalies do not necessitate the visit of a field technician to "reset" the device, clear a conflict monitor, or reset a circuit breaker. Such situations can compromise the integrity of the overall TMS at times when their use may be mission critical (e.g. evacuation during storms, evening rush hour during thunderstorms).

6.4.3 Factory Acceptance Testing

The factory acceptance test (FAT) is typically the final phase of vendor testing that is performed prior to shipment to the installation site (or the agency's or a contractor's warehouse). For a complete DMS, the FAT serves as the agency's primary opportunity to view and review the operation of the device for any special features, and to inspect the device for conformance to the specifications in terms of functionality, serviceability, performance, and construction (including materials). The DMS FAT should include all the elements of a device DAT except the environmental (temperature and humidity), vibration, and shock testing.

As with the prototype testing and the DAT, the vendor should provide the test procedure for the FAT. The FAT should demonstrate to the agency that the operation of the device, the quality of construction, and the continuity of all features and functions are in accordance with the specifications. If the device (or its constitute components) passed a DAT, then the FAT procedure should verify the as-built product is "identical" to the device (or components) inspected and tested during the DAT.

When the DAT was not performed for the complete device, such as a DMS, the FAT must include inspection and verification of the complete assembled device (with all its components) including those specification requirements (physical, environmental, functional and operational) that could not be verified at the DAT. A DMS will have to be disassembled after testing for shipment to the installation site. It is important to assure at the FAT, that the complete list of deliverables, including all the specified DMS components, cabling, fasteners, mounting brackets, installation checklist, drawings, manuals, etc. is verified for completeness and accuracy.

For more modest devices such as ramp controllers, traffic controllers, and ramp metering stations, the FAT inspection should include doors, gasket, backplane wiring, cable assembly, hardware (nuts and bolts), materials, and the list of deliverables such as load switches, prints, flashers, etc.

Each unit must be subjected to the FAT before being authorized for delivery. Once the FAT has been completed, the unit is deemed ready for transport to the installation site. For some devices, such as Dynamic Message signs and the first units of custom and new products, the agency should plan on attending the test and being part of the final testing and inspection procedure. For other standard devices, such as CCTV cameras, traffic controllers, data collection stations, the vendor will conduct the FAT in accordance with the procurement specification without the presence of agency personnel. The vendor should be required to provide a detailed description of the FAT procedure used; keep accurate records of test result including date, time, device serial number, and all test equipment, test data, etc. for each unit shipped; and identify the person(s) responsible for actually performing and verifying the test. While agency attendance at an FAT is not usually included for production devices, the procurement specification should require the vendor to notify the agency 10 days prior to when the tests will occur and reserve the right to attend any and all tests. Remember, the goal of the FAT is to ensure that the device has been built correctly and that all functions and interface circuits function properly and that all of the Quality Assurance requirements of the specifications have been fulfilled.

An extension to the FAT for electronic devices that is typically included in the procurement specifications (and referenced standards such as the CALTRANS TEES) requires that all devices be temperature cycled and subjected to a factory “burn-in” for a period of time (typically 100 hours). This procedure has been adopted to reduce the number of units that fail upon installation or during the first few days of operation (typically known as product infant mortality). In general, a 100-hour burn-in period is commonly specified for most electronic products; however, many vendors have been successful with less time, although there is no study showing the benefit to any specific number of hours.

6.4.4 Site Testing

Once the unit has been shipped from the factory, the remaining testing is closely related to the integration and final testing of the system. While the preceding factory based testing (prototype, DAT, FAT) will vary based on the maturity and experience with the product, the site testing should be the same for all of the levels of product maturity.

At this point, it is important to be aware of the potential issues that may involve payment terms. In many contracts, the owner pays 95 to 100 percent for the device the moment it is delivered to the job site. This payment is common with most civil works projects and is commonly called payment for “materials on hand.” If the agency pays 100 percent, it looses all leverage[31] if the device fails the site tests. Also note that for large items such as DMS it is an expensive proposition to take it down, repackage for shipment, and send it back to the factory. Therefore, the agency should keep sufficient funds to hold the vendor’s attention and should do as much as possible to ensure that the device has the best chance of passing the on-site tests. Provisions for withhold amounts and the conditions for releasing these funds to the vendor must be clearly defined in the procurement specification or purchasing documentation.

Another word of caution with regard to the relationship of product delivery, site installation, and system integration. The agency must coordinate these activities so that ITS devices are not

[31] Except recourse to the bonding company where a bond is in place.

warehoused or installed in a non-operational state for prolonged periods (>60 days). There have been instances where DMS have been delivered to a storage yard so that the vendor could receive payment, yet site installation and system integration were more than 6 months delayed. The DMS remained in an un-powered and neglected condition such that when they were about to be site installed, they needed to be overhauled because the sign systems were not powered and hence the moisture and dirt were not managed. In at least one case, by the time the devices were installed, the vendor was out of business leaving the project in a tough situation. Knowing that this can occur and coordinating project activity to ensure that high technology devices are delivered and tested only when they can actually be used and placed into service will protect everyone and contribute to the success of the project.

For the purposes of this discussion, site testing will be broken into three sub-phases:

1. Pre-installation testing.

2. Initial site acceptance testing.

3. Site integration testing.

The order in which and how these are managed depends on the method of procurement, the agency's facilities, and the overall program. Where a single contractor is responsible for all aspects of the system, including design, construction, and testing, these are in the most logical sequence. However, if there are multiple contractors and if the device is only one aspect of an overall ITS deployment, it may be necessary to accept the devices prior to the availability of the central system. Further, it may be necessary to accept and store the ITS device at a contractor's facility prior to site testing. Alternatively, the agency may establish an integration testing facility where all of the devices from various procurement contracts are installed, configured, tested, and burned in prior to transfer to the field.

6.4.4.1. Pre-installation testing

This phase is required to detect any damage during shipping. It is also an opportunity to fully configure an equipment package (e.g., controller) and burn it in (if not already done at the factory). The pre-installation testing is performed by the agency or its installation or integration contractor, who must receive and store the devices and integrate them with other ITS devices before installation. This type of testing may provide an opportunity to perform unit and integration testing in a controlled test environment prior to field installation. In either case, the intent is to verify that all of the equipment has been received without damage and is in accordance with the approved design (i.e., identical to the DAT-approved units). To fully verify an ITS device, a pre-installation testing and integration facility should be established by the agency or its installation and integration contractor. The pre-installation test and integration facility should include a simulation and test environment sufficient to exercise all inputs and outputs for the device and to attach to a communications connection for on-line control and monitoring.

Pre-installation testing can also provide an opportunity for the agency to fully configure the device (or full subsystem) for the anticipated field location. This approach allows the agency to verify all settings, input/output assignments, and operational features for the intended location. Some agencies and projects have established such a test facility that included environmental testing, so that the incoming units could be fully temperature cycled and monitored prior to installation.

If the purpose of the testing is to prepare to store the device, then the vendor should be consulted for recommendations on the proper procedures and environment to store the devices once their operational integrity has been verified.

6.4.4.2. Site Acceptance Test

The site acceptance testing is intended to demonstrate that the device has been properly installed and that all mechanical and electrical interfaces comply with requirements and other installed equipment at the location. This typically follows an installation checklist that includes physically mounting the device and mating with electrical and communications connections. Where necessary and appropriate, site acceptance testing can be combined with initial setup and calibration of the device for the specific installation location. Once the installation and site acceptance testing has been successfully completed, the system equipment inventory and configuration management records should be updated to show that this device (type, manufacturer, model, and serial number) was installed at this location on this date. Any switch settings, channel assignments, device addressing, cabling, calibration, etc. unique to this location should also be noted for future reference.

Prior to connecting or applying power to a newly installed device, the characteristics and configuration of the power feed (i.e., supply voltage and grounding) and circuit breakers (i.e., ground fault interrupters and proper amperage ratings) should be **re-checked** (these should have already been tested and accepted when they were installed). Remember, you own the device; improper or incorrect installation will void the warranty and could be hazardous. During this testing, it is necessary to verify all of the inputs and outputs for the device and the calibration of such parameters as loop placement, spacing, geographic location of the device, address codes, conflict monitoring tables, etc. It should also include verifying that connections have been properly made; e.g., the ramp metering passage detector is correctly identified and terminated and hasn't been incorrectly terminated as the demand detector. This will probably require driving a car onto the loop and verifying the controller (and system) is getting the proper input. The exact requirements will depend on the type of ITS device. All devices should be tested for their power interruption response to ensure that they recover according to specification requirements in the event of power interruptions. The importance of checking the power interruption response will become abundantly clear following the next lightening storm, particularly if it wasn’t done and all initial settings and calibrations must be re-applied.

The agency should develop the test procedure and installation checklists for this testing if this installation is an extension of an existing system. If an installation or integration contractor is involved, the contractor should be required to develop this test procedure, which the agency must then review to ensure that it will verify the intended usage and compliance with the contract specifications.

6.4.4.3. Site Integration Testing

Depending on the schedule and availability of the system components (i.e., central system and communications network), once the device has been demonstrated to function properly (successful completion of the site acceptance test), it will be integrated and tested with the overall central system.

The test procedure for this aspect of device testing must include the full functional verification and performance testing with the system. This should also include failure testing to show that the device and the system recover in the event of such incidents as communications outages, communications errors, and power outages of any sort. This testing must include support for all diagnostics supported by the central system. Site integration testing should also look closely at the communications loading and establish a test configuration that will most closely simulate the fully loaded network.

The agency should construct a detailed test plan for the integration testing to show that all of the required central system functions [available for this field device] are fully operational.

6.4.5 Burn-in and Observation Period Testing

Once the device has been made operational, it should be exercised and monitored for a reasonable period of time. The project specifications establish this burn-in test period. It generally varies from 30 days to 60 days depending on the policies of the agency and typically requires that the device exhibit fault-free operation. During this period, the device should be exercised and inspected regularly following a test procedure (operational checklist) developed by the agency or the system integrator (and reviewed and approved by the agency).

An observation period is normally applied to a major subsystem or the total system. Normally it begins after successful completion of the final (acceptance) test and is similar to the burn-in test except it applies to the entire system. It should be noted that there are many variations of burn-in period, final (acceptance) testing, and observation period and the agency should think the process through thoroughly and understand the cost/risk implications.

It is important that the procurement specification defines what is meant by "fault free" operation and that the vendor clearly understands not only what is meant but also that the clock may be re-set and the testing repeated or extended if the device fails within the test period. The vendor has a huge disincentive to discover faults at this point. The procurement specification must unambiguously spell out what is acceptable and what the expected vendor response and corrective action is. It should also clearly state how the vendor has to document and respond to faults the vendor observes or that are reported to the vendor by the agency or its contractor. Typically, "faults" or failures are divided into minor and major faults, and it is important that the procurement specifications identify each and how the period of observation is extended for each. The specifications must also define how the period of observation is affected by outages unrelated to the device, such as communications outages, power outages, and central monitoring system outages. Minor failures are typically those that do not affect the utility of the device, but still represent incorrect operation. Examples might include the failure of a few pixels in a dynamic message sign or the failure of one string in an LED pixel with multiple strings. Basically, if the failure does not compromise the useful operation of the device, it might be classified as minor in nature. The agency will need to review and establish the criteria for each device type. All other failures would be considered major. **The provisions that are included in the procurement specification for the observation period should be carefully reviewed as they can have a big impact on expectations, cost, and the future relationship between the agency and the vendor.**

The period of observation should also be used to track such long-term requirements as time-keeping functions.

There are a number of different approaches that have been taken when adjusting the period of observation for major and minor failures. Some require that the device maintain a specific level of availability for a given number of days, while others have established a point system for each failure type and restart the period of observation once a certain point level is reached. Still others suspend the period of observation when the minor problem is detected and then continue once the failure has been corrected. Then, for major failures, the period of observation is restarted from the beginning once the failure has been corrected.

The agency must be realistic in their expectations. It is not reasonable to expect that a system with 25 DMS, 50 CCTV cameras, and 10 ramp meters will operate continuously without a failure for 60 days. Hence, a requirement that restarts the observation period each time there is a major failure will virtually ensure that the observation period is never completed. Hence, the point system is preferred and credit is given to the level of availability for the system as a whole.

6.4.6 Final Acceptance Testing

Final acceptance is a major milestone for any project. It represents full acceptance of the product by the agency and triggers final payment terms and conditions of the procurement specification. Final acceptance often marks the start of the warranty period for the ITS devices depending on the procurement specifications or purchase agreement.

The final acceptance test for devices usually takes place once all of the devices have been installed and have completed their period of observation and all other project work has been completed.

The final acceptance test should demonstrate that the device is fully operational and that it continues to function as specified in its installed environment and operational conditions, including compatible operation with other subsystems and devices. The procurement specification must clearly establish what is expected at the final acceptance test and the pass/fail criteria. The agency or its integration contractor should develop this test procedure. The same checklist used during the site installation test to verify proper operation can also be used for the final acceptance test. However, additional testing may be necessary to verify correct operation with, for example, similar devices on the same communications channel, up and downstream traffic controllers, and with other subsystems and devices that were not available or operational in previous tests. Typical procurement specifications will mandate that on the date of final acceptance, all devices must be fully operational and functioning properly. This is an ideal goal, but not realistically achievable for a system of any significant size. The procurement specification should make allowances for final acceptance of all installed and fully operational products on a certain date, with final acceptance of the remaining delivered but uninstalled products occurring in stages or as those products are installed and tested.

In summary, the expectations for final acceptance must be realistic; equipment of reasonable quality and reliability should be able to pass and it is not reasonable to expect 100 percent operation of all devices and systems over a prolonged period of time. Therefore, allowances must be made to ensure that the agency is protected, and that the vendor/contractor can have the equipment accepted.

6.5 Other Considerations for the Hardware Test Program

The preceding sections described a complete hardware test program for ITS devices based on the assumption that the device represents a custom design or new product. For most TMS projects, standard ITS devices will be specified, so the required hardware test program is usually less involved and therefore less costly. In either case, the test program needs to be developed in concert with developing the procurement specifications. The agency should consider the following recommendations when developing the test program.

If the agency determines the risk to be acceptable, develop the procurement specifications to allow acceptance of past testing for standard product oriented ITS devices (i.e., those that are listed on a

QPL and have proven design and deployment history). However, since it is not known with certainty that a standard product will be furnished, the specifications need to describe the minimal level of testing desired. Therefore, there must be criteria for the prior tests and specification of what must be done if they fail the criteria.

Evidence of past testing will generally take the form of either a NEMA test report or a CALTRANS (or other State's) qualified products list status. For *standard* devices that will be deployed in a *standard* environment (i.e., not at environmental extremes), this is likely to be adequate. However, to make that determination, it is recommended that the agency request a copy of the complete test report showing the test environment, exactly what units were tested, and how the results were measured and logged. The test report should include all observations and measured results and should come from an independent testing lab, another State's testing lab, or a university testing lab. The agency should review the results and confirm that the testing demonstrated that the device was subjected to the testing required by the project specifications and that the results were acceptable. It is also important that the electrical, electronic, and mechanical design of the unit that was tested be the same as the device being proposed for your project. Any design differences require a careful review to determine if those differences could have a material effect in the overall performance of the unit you are procuring.

In the case of a device that is on another State's QPL, **caution needs to be taken**. The agency should insist on receiving a complete record of the test program (procedures) along with the details on the test environment and the test results. These should be evaluated for conformance to the agency's test requirements and contract specifications. Not all State laboratories fully test the product received, and, in some cases, the test procedure followed may be based on their expected usage rather than verifying the extreme conditions of the procurement specifications. Experience has shown that some QPLs may include older (or obsolete) versions of products and that the actual test process was more ad hoc based on the experience of the tester than a rigorous full feature test procedure.

If a review of the prior test procedures indicates that some of the requirements of the agency's specifications were not verified, the agency should require that the vendor conduct a subset of the testing to verify those additional requirements. Examples of extreme conditions that may not have been included in prior testing: slowly varying the AC line voltage or operation at an ambient temperature of –10° F (to verify a sub-second LCD response time requirement that may require a heater to satisfy).

> REAL WORLD EXAMPLE: During a DAT for a large-scale traffic control project, the vendor was required to perform simultaneous environmental testing on five units. During the testing, one of the units failed, requiring the DAT to be rescheduled for a later date. The failure was traced to a **design defect** in a device that was on the CALTRANS QPL and approved for use. Design modifications were required to correct the defect that had not been detected in previous testing. It was found that CALTRANS testing had been performed on only one unit and their test environment did not subject the unit to AC power line impulse noise while in full operation.

For ***modified*** devices, the agency needs to review the nature of the modifications and determine how much of the testing program should be required. The most conservative approach is to mandate the full testing suite of prototype testing followed by the DAT. However, such testing is expensive for both the agency and the vendor, and may be unnecessary if the modifications do not significantly alter the basic design of the device. Examples include DMS where the only change is the number of rows and columns. In this instance, there should be no need to submit the sign to a complete re-test just because of these changes. However, changes to features such as sign color and the ventilation system could affect the thermal characteristics of the device, which is critical for LED technology. Therefore, although a complete re-test may not be necessary, a complete review of the thermal design should be conducted. It is also likely, for a DMS, that the testing was performed on a "sample" device consisting of several panels with the controller and some of the electronics. While the environmental testing would be acceptable, this does not subject the whole sign to the power interruption, transients, and varying line voltage. It is recommended that this subset of the testing program be repeated on the whole product, most likely as part of the FAT. Do not be dissuaded because the device requires a significant power source to conduct these tests. This part of the testing is intended to prove that the complete sign will not be affected by specific conditions—and there can be subtle problems when the entire sign is subjected to such testing.

For other devices such as traffic controllers, detector monitoring stations, and ramp controllers that use proven hardware, a difference in the number of load switches in the cabinet would be considered minor. As long as the full collection of devices is the same as previous test programs and as long as the previous testing was in a "fully loaded" cabinet, the testing can be accepted in lieu of a complete repeat of the prototype and DAT. However, if the vendor designs a new controller unit that uses a different processor, or changes the packaging of the controller electronics, then a complete re-test is probably warranted.

Testing is about controlling risk for both the vendor and the agency. Under ideal conditions, the ITS devices procured for a TMS will be standard products provided by vendors with a proven track record. However, in a low bid market place, the agency is required to purchase the lowest cost compliant units. A procurement specification that includes a rigorous testing program serves notice to vendors that the agency will test those units for compliance with specification requirements. Noting this, the vendor is more likely to review their design and ensure that it fully complies with the procurement specification.

6.5.1 Who Develops the Test Procedures

The issue of which party develops the test procedure has been discussed in several sections above. It is generally recommended that the vendor develop the test procedures for all phases of factory testing; i.e., from the prototype testing to the factory acceptance testing. This accomplishes two things that can improve the overall design of the device and the testing program. First, the vendor is forced to conduct a thorough and complete review of the specifications and standards when developing a test procedure to verify conformance to all of the requirements. From past experience, this has had the effect of improving the overall device reliability. Second the vendor can develop the test procedure based on their available resources. This means they can setup the test environment and base the test schedule on the availability of test chambers, test equipment and personnel. To make this happen, the agency must include requirements in the procurement specification for vendor development of test plans and procedures and for conducting the testing in their test environment. The procurement documents must also give the agency the right to require additions and modifications to the vendor prepared test plans and procedures, test environment, and approval rights. The procurement specifications must also stress that the test procedures are to be detailed, thorough and cover all aspects of the requirements. Make it clear in the procurement documents that sketchy and rough outlines for a testing program will not be acceptable.

However, once the vendor has developed the test procedure, the agency personnel must review the procedure to ensure that all aspects of the requirements are verified. It is best to require that the vendor include a requirements traceability matrix[32] in their test plan to show that there is a test case for every requirement. Consider the following perspective when reviewing the test plan:

1. The vendor can be required (if stated clearly in the procurement specification) to perform any test and inspection necessary to demonstrate product compliance.

2. If the vendor passes the test, the agency has agreed to accept and pay for the product.

When it comes to the field or site testing, however, it is not clear who is the best party to develop the test procedures. This will largely depend on the contracting process and the expertise of the agency. However, the procurement specifications must clearly state the requirement for the test procedures themselves and that the acceptance test must demonstrate that the product meets specifications. Where an integration facility is developed, it is likely that the vendor will develop the test procedure in concert with the central system provider. The agency should establish the requirements for this testing (e.g., full configuration of signal displays, maximum number of loop detectors), and then let the contractor develop the procedures and facility. In other cases, the agency may be providing the facility and must tailor the procedures around what is available to them. In this latter case, the agency should develop the test procedure.

What is important is that the test procedure be thorough and include test cases and test steps that fully verify that the device complies with the requirements of the specifications. The level of detail will vary through the testing program where the DAT is the most rigorous and the site testing is typically only verifying that all features and functions are operational. Other sections of the guide will deal with the development of a good test procedure.

6.5.2 Cost of the Testing Program

Testing can be expensive because it generally involves travel and consultant support for the agency and laboratory facilities and personnel for the vendor. How these costs are allocated and accounted for will depend on the contract specifications and the procurement rules of the agency. The following provides some suggested considerations when developing procurement specifications.

If it is likely that the procurement specifications will result in the development of a custom device and the agency plans a full testing program, there are two aspects to the cost that can be treated separately: 1) travel expenses and 2) the agency's labor expenses.

If the agency plans to visit the vendor's facility to observe and participate in the testing, it should be noted that the travel expenses could vary greatly depending on the location of the vendor. It is likely that a week of travel (on average) will cost $1600 or more per person for travel expenses depending on the location, the amount of advance notice, and the airlines servicing the location. Some procurement specifications require that the vendor cover the travel expenses (hotel, air, and local transportation) for a specified number of agency representatives. However, this increases the cost to

[32] The requirements traceability matrix lists each of the requirements to be verified by the test in a matrix format. It includes for each requirement: the requirement's source reference paragraph no., the requirement statement from source specification; and provides the test method, test case no., and test responsibility for verifying the requirement.

the vendor based on its location relative to the agency. When the revenue for the procurement of the devices is relatively small (e.g., $100K) this could have a significant impact to the vendor's bid price and is likely to place some of the vendors at a cost disadvantage. To mitigate this situation, the agency may wish to fund a limited number of factory visits internally and then only require that the vendor pay the expenses if a re-test is required due to test failure. This latter approach allows the vendor to share the risk and will not discourage more distant vendors from bidding quality devices for a project. The number of "free" tests may vary depending on the scale of the project and the complexity of the device. Simple devices should only require a single visit, while more complex, custom developments might require more than one attempt to pass the DAT since all five units must function properly.

If the costs (including travel, per diem and labor costs for the agency's personnel and the agency's consultants) are borne by the agency for a limited number of tests, this becomes an incentive to the vendor to pre-test and ensure that their test environment and devices are ready. This is especially true if the vendor knows that they will bear these agency costs for additional test attempts. Note that unless specific provisions for the agency to recover these costs are included in the procurement specification, the agency may find they have no recourse if the vendor repeatedly fails the testing, or if it is apparent that the vendor is not properly prepared or that the testing environment is not as proposed and reviewed. Of course, the final recourse is to cancel the contract (assuming such provisions are included in the specifications) but such drastic measures are seldom invoked.

In summary, it is recommended that the agency self-fund one or two rounds of the required factory or laboratory testing. However, each "visit" to the factory location counts as a test attempt. After two (or three) attempts, the vendor should be expected to bear all costs associated with additional testing. Each stage of testing is considered a separate "start" hence the vendor may be allowed one attempt to pass the prototype test, two attempts to pass the DAT, and two attempts to pass the FAT. Any additional attempts for each of these test stages would require that the contractor pay all expenses; when the vendor is expected to bear the costs, the specifications should indicate the labor rates and rules for per diem that will be employed. Note that if the agency includes the cost of consultant services to assist in the testing, these labor costs are very real so they should be dealt with in the contract specifications.

This approach shares the risk between the agency and the vendor. It is important that the test procedures provided are thorough and complete, showing the entire test environment, listing all test equipment, and detailing how the testing will be performed. The agency must review the proposed test plan and ensure that it is well documented, well designed, clearly stated and understood by all, and well planned in terms of a daily schedule of activities. **The testing date should not be scheduled until the test plan has been approved.**

6.5.3 Test Schedule

The procurement specifications should outline how the testing program fits into the overall project schedule. The project schedule must allow sufficient time for the agency to review the test plan (allow at least 30 calendar days for this review) and for the vendor to make corrections. The testing can be complex and lengthy. A typical test plan can easily run into several hundred pages with the inclusion of the test environment schematics, descriptions of the test software and simulation environment, and copies of the test cases, inspections, and requirements traceability check lists. The agency is likely to need maintenance, electrical engineering, and mechanical expertise to review this submittal. It is recommended that example test plans and test procedures be provided at the level of detail required by the procurement specification as part of the pre-bid vendor qualification materials or in response to the RFP.   Additionally, they should be discussed again during the project “kick-off” meeting, particularly if the vendor’s example test plans and procedure fall short of what was required by the procurement specification. This will serve to re-enforce the level of detail required by the procurement specification.

Because the testing will involve travel and is subject to travel industry pricing policies, testing schedules should not be set until the test plan has been approved. The project specifications should allow mutually acceptable test schedules to be set within pre-defined limits. However, once set, any adjustments by either party could be a cause to adjust the project schedule (and may result in claims by the vendor due to delays).

If the test fails or must be terminated due to environmental or equipment problems, the project specifications should require that the vendor provide a complete report as to the cause of the failure and corrective action(s) taken. The agency should establish some minimum waiting period between tests – typically the same advance notification required to schedule the initial test.

The agency should avoid the circumstance where there is a desperate need for the product, the vendor is behind schedule, and the testing and inspection finds a defect that should have been addressed during the design phase of the project. This situation can force compromises that one would not even consider earlier in the project. The best way to avoid these issues is to develop the inspection check list and test procedure very early in the project so that reviews and discussions of possible deviations can be explored before the looming deadlines with consequences of liquidated damages and project delays. For this reason, it is recommended that a project milestone of an approved test procedure be established very early in the project; if this is concurrent with the design submittals, then it is likely that these issues can be avoided.

6.6 Summary

This chapter has considered the testing requirements for a typical ITS device from a hardware perspective and outlined a testing program that should be employed in stages depending on the maturity of the product. It has offered some guidance as to the elements of a good testing program and addressed some of the issues associated with that program.

However, the testing program is dependent on the procurement specifications. The procurement specifications must establish the requirements for the contract deliverables and the testing program, determine the consequences of test failure, and identify the schedule and cost impacts to the project.

Where the vendor provides evidence of prior testing, the agency should review these previous test results and independently determine if the testing was sufficient for the immediate procurement specification. The agency should also require the vendor to prove that the product tested is the product being offered. The agency should also contact current users of the device to determine their operational experience.

Further, software (embedded firmware) is generally an integral part of the ITS device. Great care must be taken when accepting a device with software changes that have not undergone a complete re-test for all functionality and performance requirements. While the software changes are unlikely to affect the environmental performance of the unit, any change could have unforeseen consequences and may warrant re-testing the unit. The requirement to re-test the unit subsequent to a software change should be clearly stated in the procurement specification.

Finally, testing can be expensive for all parties; the agency must weigh the risks associated with the use of the device and the device's track record before undertaking a complete test program. If the device is of a new design or custom to the agency, then the full testing program is warranted. On the other hand, if the agency is only purchasing a few standard devices for routine field deployment and the device has already been tested according to the NEMA testing profile or is on a specified QPL, the risk is relatively low.

A word of caution: there is no "NEMA certification" for ITS devices. The term "certification" should not be used by any vendor to claim that their ITS device is *NEMA certified*. NEMA TS2 (and also TS4) present a test procedure and environmental requirements (electrical, temperature, humidity, shock, vibration) and describe what is expected during the test. Vendors must construct a test environment for the device application (e.g., a lamp panel, detector inputs, central communications tester) that can demonstrate the operation of the unit, and then submit the unit to an independent testing laboratory to actually perform and monitor the testing. The independent testing laboratory typically provides the temperature and humidity chambers and instrumentation for monitoring both the DUT and the test environment and provides a certification as to the authenticity of the testing and the logged results. However, it is up to the vendor to instruct the testing laboratory in how to operate the equipment and how to determine "proper" operation. The testing laboratory simply certifies that they measured and observed the recorded results. NEMA does not certify products.

7. Software Testing

7.1 Overview

This chapter addresses the testing of the TMS software that resides on and executes from central computer systems (or servers) and personal workstations within the traffic management center (TMC) as well as remote TMCs and user personal computer systems with communication access to the TMS.

There are two major classes of software: operating system software and application software. The following describes each of these and how they are used in a typical TMS.

The **operating system software** provides:

- The ***basic environment*** for command and inter process control, computational, communication, and data storage capabilities.

- The ***services*** for device handlers and communication interfaces that support the application software.

The operating system may also include basic third party software such as **relational database management software** (RDBMS – such as Oracle), or other middleware such as **Object Request Brokers** (ORBs), **backup** and **cluster** utilities, and **report generation utilities**.

The **application software** provides the traffic management capabilities for the TMS including:

- ***Field device management*** (traffic surveillance devices – which include vehicle detection devices and CCTV cameras as well as and traffic control and motorist information devices – traffic signal controllers, dynamic message signs, and highway advisory radio towers and beacons).

- ***Centralized services*** (for incident and congestion management, traffic control and ramp metering coordination, graphical user interface and information display screens, video routing and display control, regional map displays, and relational data bases).

- ***External user support*** (for center-to-center communications, information sharing, and device control).

Computer operating system software and some *standard product* application software (e.g., relational database management) are typically referred to as commercial-off-the shelf (COTS) software products. COTS products are designed for general use and many serve a number of different and diverse types of applications (e.g., banking, engineering, transportation, etc.). Also, most ITS device vendors have developed *standard product* software to provide for command and control of their devices. In most cases, this COTS software has undergone rigorous testing for a wide variety of different application environments and represents a large installed base of proven software. COTS software will usually not need to be tested to the same extent that *modified* or *new* (custom) application software that is modified or designed for your specific application and requirements. However, COTS products must be uniquely configured for installation and operation on your systems, and to support the other non-COTS (modified and custom) application software that provide the traffic management capabilities of the TMS. Therefore, COTS products must be included in software test program.

All software proposed for use in your TMS should be subjected to testing before it is accepted and used to support operations. The extent and thoroughness of that testing should be based on the maturity of that software and the risk you are assuming in including it in your operations.

For most agencies, the bulk of software testing, at various levels of completeness, will be done at the software supplier's facility and the agency will not participate. It is suggested that the procurement specifications should contain provisions that give the agency some confidence that a good suite of tests are actually performed, witnessed and properly documented. This may require the software supplier to provide a description of their software quality control process and how the performance of the tests for the agency's project will be documented. The agency should review the supplier’s documented quality control process and sample test reports, and be comfortable with the risk associated with accepting them. Where possible, the agency should plan to send personnel to the developer’s facility periodically to observe the progress and test the current “build” to ensure that the translation of requirements to code meets the intended operation.

The following sections discuss what software should be tested and when that testing should occur. These sections also describe software test scheduling considerations and other considerations for a software test program.

7.2 What Types of Testing Should Be Considered?

The testing for a software product can be broken down into the following general categories:

- Design verification.
- Functionality.
- Prototype.
- Standards compliance (ISO, CMM, NTCIP, and others).
- Environmental.
- Maintainability.

Each of these will be discussed to gain a better understanding of what is meant and what is required for each relative to software testing.

The following describes the elements of a complete testing program based on the assumption that the software product being offered is a new design or custom product and hence should be subjected to all aspects of requirements verification. After this initial discussion of the most intensive case testing program, this guide will consider what steps can be eliminated or minimized for *standard* products and *modified* products (see Section 5.3.2).

7.2.1 Design Verification

New or custom software will require design verification testing. The agency's procurement specification should include design requirements for developing the software and verifying the design at various points in the development and implementation phases. Typically, the procurement specification will reference or invoke process standards for how the design itself is to be developed, documented, and reviewed and implementation standards defining how that design will be implemented and tested (i.e., transformed into an operational requirements compliable product). While the agency's information technology (IT) staff may be both knowledgeable and experienced in software design and development, it is recommended that the agency hire a qualified and experienced ITS consultant or systems integrator to assist in the development of the software procurement specifications. When the acquiring or operating agency has extensive experience in software development and implementation on the scale required for a successful TMS project, this recommendation can be evaluated with respect to the internal staff's existing workload.

Standards help assure compatibility between systems (both hardware and software) and promote multi-vendor interoperability and ease of integration. However, if the agency's requirements include compliance with standards, then there must be test procedures to verify that compliance (see Section 7.2.6). With this in mind, the following describes where standards apply and how they are selected for design and implementation. Because of their importance to the development of a TMS, selection of manufacturer's extensions to the NTCIP standards are discussed here as well. An example DMS specification is also provided to illustrate how the NTCIP standards are applied.

7.2.1.1. Selecting Standards for Design

The software development plan (SDP) will detail the approach to software requirements analysis, design, and coding techniques for the development the TMS software. This plan describes the software development process, establishes the design, coding techniques and standards to be used, and details the development environment and configuration management practices to be followed. Because software development is an ongoing process that continues throughout the life cycle of a software system, the SDP will need to be updated periodically to reflect changes made to the development, test and operational environments, and applicable development procedures and standards. The procurement specification should establish who (the software developer, a system integrator, or the agency) develops the SDP. The procurement specification should also state that the agency will review and approve the SDP prior to the start of software product acquisition (i.e., COTS or ITS standard products) and the development of modified or new software. The software developer (and most system integrators) will have an in-house software development plan that includes software design standards. Instead of starting from scratch, the agency may wish to adopt a developer's or integrator's in-house plan after it has been updated and revised to reflect the specific requirements of this TMS project. The agency approved SDP should govern all software development for the TMS.

This plan should include, at a minimum, the following software design standards:

- Coding standards.
- Graphical user interface standards.
- Geographical information standards.

7.2.1.2. Selecting Standards for an Implementation

The agency should inform the developer which standards to use in an implementation via the procurement specification; however, the agency (or its consultant) and the developer must be familiar enough with the standards to ensure that the implementation is consistent with the intended purpose of the standard. For example, the Common Object Request Broker (CORBA) standard typically invoked for object-oriented code development is not intended for field device communications using the “object definition standards.” Another example would be a request for a particular function for which none of the standards have a corresponding data element(s).

Frequently, implementation standards will also include requirements for specific COTS software packages such as operating systems and databases. When the acquiring agency already has internal support for those COTS software package and desires to minimize its long-term support costs, it should consider requiring that the implementation be based on those existing internally supported software packages.

Another reason for requiring specific COTS software packages is the packages’ conformance to other technology standards. For instance, the acquiring agency may specify an open standard such as POSIX and require that the operating system be POSIX compliant.

One word of caution; such selections must be explicit to reference specific versions of the COTS packages and recognize that such requests may be incompatible with the hardware platform since the COTS products tend to be constantly upgrading along with the hardware. Further, mandating specific COTS products may significantly increase the development costs and establish the need for ongoing software maintenance contracts at a considerable cost to the agency. As a result, each COTS “requirement” in the specifications should trace back to some important department/agency benefit.

7.2.1.3. Selecting Manufacturer Extensions

The agency must also determine whether the features of the selected standard(s) support all of the functional features that are being required for a particular ITS device. The NTCIP standards and framework are of particular importance to the development of a TMS. They are designed to allow for innovations to keep pace with advances in ITS technology; however, these standards do not currently define standardized data elements for every technology or functional feature of every device. In fact, these standards were designed to allow for future flexibility with the addition of custom “objects” and special, non-standard features.

The developer, acting for the agency, must determine if there are special features of the subject device that are not yet standardized by the NTCIP. If such features are present, then the developer will need to determine precisely how these features will be supported without conflicting with the standardized implementations. (It should be noted that the use of manufacturer specific extensions might tie the agency to a single source of software for all similar devices in the system.) Usually, this adaptation is accomplished by simply extending the capabilities of existing features of the standard, or by defining additional data elements or features under a developer-specific or agency-specific node for these specific management information base (MIB) extensions. It is important that the agency be aware of the use of these benign extensions and request that the systems developers or integrators clearly identify these in their implementation.

Another style of extending the standard might be based on replacement of a partially incomplete feature with a complete custom feature—this would be considered an unsupportable or malignant extension as it defeats the purpose and goals of any open standardization effort: interoperability and interchangeability. An implementation that uses benign extensions is likely to achieve a level of conformity with known exceptions; for example, where the specific extensions are listed. However, an implementation that includes unsupportable extensions, for example, replacement of the standard's features with custom features, will not achieve conformity as this would mislead customers and negatively impact the ability to achieve interoperable and interchangeable ITS products.

In any case, if specific benign or malignant extensions have been introduced and the agency wants to have the associated functions available in future purchases of the same device type, it is imperative that these extensions are made part of the agency's specifications and documentation deliverables. This requires that the agency obtain redistribution and/or re-use rights to these MIB extensions, even if the original manufacturers, vendors, or integrators developed and implemented them. Additionally, the agency should obtain both electronic and paper copies of the entire MIB, including the manufacturer-specific extensions. Negotiating the rights for re-distribution and/or re-use, along with documenting the requirements for MIB delivery, is much easier to complete up front in the procurement process rather than after the fact. The procurement specifications should include provisions that specifically address this eventuality.

These same concerns hold true for other COTS software products. There are usually base standards (such as CORBA) that are industry recognized. However, some vendors will enhance their product with **non-standard extensions**. Programmers tend to like this approach as it simplifies their work, but the cost can be very high when it becomes time to upgrade or migrate to newer platforms. The system becomes wedded to a specific vendor and if the same extensions are not available in the next version or conflict with the standards, then upgrading becomes very costly and time consuming. Where standards are invoked (Web services, CORBA, etc.) it is important that only the basic standard be used and that custom extensions be avoided.

7.2.1.4. Development Resources

There are wide varieties of resources available that relate to the NTCIP standards. The following lists some of the resource materials that have been used in the development process and early implementations, as well as the location of developed materials.

Websites

A wide range of documentation is available on the World Wide Web NTCIP Home Page located at *www.ntcip.org*.

The site currently includes such items as:

- NTCIP guide (9001).
- NTCIP profile publications.
- NTCIP data element definitions for a variety of devices.
- NTCIP case studies.
- Various white papers written during the development of the initial standards.
- FHWA-sponsored software packages, for example, NTCIP demonstration, NTCIP Exerciser and NTCIP Field Devices Simulator.

Other web sites of interest are shown in the following table.

These sources provide copies of the various standards and the TMDD guide describes the process of selecting and configuring the standards for use in procurement specifications. There currently is no testing program tied directly to the NTCIP and related standards. The testing and conformity assessment working group (TCA) developed a testing documentation guide, NTCIP 8007, and a user guide to NTCIP testing – NTCIP 9012. These can be used by public agencies and integrators as guides to the development of test procedures for the evaluation of both central software and field devices.

Table 7-1. NTCIP-Related Web Sites

Web Site	Address	Description
NTCIP	*www.ntcip.org*	The official web site for NTCIP and related publications and information
DATEX-ASN	*www.trevilon.com/library.htm*	The web site for DATEX-ASN documents and information
DATEX-Net	*www.datex.org*	The web site of the DATEX-Net Standard currently in use in Europe.
IANA	*www.iana.org/numbers.html*	The Internet Assigned Numbers Authority web site.
IEEE	*standards.ieee.org*	Links to all of the IEEE standards efforts, including ATIS, Incident Management, Data Dictionaries and Data Registries.
ISO	*www.iso.ch*	The Official ISO home page.
ITE	*www.ite.org*	ITE web site – go to the technical area and standards which include the TMDD and the ATC standards.
ITS America	*www.itsa.org*	The home page for ITS America.
NEMA Standards	*www.nema.org/index_nema.cfm/707/*	Site for ordering NTCIP standards. This is also a site for ordering the commonly used NEMA standards such as TS4 and TS2.
RFC Index	*www.nexor.com/public/rfc/index/rfc.html*	A search engine for all of the Internet RFCs.
SNMP	*www.cmu.edu*	A library of information on SNMP and related topics.
TCIP	*www.apta.com*	The home page for Transit Communications Interface Profiles.

Sources of Public Domain Software

There are two basic prototype implementations of NTCIP software. Neither of these packages was designed to operate a real system; rather, they were designed to provide tools to the industry to test equipment submitted as being compliant with a specific protocol. Unfortunately, there is no ongoing program to maintain these packages. They are available with documentation for downloading at www.ntcip.org/library/software/. Integrators may find them useful as a reference but these are not intended as products since their development was halted and has not kept up with the latest development in the NTCIP standards arena.

NTCIP Exerciser Software, Build 3.3b7a

This NTCIP Exerciser is able to read in a properly formatted management information base (MIB) from a floppy disk and support the exchange of fully conformant NTCIP messages under the direction of the operator. The package supports the creation of simple macros to enable the user to perform a number of operations sequentially and to record the results. The current version supports the simulation of either a management station (funded by the FHWA) or an agent (funded by Virginia DOT). It currently supports the STMF Application Profile (SNMP only), Null Transport Profile and both the PMPP-232 Subnetwork Profile and the PPP Subnetwork Profile. The most recent version of this software is available for free on the NTCIP website. It is designed for Windows NT.

Field Device Simulator (FDS), Beta 2

The FHWA also developed a DOS-based program to emulate a field device that supports the data elements contained in the global object definitions. This program supports the STMF Application Profile (SNMP-only), the Null Transport Profile and the PMPP-232 Subnetwork Profile. This software is available for free on the NTCIP website.

Application of the NTCIP Standards

Appendix D contains an example application of the NTCIP Standards.

7.2.2 Prototyping

A valuable tool in design verification for new and custom software is the development of prototype software. Prototype software allows design concepts to be verified before committing resources to developing the complete code unit that will implement that design. Prototype testing reduces the development risk, particularly for a new design, and increases the chances that the design can be successfully implemented. The procurement specification should require prototype testing for:

- Communication protocols.
- Device handlers (drivers).
- User interface.

Communication protocols and device handlers typically have very detailed data handling and timing requirements. The user interface (displays, GUI screens, and GIS maps) has a very demanding common “look and feel,” and interactive response requirements. Prototype testing will require that test procedures be developed by the developer and approved by the agency to verify that these requirements are being met. Prototype testing also allows the agency to provide design feedback to fix undesirable aspects of the design before full implementation. It should be noted that prototype testing has the potential to uncover missing, incomplete or inadequate requirements. Identifying and resolving these deficiencies early may necessitate a revision of the requirements specification. However, it avoids additional or more complex testing, and changes to delivery schedules and project costs, and therefore mitigates their programmatic impacts.

One potentially significant drawback to prototype testing is “requirements creep.” That is, once the developer has revealed the design and proposed implementation (at least partially) with a prototype, the agency (or its consultant) may not like what they “see” or say that it is not what they “expected, ” even though the implementation meets the current specification requirements. This can occur if there is a misunderstanding of the requirements or the requirements were poorly written (vague or ambiguous). This can also occur when people from a completely different frame of reference (programmers vs. traffic engineers) interpret the requirements. A requirement change will typically result in additional cost to the agency (even if the feature or capability is reduced or eliminated). Most agencies are well aware of this fact and will try to get the developer to accept their interpretation of the subject requirements as written (without contract changes). Many developers will accede to the agency’s interpretation, make the necessary design or implementation changes, and proceed with development as long as those changes are relatively minor and don’t significantly impact the developer’s schedule and costs. Unfortunately, this practice leads to undocumented—or documented but not delivered and, therefore, un-verifiable—requirements. Therefore, it is important that the requirements as well as test procedures be updated based on such changes. Prototyping is encouraged whenever the agency cannot “see” the planned implementation prior to deployment and is particularly important for reports and user interface interactions. One word of caution: it is important that the performance of the prototype be no better than the performance expected from the production system.

The agency must be cognizant of the fact that requirements will change during design and even during development and implementation. These changes should be managed through the configuration management process so they are fully documented, their impacts are assessed, and changes made to the appropriate requirements documents, test plans and procedures, schedules and procurement specifications or purchase orders. These will change the baseline against which the final system is judged.

Since prototyping is significant value, the project work plan and schedule need to allow for both time and budget for possible changes to be introduced into the system. Even when “off the shelf” products are used, an agency may request that minor changes be made to accommodate their specific needs. As long as the changes are documented, evaluated for cost and schedule impact, the process is manageable.

7.2.3 Environmental

Environmental testing verifies that the product operates properly (i.e., meets requirements) in the installed environment. For software, the “environment” is the target processor (platform or server), and for non-operating system software products, it is the installed operating system. This aspect of testing is usually the most extensive and complicated required for any product. For large complex systems such as a TMS, there are two distinct test environments: the development environment and the operational (field or production) environment. Each of these environments and the kinds of product testing that can be accomplished is discussed below.

7.2.3.1. Development Environment

The development environment is usually established at the software developer’s facility and will utilize host processors, operating systems, and software development tools that the developer has or acquires specifically to support this TMS project. It will also include office space for the development staff and support technicians; and an equipment room to house the processors and computer peripheral equipment (e.g., monitors and keyboards, printers, local area network devices, etc.), equipment racks and floor space for representative TMC equipment (e.g., workstations and display devices), and typical field devices (e.g., communications devices, traffic controllers, CCTV cameras, etc.). The robustness of the development environment and the extent to which it is representative of the operational environment will dictate what testing can be accomplished.

Development environments are expensive to setup, operate, and maintain. The agency's procurement specifications must clearly define what is expected, when and what costs will be borne by the agency. For example, operating systems, relational databases, and software development tools are vendor-licensed products (either one-time or renewable on an annual basis). These product vendors constantly issue updates to fix problems and upgrades to add new features and capabilities to their products. The updates in the form of patches are free as long as the software product license is current. However, upgrades in the form of new releases are provided at extra cost, although they are usually offered at a reduced price to current licensees. Most vendors will only provide maintenance support for the last three software revisions. Developers will often have their own software licenses, but eventually the agency will have to assume these costs for the production system. The decision to accept and install a patch or purchase and install a new revision is not always straightforward, but will have definite downstream ramifications particularly with respect to testing. Software tested in the pre-patch or pre-upgrade environment will have to be re-tested in the new development environment.

The development environment should be separate and distinct from the operational environment, particularly for a project that will be incrementally deployed. Its configuration (both hardware and software), like the operation environment, should be well documented, and all changes to that environment configuration managed through the configuration management process. A separate and distinct development environment accomplishes two very important goals. First, it allows the software to be tested from prototype versions through the latest build release in a controlled environment (i.e., with repeatable simulated inputs and events) without impacting ongoing operations with the production system (e.g., no system crashes that may disrupt traffic flow or affect incident response, and no loss of operational data while the system is down). Second, it prevents polluting the production database with simulated data (e.g., DMS test messages; simulated volume, occupancy, and speeds; congestion and incident events and test response plans; and event logs).

The development environment will also be used to further investigate, resolve, and test solutions for software or hardware/software problems found in the production environment. To maintain configuration control, all problems, whether found in the production system or development system, should be recorded on the system problem/change request (SPCR) forms and processed by the change control board under the configuration management process.

The development environment must be sustained even after final system acceptance, at least at some minimal level, for the life of the TMS in order to maintain the software and develop enhancements. Whether the development environment is left with the developer, a system integrator, or transferred to the agency's facilities will depend on what level of system maintenance and future enhancements the agency is comfortable with and can staff with its own personnel and/or a maintenance contractor. Note that transferring the development environment to the agency will also necessitate the purchase of the appropriate software licenses as the developer is likely to need to retain the original licenses to ensure the ability to support their maintenance obligations.

It is recommended that even if the development environment is maintained at the developer's facility, the agency should consider the purchase of a test environment at the TMC. This is invaluable as a training aid and for the evaluation of future updates by the agency. Such as system must include simulators and/or connections to the production system to support data collection and analysis that matches the production system. The more robust the test system, the more likely the agency will be able to minimize disruptions when the new software is installed on the production (operational) system. However, as noted above, the agency will have to purchase software licenses for all COTS products. It is best if this requirement is included as part of the procurement specifications so that the issue of license costs and the hardware configuration will more closely match that of the operational environment. Note that as upgrades are required for the operational hardware, the test environment must also be updated so that it always reflects the current production system.

7.2.3.2. Operational Environment

The operational environment is the actual TMC and field environment that the delivered software will be installed on and operated in for the life of the TMS project. Software acceptance testing will be performed in this environment under actual operational conditions using the system's communication infrastructure and installed field devices. This acceptance testing includes:

- Software build releases (of the latest TMS software version that includes new features and functionality, SPCR fixes, and operating system patches or new versions).
- Hardware/software integration.
- Final system acceptance testing.

This may be the first opportunity to fully exercise and verify features and functionality not possible in the development environment. Expect some problems to be uncovered in testing in the operational environment; however, they should not be *showstoppers* if prior testing in the development environment was detailed and thorough. The last two versions of previously tested and accepted build releases of the software and their associated databases should be retained in case the new version experiences a problem that cannot be tolerated. This allows a previously accepted version to be re-installed and operated until the new version can be fixed. Note an older version may not have a capability, feature, or functionality that is desirable and working in the new version, so it is a judgment call whether to live with (or work around) the operational problems in the new version or revert to a previous version. The new version can be conditionally accepted by the agency, with the understanding that final acceptance of this version will be withheld until the problem is found, fixed and re-tested. Again, it is paramount that strict configuration management processes be followed in this environment. Undocumented and unapproved changes to the configuration will compromise testing that has been accomplished as well as testing that has been planned.

Each time a new build release (or a previous release) of the software is installed in the operational environment, major elements of the system's functionality must be shut down or suspended. The software installation takes time (sometimes hours) and some level of regression testing (see section 4.4.8) should be performed before trying to restart it to support operations. Acceptance testing of SPCRs, new features and functionality, etc. can usually be delayed until a more convenient time. The immediate need is to complete the regression testing to verify that all of the original functionality is uncompromised. The impacts to ongoing traffic management operations will depend on the duration of the operations interruption and what operational backup or fail-over capabilities have been incorporated into the TMS design. Suffice to say, this is typically a major disruption to ongoing activities and should be performed when the impacts may be minimized (i.e., at night or on a weekend). It is also likely, that the agency will want to "collect" a significant number of "changes" before updating the production system.

7.2.4 Functionality

Functionality testing verifies that the software performs all of the specified *operations* listed in the requirements. At the software component and integrated software chain levels, this testing can usually be accomplished using test software to simulate inputs and verify outputs. That is, to verify data handling and computational requirements such as the proper storage and retrieval of DMS messages in the database and verification of incident detection algorithms. At higher levels of testing such as hardware/software integration and system testing, some representation of the system's hardware, communications infrastructure, and field devices will be required to verify operational requirements.

Software simulators of hardware interfaces and field devices are sometimes used if they exist or can be cobbled together without their own extensive development program. These are usually of limited value except where there are long lead times for actual system hardware and field devices. In that instance, software simulators will allow software functional testing to proceed (to raise confidence levels that the design works as intended or to uncover potential problems), but much if not all of this testing will have to be repeated with the actual hardware and field devices.

A note of caution here – software simulators will themselves need to be verified (against design and implementation requirements reviewed and approved by someone – preferably the agency). If any test results or analyses derived from their use is suspect, it could lead to solving problems that don't really exist and missing those that do. It's much better to use actual hardware and field devices when testing software functionality. However, controlling the test conditions, particularly for field devices, can be daunting unless testing is performed in the development (or test) environment.

When dealing with functionality testing involving communications subsystems, it can be difficult and expensive to build and maintain simulators. One means to reducing these costs is to specify that device vendors provide an additional special firmware package that responds to central polls for all addresses on the channel except the address configured for the device. This allows the system to communicate to all channel addresses using only two devices. The central system database can be configured with one database configuration for all devices on the channel. This configuration can be used to test communication protocols, communication performance, and system response to major simultaneous events. Care must be taken with this type of configuration to ensure that the field device has the processing resources to support the communications traffic and that the special firmware stays within the development/test environment. Accidentally deploying this modified firmware in the production environment would result in major operational issues for the system.

As with hardware functionality testing, software functionality testing will also be extensive, but it is virtually impossible to completely verify the required functionality under all possible combinations of expected operational conditions. At the system level, the test plans and procedures should address a few of most likely and potentially demanding operational circumstances to verify both operational and performance functionality. Examples include operations during peak traffic conditions with multiple simultaneous incidents, interoperability between multiple TMCs (if applicable), and following a failover recovery at a backup TMC and subsequent transfer of control back to the primary TMC. This will also be a good time to assess the agency's operational procedures, staffing levels, and operator training. The number and complexity of test scenarios to use in the verification of functional requirements and the time and budget devoted to this effort must be weighed against the risk of having a problem that goes undetected because some combination of possible operational conditions wasn't included in the testing scenarios.

When attempting to determine what testing is important, one might consider some of the following:

1. Can multiple operators view the same incident tracking form?

2. Are the operators prevented from modifying it while another operator handling the incident enters updates? In other words, does the system handle operator contention for the same data or resource?

3. When operating the pan, tilt, and zoom controls for a remote CCTV camera, does the video image from that camera indicate the camera is responding without a significant time lag to the commanded operations?

4. How do the network loading and the number of concurrent users affect the operation of the system?

5. When workstations "crash" or when there are network disruptions, do the servers recover when the workstation is restarted or the network is restored? Is it necessary to restart the whole system?

6. If the system experiences a "problem," are the operators alerted to the situation so they can take corrective action?

These scenarios represent typical operational issues that the system design should accommodate. Requirements should be derived from these scenarios during the design process so that they can be properly incorporated into the system design and test planning.

7.2.5 Performance

In addition to functional testing, verification of software performance requirements is also required. *Performance* requirements specify things such as the interactive response time between an operator command input and a display update or the change that comes as a response to that command input, the maximum time interval between regional map display updates, and the minimum number of traffic controllers that the system can effectively control and coordinate. Performance requirements are qualitative (i.e., measurable) and apply under specific environmental conditions. For software performance requirements, the applicable environmental conditions typically relate to the operational environment as opposed to the development environment and apply to the quality and timeliness of the service provided to the end users. Verifying performance requirements will most likely require making accurate measurements and performing some quantitative analysis.

The following are some examples of desirable performance characteristics/requirements that need to be addressed (and verified) to maintain and assure responsiveness to operator actions and provide for near real-time command and data exchanges between the traffic management center and the various system hardware and software components.

Graphical User Interface (GUI) Control Screens – the primary operator interface is the GUI control screen. It is imperative that the operators receive timely if not immediate feedback to mouse button and keyboard entries. Where there is an expected delay of more than two or three seconds between the operator entry and the command response, some mechanism needs to implemented to let the operator know that the command has been received and is being processed. A simple response showing the button depressed/released, shading or color change and even the hourglass wait symbol is sufficient. Without this mechanism the operator may continue to enter additional unnecessary and probably undesirable inputs.

Closed Circuit Television (CCTV) Camera Control – interactive control of the pan/tilt/zoom, iris and focus features and today’s Internet Protocol (IP) streaming video will place the highest demands on the communications infrastructure. The video image feedback during camera control operations should be as close to real-time as possible to avoid command overshoot.

Detector Data Acquisition – vehicle detection stations are typically capable of storing data locally and transmitting the data from the field devices in response to a polling command. In order for this data to be effectively utilized by the system's incident and congestion detection algorithms, a polling cycle of approximately 20 to 30 seconds is necessary, although some systems with lesser bandwidth to the field devices may fall back to once per minute.

Automated Vehicle Identification and Location Systems – data from these system sensors is time sensitive but is typically stored and time tagged by the field device. The system need only poll these devices as needed to effectively utilize their data before the local storage buffers overflow.

Traffic Signals, Lane Control Signs, Dynamic Message Signs, Highway Advisory Radio, etc. – these system devices are commanded to change timing patterns, messages, etc. as necessary, but not at a once-per-second command rate. Timing plans and messages are downloaded at nominal data rates typically requiring many seconds to complete a download. They are cyclically polled to check control mode and current status. The specifications should state the requirements, and the system design, primarily the communications infrastructure, should ensure that these requirements can be met through the distribution of devices on various communications media.

7.2.6 Standards Compliance

Where the procurement specifications require that the software comply with a specific standard, the test plan and test procedures must include steps to confirm that compliance. Software standards typically refer to processes that must be followed, such as coding for inter-process communications, communication protocols, and documentation. Verification that the process standards are met can be accomplished by inspection (i.e., at design reviews and code walkthroughs), by analysis of data exchanges between software components and between software components and hardware devices, and by demonstration; for example, a requirement that the GUI screens all have the same attributes (e.g., color, shape, style) with respect to pull down lists, pop-up windows, etc. Other process standards refer to the methodology for the software development process itself and relate to quality assurance. Some of these are listed below. Verification of compliance with these standards is accomplished by inspection.

The following are software development, test, and quality assurance standards that should be considered for incorporation (by reference) in software procurement specifications. Requiring certification or compliance with these standards does not ensure a quality product, but does serve notice that the agency expects the required certification and/or rating level to be maintained by the development organization and that required documentation, test and quality assurance standards be met. Where the integrator or developer does not hold the “certification,” they should be required to provide documentation of their software development processes, testing processes, and configuration management procedures. What is important is that the developers have such procedures and follow their internal procedures rather than any specific certification.

This material is included here because it falls to the agency to conduct "tests" or inspections to verify that these requirements are being met.

ISO 9001:2000

The International Organization for Standardization (ISO) 9001:2000 quality standard addresses quality systems that are assessed by outside auditors. It applies to software development organizations (as well a many other kinds of production and manufacturing organizations) and covers documentation, design, development, production, testing, installation, servicing, and other processes. A third-party auditor assesses an organization and awards an ISO 9001:2000 certification (good for 3 years, after which a complete reassessment is required) indicating that the organization follows documented processes.

Note that this is an expensive process and it is not sufficient that the "firm" have this certification, it is important that the specific group developing the TMS software be certified or have at least established the appropriate quality control procedures.

CMMI

The Capability Maturity Model Integration (CMMI)[33] developed by the Software Engineering Institute at Carnegie-Mellon University is a process improvement model that determines the effectiveness of delivering quality software. The model has five levels of process maturity defined as follows:

Level 1 – Characterized by chaos, periodic panics, and heroic efforts by individuals to successfully complete projects. Few if any processes in place – successes may not be repeatable.

Level 2 – Software project tracking, requirements management, realistic planning, and configuration management processes are in place; successful practices can be repeated.

Level 3 – Standard software development and maintenance processes are integrated throughout an organization. A Software Engineering Process Group is in place to oversee software processes, and training programs are used to ensure understanding and compliance.

Level 4 – Metrics are used to track productivity, processes, and products. Project performance is predictable, and quality is consistently high.

Level 5 – The focus is on continuous process improvement. The impact of new processes and technologies can be predicted and effectively implemented when required.

Organizations can receive CMMI ratings (equivalent to one of the five levels) by undergoing assessments by qualified auditors. (A minimum CMMI rating of Level 2 is recommended if compliance with this standard is required.) Again, this is an expensive process, and what is important is that the Quality Assurance procedures are in place and are followed by the specific software development team that will be working on the TMS software.

[33] Formerly known as Capability Maturity Model (CMM).

IEEE/ANSI Standards

The Institute of Electrical and Electronics Engineers (IEEE) in association with the American National Standards Institute (ANSI) creates software related standards such as the IEEE Standard for Software Test Documentation (IEEE/ANSI Standard 829), the IEEE Standard of Software Unit Testing (IEEE/ANSI Standard 1008), the IEEE Standard for Quality Assurance Plans (IEEE/ANSI Standard 730), and others.

7.2.7 Software Maintenance and Maintainability

Software maintenance involves implementing changes to a controlled software baseline (release version) for the purposes of correcting errors (i.e.,bug fixes), adapting to the system's changing environment (e.g., external interfaces) and implementing enhancements (i.e., adding new features and revising or deleting old ones). Once an operational version of the software is placed under configuration control, all changes, whether corrections, deletions, or enhancements, should first be recorded on an SPCR form and submitted to the configuration control board (CCB) for approval.

There are three categories of software to be maintained: commercial-off-the-shelf (COTS) products (e.g., operating systems, databases, communications middleware, development tools, etc.), ITS standard product software, and your system's unique software. Each requires a different maintenance and acceptance test approach.

COTS software is typically maintained by the manufacturer or developer under a written maintenance agreement or product use license and is warranted to the original purchaser. During the term of the agreement and renewal periods, if any, the manufacturer will advise of known and reported problems and the necessary corrective actions, which may include patches, partial updates, or new releases. The COTS ***user*** (i.e., the agency) is responsible for implementing the corrective actions unless specifically covered by the maintenance agreement. To receive manufacturer support and receive reduced prices for upgrades and new releases, it is very important to execute software maintenance agreements prior to expiration of the initial warranty period and renew extensions of the agreements before they lapse.

Maintenance of operational software that is acquired from other TMS or ITS libraries and TMS consortiums is the responsibility of the consortium members and specifically the developer of the components and/or modifications and enhancements to them. Software updates are typically made available upon request by consortium members, who must then deal with any compatibility and release level issues, and the integration and installation of the updates on their respective systems and environments. Because the acquiring agency will be responsible for integration, installation, and maintenance of this software, all available documentation, including requirement specifications, design, code listings, installation and test procedures, test results, and user's guides should be requested and reviewed before attempting to include software from these sources in your system. Missing, incomplete, or inadequate documentation will have to be generated and/or brought up to your system's standards in order for that software to be acceptance tested, brought under configuration control, and maintained. There may be restrictions on the use or further distribution of this software or licensing agreements for its use. These potential issues should also be carefully considered with respect to how they may affect your intended usage.

System unique software is all of the operational software developed or explicitly modified for use in your system. By agreement with ITS libraries and TMS consortiums, system unique software is usually made available to ITS libraries and for use by TMS consortium members at their own risk.

Typically the system software development team (comprising agency and software developer or integrator personnel) performs software maintenance for the system unique software at the direction of the configuration control board, and in accordance with the configuration management plan and the system maintenance plan.

In order for maintenance activities to be carried out effectively and efficiently, it is important that the procurement specification include some software maintainability requirements. Software maintainability requirements are typically included in the software development plan. They are verified by reviewing compliance with applicable design and implementation standards, documentation requirements, and the procedures in place for acceptance testing and configuration management. The ability to maintain the software requires extensive knowledge of the software architecture as well as the build environment. Build processes transform the source code into distributable files and packages them for distribution and installation on the system's various computing platforms.

Specific maintainability requirements that should be included in individual COTS software product specifications include high-quality documentation (user guides, operations and maintenance manuals), the ability to readily configure the product for the intended operational platform and TMS application, and technical support for product installation, initial operations, and problem analysis, including help desk telephone numbers. For example, for geographical information system (GIS) map display software, the procurement specification should require the vendor to assist the agency and the software developer or system integrator in implementing the base map. This should include defining map zoom characteristics and map display layers showing various static features, dynamic or animated features, and refreshing and distributing the GIS map display throughout the TMS operational environment.

Because the base map and regional features change over time, the procurement specification should include provisions for periodically updating the base map. COTS vendors will typically cap the technical support by limiting the number of support hours (both off-site and on-site, including labor, travel, and per diem costs for their technicians) and/or provide a quoted rate for that support. However if technical support is covered in the procurement specification, some acceptance criteria for that support should be specified such that the product can be accepted or rejected as unsuitable for the intended purpose.

For non-COTS software, i.e., modified or new system unique software, the procurement specification should include provisions for help desk and on-call maintenance. Here the criticality of the product to ongoing operations will dictate the specific provisions necessary. A shorter response time will be necessary for the basic TMS operations and control features and on-call (emergency) support will be needed for major problems that cannot be resolved over the telephone using the help desk support. Discussions with other users of this or similar products from this software developer or system integrator can aid in establishing the initial provisions for help desk and on-call support to be included in the procurement specification, and these provisions should be adjusted in maintenance contract renewals based on actual operational experience. Again, some acceptance criteria for that support should be specified and in this case should include measurable qualifiers such as availability and adequacy of the expertise for help desk calls (24/7, initial response and call back) and specific response times (e.g., 2 hours for 8-5 non-holiday weekdays, 4 hours for non-holiday weekend days and weekday nights, and 8 hours for holidays) for on-call software support personnel to show up on-site ready to diagnose and resolve problems.

Note that what is important from this section is that the procuring agency needs to understand (and include) the specification requirements and conduct reviews and evaluations of the vendor's conformance to these requirements.

7.3 When Should Testing Occur?

In the previous section a discussion of each of the general categories of software testing was presented. In this section the chronology of software testing is discussed. Depending on the maturity of the product, not all of the test phases described below will be required. Each of the following discussions will provide some guidance on the testing required based on the maturity of the product.

7.3.1 Acceptance of Previous Tests

There will be few if any opportunities to consider acceptance of previous software test results that would be directly applicable to your TMS project. Even COTS software will have to be subjected to unit testing prior to acceptance for integration with other system-unique software. For a stand-alone, standard product application installed by the vendor, such as a DMS subsystem (running on its own platform with direct communication interfaces to the field devices that will not be integrated with other system software), the procurement specification should require a complete suite of both hardware and software functional and operational tests at the subsystem level. This testing should be conducted on-site following installation to verify all specification requirements. In this case, the vendor would develop the subsystem test plan and all subsystem test procedures, and the agency would approve them prior to conducting the acceptance testing. Since DMS testing for your TMS project would only be performed at the DMS subsystem level, the agency is effectively accepting the lower level software testing, i.e., unit, software build integration, and hardware/software integration testing performed by the vendor prior to shipping to the installation site.

For a mature stand-alone standard product that does not require custom software to meet the agency's specification requirements, agency acceptance testing at the subsystem level should be acceptable. If the agency plans to maintain the software in this subsystem, it should require full software documentation, including requirements, design, source code, implementation details, and test procedures in the procurement specification. Vendors will not want to provide proprietary documentation or source code for their standard products. So some provision to access these may be necessary to develop bug fixes and enhancements, and should be included in the procurement specification when applicable. One possible solution is to escrow[34] a copy with an independent third party. Previous software test results for ITS products acquired from an ITS library or other TMS projects should not be accepted (even if those products can be used "as is" without modification). These software products should be subjected to the same testing required for COTS products.

[34] This technique has been used to "hold" the source code for use by the agency in the event the supplier does not survive or terminates support for the product. What is very important is that the source code must be current – i.e., maintained at the same level as the operational system – and it must include a complete development platform including all libraries, compilers, linkers, loaders, and "build" environments – essentially everything necessary to convert the source code into the executable that is running on the production system.

7.3.2 Unit Testing

This is the lowest level of testing for computer software components delivered for software build integration testing. For modified or new components, the software developer conducts stand-alone software unit tests following design walk-throughs and code inspections. At this level, the software design is verified to be consistent with the software detailed design document. Unit testing is typically documented in software development folders. Functional testing may be limited due to the fidelity of the test environment (usually constrained by what is available in the development environment) and the ability to provide the required inputs and respond to outputs. Test software is often utilized to simulate these, particularly where it is necessary to verify specific data handling or interface requirements and algorithms. Receiving inspections and functional checkout are performed for COTS software to assure that these components are operational and in compliance with their respective specifications.

7.3.3 Software Build Integration Testing

Software build integration testing is performed on the software components that pass unit testing and are suitable for being combined and integrated into the deliverable computer software configuration items. Additional functional testing is usually possible at this level, especially for inter-process communication and response requirements. A software build typically consists of multiple items and is ideally tested in the development environment as opposed to the operational environment. This is not always possible due to the expense of duplicate hardware platforms and communications infrastructure. A software build that has passed this level of testing is called a build release. There may be multiple build releases that compose a new software version.

7.3.4 Hardware / Software Integration Testing

Hardware/software integration testing is performed on hardware and software configuration items that have passed hardware integration tests and software build integration tests, respectively, and that have subsequently been integrated into functional chains and subsystems. Hardware and software integration testing is performed to exercise and test the hardware and software interfaces and verify the operational functionality in accordance with the requirements contained in the specifications. Integration testing is performed according to the integration test procedures developed for a specific software (build or version) release and hardware configuration. Testing is typically executed on the operational (production) system unless the development environment is sufficiently robust to support the required interface testing.

7.4 Software Test Phases

In general, the software test program can be broken into three phases as described below.

1. **Design Reviews** – There are two major design reviews: (1) the preliminary design review conducted after completion and submittal of the high-level design documents and (2) the detailed design (or critical) review conducted after submission of the detailed design documents.

2. **Development Testing** – For software, this includes prototype testing, unit testing, and software build integration testing. This testing is normally conducted at the software developer's facility.

3. **Site Testing** – This includes hardware/software integration testing, subsystem testing, and system testing. Some integration testing can be conducted in a development environment that has been augmented to include representative system hardware elements (an integration facility) but must be completed at the final installation site (TMC) with communications connectivity to the field devices.

The following sections will further detail these phases and what to expect in each.

7.4.1 Design Reviews

For new or custom software, the procurement specification should require the software developer to provide both high-level and detailed software design documents. Following the submission of the high-level software design document, a preliminary design review is held to allow the agency to approve the design approach, basic architectural concepts, interfaces, and allocation of specification requirements to the configuration items that will become the contract deliverables. Agency approval of the high-level design document is required to proceed with the detailed software design.

The detailed design review follows the submission of the detailed design document that completes the software design (by detailing the design to the computer software component level) and the behavior and interfaces for all computer software components within their respective configuration items. Each computer software components defined is traceable back to the requirements allocated to a configuration item from the software requirement or procurement specifications. Approval of the detailed design document by the agency provides a go-ahead to the developer for coding.

Design reviews can be held at the agency's facility or the developer's. In either case, there will be travel and labor expenses associated with the reviews. A system integrator, if involved, should be required to review all design documents submitted, provide written comments to the agency prior to the review, attend both design reviews, and participate in the design approval process. The agency should integrate its written comments with those of the system integrator (if applicable) and formally provide them to the software developer at least 2 weeks prior to the preliminary design review and 1 month prior to the detailed design review. The software developer should be required to prepare a written response to each of the agency's formally submitted comments at least 3 business days before each review and to respond to comments recorded at the review within 2 weeks of the review. It is recommended that the detailed design review be held at the agency's facility to allow greater participation by agency technical and managerial staff. This will be the last chance to catch design oversights and shortcomings prior to committing to design implementation.

7.4.2 Development Testing

As defined above, development testing includes prototype testing, unit testing, and software build integration testing for single or multiple configuration items. This testing is normally done in the development environment established within the software developer's facilities. The agency or its system integration contractor will want to witness some of this testing, but probably not all of it. The agency may wish to review some of the software development folders (which contain development test procedures and test results) in lieu of directly participating in the development testing. The software developer should be required to provide test reports for completed tests and periodic test status reports detailing what testing has been completed and the status of testing to be completed and proposed schedules. Successful completion of development testing allows those components to be delivered for site testing.

It is important that the agency or its consultant review these test procedures to ensure that they adequately represent the intended system operation and are robust enough to actually stress the software and test “normal” and corrupted data and operator actions.

7.4.3 Site Testing

Site testing is testing that is performed at the final installation or operational site and includes hardware/software integration testing, subsystem testing, and system testing. For software, this typically involves installing a software build release on the operational platforms at the TMC(s), on servers at field locations such as communication hubs, and in field devices such as traffic controllers (as firmware embedded on computer processor or memory chips), and conducting testing to exercise and test the hardware/software interfaces and verify the operational functionality in accordance with the requirements.

System acceptance is typically accomplished in stages – hardware/software integration acceptance tests, subsystem acceptance tests, and finally system acceptance tests. Following system acceptance, regression testing is preformed for each new software release or the addition of new hardware components to assure that prior system performance and functionality has not been adversely affected by the new or modified code.

7.5 Other Considerations for a Software Test Program

The preceding sections described a complete software test program for the three general categories of software: COTS (including ITS standard products), modified standard products, and new or custom software. For most TMS projects, a considerable amount of new or custom software will be required to meet the agency’s total set of requirements. Even if standard ITS software can be acquired from others, new test procedures will be required to accept this software and integrate with software that will be specifically designed, developed, and implemented for your TMS project. In either case, the test program needs to be developed in concert with developing the procurement specifications. The agency should consider the following recommendations when developing the test program.

7.5.1 Who Develops the Test Procedures

The software developer should develop the test procedures for unit and software build integration testing with the agency’s right to approve. If a system integrator is involved in the TMS project, the system integrator should be required to develop an overall system test plan, test schedule, and all hardware/software integration, subsystem-, and system-level test procedures. The agency has the responsibility for final approval over those plans and therefore must carefully evaluate them to ensure that all of their operational requirements are included in the test procedure. The system integrator should also be required to develop regression test procedures (see section 4.4.8) and update them as system deployment progresses. Following system acceptance, the maintenance contractor should be required to update the regression test procedures as appropriate for system maintenance, enhancements, and expansion.

7.5.2 Cost of Testing Program

Because much of the software for the TMS will be new or custom, the software test program will be extensive and expensive. Depending on the robustness of the software development environment and its proximity to the TMC site, some costs associated with agency or integration contractor testing can be mitigated. To keep cost down, the agency should consider requiring (in the procurement specifications) that all COTS and ITS standard product software (including documentation) be sent directly to the software development facility rather than have the agency receive and then trans-ship it to the developer or integration contractor. The agency should also consider having hardware vendors "drop ship"[35] one or two factory acceptance tested products to the development site to allow for hardware/software integration testing. This is particularly useful, and cost effective, when elements of the final system can be utilized at least temporarily in a development environment for development and integration testing. Items that are no longer needed for the development environment can be shipped to the installation site or placed in the spares inventory. However, as previously discussed (see section 7.2.3.1), the development environment must be sustained even after final system acceptance for the entire life of the TMS.

7.5.3 Test Schedule

The procurement specifications should outline how the software test program fits into the overall project schedule. The integration contractor should be required to develop a detailed development and test schedule with the TMS test plan. This schedule must allow sufficient time for the agency to review and approve preliminary and detailed software designs and plans for setting up and operating the software development environment as well as reviewing and approving test procedures and witnessing acceptance tests. Because much of the software will be a new or custom design, it may take a year or more to develop. However, early deliveries of representative computer platforms and other system hardware elements (e.g., communication equipment and field devices that have passed factory acceptance testing) to the software development site can improve the overall development and testing aspects of the project.

Test planning and schedules must allow for test procedure review, correction and subsequent approval, occasional re-testing after test failures, and rescheduling for unforeseen events such as delays in hardware shipments, weather-related delays, and the unavailability of key test personnel. The procurement specifications must include specific provisions to address the inevitable test failures, re-testing, and consequences with respect to schedule and cost.

7.6 Summary

This chapter has considered the testing requirements for TMS software from design reviews through hardware/software integration testing. It has offered some guidance as to what types of testing

[35] This is delivery to an intermediate shipping address for a limited number of products that would normally be sent to the installation site. Note: where this is done, there must be a methodology for conducting receiving inspections and limited functional testing at the intermediate address (i.e., the development facility) for these items to be accepted. Otherwise, conditional acceptance (for partial payment purposes) may be allowed at the intermediate location with final acceptance at the installation site.

should be considered and when, who should develop test procedures, and testing in both the development and operational environments. The need for maintaining a software development environment even after final system acceptance was also stressed.

As with the hardware test program, the software test program is also dependent on the procurement specifications. The procurement specifications must establish the requirements for the contract deliverables and the testing program, specify the consequence of test failure, and identify the schedule and cost impacts to the project.

The majority of the discussion has assumed that the TMS software is new or custom. There are now a number of “standard” ATMS software packages available from a variety of vendors that are likely to be able to provide most if not all of the functionality necessary of a robust TMS. Such software typically includes DMS control (including travel time calculations), traffic monitoring, arterial control, CCTV control, HAR control, incident tracking and management, web interfaces, and center-to-center capabilities. Agencies are encouraged to review the products generally available to determine if and how they might meet their needs.

The selection of such “standard” products does not eliminate the need for extensive software testing, regardless of the track record for the product. The agency needs to work with the supplier to ensure that the system functionality is well defined (requirements) and that the system can be tested to show how it meets those requirements. Examples include data collection accuracy, data errors, map displays, algorithm accuracy, and screen performance, to name a few. It is also important that an acceptance test procedure be developed, possibly by the vendor, that will serve as the basis for acceptance of the system. Again, this is the agency’s opportunity to verify the full and complete operation of the system and to verify that it can handle the full load and expansion requirements; such a test should include the maximum number of workstations, intersections, CCTV devices, DMS, etc. It is likely that simulators will be required to support this type of extensive testing.

One final comment: most of today’s underlying COTS products such as Windows, Oracle, and others have software flaws; bugs, if you will. Some of these may adversely affect the stability of the TMS applications software. Hence, it is important that system upgrades be handled in a cautions manner as the TMS software will have achieved a track record, most likely on older versions of the COTS products. The rate of update needs to be controlled by the developer to ensure that a previously stable product does not become unstable due to some unforeseen consequence of an operating system upgrade. Controlling such updates is one of the responsibilities of the CCB discussed earlier. The task of the CCB is to assess the likely impact on the operational system when COTS updates are suggested. The CCB can then examine the pros and cons of the upgrade and develop a cautions procedure – perhaps starting with the “test” environment first.

8. System-Level Testing

8.1 Overview

This chapter discusses system-level testing on the installed, operational (production) system. Testing at this level is typically conducted to verify both completed subsystems and the system as a whole from the software developer, vendor(s), and systems integration contractor under the terms of the contract.

Before attempting system-level testing, all unit, installation, and hardware/software integration testing should be complete. Any problems identified at these levels should have been corrected and re-tested. It is also possible that the agency has decided to accept certain changes; under these circumstances, it is important that the system requirements be changed and documented and that the revised requirements serve as the basis for the final systems testing.

If the TMS is being incrementally deployed (see section 3.3.2), then the system-level test planning and procedures must be developed such that each increment can be separately tested and accepted. Under these circumstances significant regression testing may also be required to ensure that the incremental functionality or geographical extension does not compromise the operation of the system.

8.2 Subsystem Testing

Subsystem verification testing is performed as a prelude to system testing. It is performed in the operational environment using installed system hardware and software. Testing at the subsystem level should be performed:

(a) When different developers, vendors, or contractors have been responsible for delivering stand-alone subsystems.
(b) When the complete functionality of a subsystem could not be tested at a lower level because it had not been fully integrated with the necessary communication infrastructure.
(c) When it was previously impossible to connect to the field devices for the testing phase.

Testing at the subsystem level has distinct benefits over delaying that testing to the higher level system testing:

- The test procedures and test personnel can concentrate on a limited set of system requirements and functionality.
- Problems encountered during the test can be resolved independent of other testing.
- Testing can be completed in a shorter time span and with fewer resources and disruption to other operations.
- Acceptance can be incrementally achieved and vendors paid for completed work.

Note that conditional acceptance for subsystems that have lengthy burn-in periods or specific operational performance requirements may be granted by the acquiring agency in order to allow progress or partial payments to be made prior to final acceptance.

8.3 Systems Testing

System verification testing is the highest test level; it is also usually the one with the fewest requirements remaining to be verified. Only those requirements relating to subsystem interactions, quantity of field devices, external interfaces, and system performance should remain to be formally verified. System acceptance testing is performed after all lower level testing has been successfully completed. It is performed in the operational environment using all available and previously installed and tested system hardware and software.

The system acceptance test should include an end-to-end or operational readiness test of sufficient duration to verify all operational aspects and functionality under actual operating conditions. While this may not be possible in a reasonable period of time,[36] the system test plan and test procedures should specify which requirements must be tested and which are optional given that those operational circumstances occur during the test period. The test should be conducted using trained agency or contractor staff that will manage, operate, and maintain the system following acceptance. One aspect of the readiness test should be to assess the sufficiency of staff training and operations manuals such that any deficiencies can be corrected before the agency assumes full ownership and all operational and maintenance responsibilities. This latter point is important for the agency to consider; all too often, the final system documentation such as user manuals are pushed until the end of the project. Because everyone is anxious to start using the system, preliminary documentation or minimal documentation is provided and the agency moves forward with the testing because “things” seem to work properly. It is recommended that the agency resist this temptation and evaluate the documentation as part of the final system acceptance test. If the contractor is aware of this situation in advance, it is more likely that they will complete the documentation for review much sooner.

Note that it may be necessary (due to contractual relationships) to move into overall system testing even though certain elements of the system may not be in place or operational. Examples may include outlying ITS devices or interfaces to other systems that are not yet available. Under these circumstances, efforts should be made to include simulators or demonstration devices to allow the full system testing to move forward. This approach brings some risk because the simulators or demonstration devices (and their interfaces) may differ from the final implementation, but it is often the only way that a project phase can be “closed out.” Under these circumstances, it should be recognized that the introduction of such simulators and demonstration devices may increase the cost of the contractor’s activities. Further, the agency has a responsibility to test and evaluate the simulators to ensure that they are representative of the actual devices.

[36] This limitation is likely to be because during the planned test period a specific set of circumstances (blizzard) may be unlikely to occur.

8.4 Summary

This chapter has provided a brief discussion of what system-level testing is and what it should accomplish. It represents the final step in the TMS acquisition process.

It is very important that the procurement document clearly and unambiguously describe what constitutes final acceptance. Procurement documents frequently require the conduct of an acceptance test followed by an observation period. The contractor's expectation is that title to the system transfers at the completion of the test, but the agency intended the transfer to occur after completion of the observation period. The transfer can occur any way the agency wants, but it should be uniformly understood by all parties. Formal acceptance at the subsystem or system level may also trigger the start of equipment warranty periods, software licensing agreements, operations and maintenance agreements, etc. The procurement documents should clearly specify which of these are applicable, when they become effective and need to be renewed, and what the payment schedules and provisions are.

9. Other Testing Considerations

9.1 Overview

This chapter provides some other helpful and practical testing considerations that did not fit elsewhere into the basic structure of this document. They have been include here because they are important and provide specific guidance and recommendations that should be useful in your test program.

9.2 Meaning of Shall, Will, May, Should and Must in Requirements Statements

The language, syntax, and structure of you requirements statements are extremely important as they directly affect the quality and thoroughness of the test program that is based on them. Certain terms used in requirements statements have specific contractual implications.[37]

"Shall" is used to confer a requirement on the provider of the product or service and is typically understood to mean at the time of delivery.

"Will" is used to confer a requirement on the receiver (the accepting agency) of the product or service when that product or service is delivered. "Will" is also used to imply a *future* requirement on the provider that should be clear from the context of the statement.

"May" is a conditional term and implies that either the provider or the receiver has the *option* to meet the stated requirement. "May" statements are generally not testable unless additional conditions are included to indicate what is expected if the provider or receiver elects that option.

"Should" falls into same category as "may" and is considered an optional requirement that may or may not be included in the system depending on the provider's perspective.

"Must" is used to add additional emphasis to the requirement statement that can be directed at either the provider or receiver, but more typically the provider. "Must" is typically used in a requirement that has specific legal or contractual ramifications such as may be invoked by requiring a particular State statute or governing regulation be strictly adhered to in the performance of the work. In the contract specifications, it has the same basic meaning as "shall."

From a contracting perspective, only requirements with MUST and SHALL statements are likely to be provided by the contractor. All other requirements should be considered part of a "wish list" and will not be part of the testing program.

[37] The following terms are defined in MIL-STD-490A Specification Practices, Section 3.2.3.6 Use of "shall," "will," "should" and "may."

9.3 How To Write Testable Requirements – Do's and Don'ts

The requirements statements contained within the procurement specifications are the basis for test and acceptance. Poor or missing requirements statements result in requirements that cannot be verified and products or services that don't meet expectations. Requirements statements should be written as clear, unambiguous, declarative sentences. Proper grammatical sentence structure is as important as is use of "shall" and "must," particularly in defining who is the responsible party for providing the product or service and who will be accepting delivery of that product or service. The following are some do's and don'ts for writing and reviewing testable requirements.

Do's

Write the requirement in simple, understandable, concise terms; be short and to the point. If complex technical terminology is necessary, make sure those terms are defined or well understood by the provider as well as the receiver.

For each [individual] requirement there should be one *shall* or *must* statement. If the requirements are complex, then they should be subdivided into a string of individual requirements to the greatest extent possible. A test case will be generated to verify each "shall."

Write the requirement as a positive statement. If something is not desired, try to phrase the requirement to state what is desired. However, this is not an absolute; if the system is not supposed to allow expired passwords to be used, then an explicit "shall" statement with that requirement should be included. For example, "The system shall reject and log each user logon attempt that uses an expired password."

Have a test method, such as inspection, certificate of compliance, analysis, demonstration or test, and pass/fail criteria in mind when writing or reviewing the requirement. If you can't figure out how to verify the requirement or what criteria constitutes acceptance, you can't expect the provider to demonstrate compliance with the requirement. This approach may cause the writer to re-write the requirements with testing in mind.

Consider the complexity, technical expertise, and expense of the testing that may be necessary to verify the requirement —simplifying the requirement may result in the same end product or service, but at a reduced test expense.

When preparing the requirements statement, be careful what *frame of reference* is used for the reader. As noted earlier, software developers and traffic engineers have entirely different frames of reference. What may seem clear to the traffic engineer may become mangled when interpreted by a software developer! As the requirements are prepared, make sure that the requirements will have the same interpretation regardless of the background and experience of the reader. Add clarifying information when necessary to ensure a common understanding by readers with radically different backgrounds.

Don’ts

Avoid the use of “may” and “should” in the requirement statement unless you specifically want to give the provider an option in how that requirement can be met, or give the receiver an acceptance option or an “out.”

Avoid negative requirements. For example the statement “tightening torque shall not exceed forty foot-pounds” implies anything less than forty foot-pounds would be acceptable, but if the requirement applies to the torque applied to tightening a fastener, a positive statement such as “shall be tightened to 35 foot-pounds +/- 4 foot-pounds” is much better, because it defines the minimum as well as maximum torque to be applied and can be definitively measured for acceptance testing.

Don’t mix dissimilar or unrelated requirements in the same statement. This practice complicates requirements traceability and verification testing. Unrelated requirements will usually be verified at different times, under different test conditions, and using different test procedures or methods.

9.4 Test Pass/Fail Criteria

Test procedures should detail each test step. They must indicate what requirement (or partial requirement) is being verified; what action, event, or condition must occur to execute or complete the test step; and what is the expected outcome or response to that action. The expected result (outcome or response) is the pass/fail criteria for that step. If the test step was executed and the expected result did occur and was either witnessed or recorded and can be confirmed, then that test step can be considered to have passed, and the test step’s requirement is verified. If the test step could not be executed or was executed and the expected result did not occur, was not witnessed, or cannot be confirmed from the record, then that test step must be considered to have failed and that requirement not verified. Any outcome other than the expected one should be considered an anomaly or error (i.e., failed).

Beware of transient conditions. Testing is an important aspect of system acceptance and **everything** that happens during the test counts. Hence, all test participants must be focused on the testing operation. With today’s complex systems, it is not unusual for “strange” things to happen that are not repeatable. For example, as the operator is performing a test, a specific screen appears to show a non-existent error which does not re-appear when the screen is refreshed. Was this an error or anomaly? At this point, the system might be considered suspect and the tester may want to repeat the step (to confirm the final condition). Be sure to log this type of event and file a report. Although it may not re-appear during the testing, it may provide a clue to some other unrelated problem experienced later. At the very least, the vendor should be required to explain what and how it could have occurred.

9.5 Test Reporting

Test reporting requires an accurate log of the test configuration, test conditions, the requirements that were verified, specification of the pass/fail criteria, and identification the completed test steps. Good test procedures establish what was intended and provide a checklist for tracking the progress of the testing. The test report should summarize the test activities, including test date, time, and location, test witnesses and observers present, exceptions or anomalies noted, and SPCRs written. The test report should include the original copy of the procedure checklist, with test witness-initialed steps, data colleted, supporting analyses, and a test completion status and/or re-test recommendation. The test report is usually prepared by the test conductor and submitted to the test director. The accepting agency determines final test completion status from review of the test report.

One approach that may be useful is to construct a large 3-ring binder with the complete test procedure. Then, as each test step is taken that requires inspection, calibration certificates, print-outs, pictures, etc., this data can be added to the book and provide a complete record of what was done and by whom. If one is performing hardware testing, it is advisable to take pictures of the test configuration, test actions, scope traces, and the environment. Such additional information can be invaluable when preparing the final test report and provides further proof of the activities and actions. There are techniques such as using “alt-PrtScn” and “ctrl-PrtScn” to capture screen shots (the active window or the whole screen) that can be used to provide snapshots of the user interaction with the system.

It is important that the agency maintain control of the test data collection document or test “book.” The agency must be diligent in recording the results. The perspective, while unpleasant to consider, must be to keep records that the agency could use in a court of law to prove or show contractor non-compliance – i.e., test failure. Under worst case scenarios, the agency may be called on to show cause as to why and how the test results show that the contractor did not complete the work as contracted. These test records may be the only record of what happened since both the agency and contractor personnel witnessed the tests and initialed the logs.

9.6 Test Failures and Re-Testing

Tests fail for a variety of reasons, many of which have nothing to do with whether or not the requirements being verified by the test have been met. Examples of test problems that may be encountered resulting in a test failure include:

(a) Poor or inadequate test procedures.
(b) Incorrect test procedure execution (e.g., skipping a test step, executing test steps out of sequence, failure to check off completed test steps, and failure to record test data).
(c) Inappropriate test conditions (e.g., lack of test support personnel, insufficient traffic volume to trigger congestion detection algorithm response, poor visibility due to deteriorating weather conditions, etc.).
(d) Device failures, including the communications infrastructure.
(e) Failure of test equipment.

Many of these situations can be avoided by thoroughly reviewing all test procedures, executing a dry run of the test before the formal test in front of the customer (agency), providing additional on-call test and maintenance support personnel, checking expected test and weather conditions before starting the test, and ensuring the availability of backup test equipment.

Even with these pre-test precautions, however, things happen and tests fail. The procurement specification should allow for re-testing both for cause (i.e., the test failed to verify the requirement due to product design, implementation, or test inadequacy) and for reasons such as those listed above. Where possible and at the direction of the test conductor, the procurement specification should allow the test to be re-started. Examples include restarting from a point before the skipped step, or steps executed out of sequence, before the failure of a supporting device – not under test, or failure of the test equipment, etc. Alternatively, the test may be repeated from the start (either immediately or within an hour or so), provided the problem is recognized before the test is terminated by the test conductor and if test conditions can be re-set (i.e., error conditions cleared, processes or devices re-initialized, equipment replaced or repaired within a *reasonable* period of time) and the other necessary test conditions can still be met. Testing is expensive, resources and schedules are finite, and thus it is to everyone's advantage to complete a test that would otherwise result in a failure if a minor procedural error or test condition can be resolved quickly allowing the testing to proceed to a successful conclusion. The procurement specification should also allow minor test irregularities to be waived or partial test to be executed to "clean up" an irregularity. The procurement specification must also state very clearly the agency's process for resolving test conflicts or disputed results.

There may be conflicting interests once a test step has clearly failed due to equipment or software malfunction. If such a failure is discovered during day 2 of a planned 5-day test, does one press on and complete the entire test (where possible) to see if there are other problems? Or does one halt the test until the repair/correction can be made and then restart the testing? If the repair requires a week to complete, what is the proper course of action? The vendor might argue that to continue simply expends time and resources better spent on correcting the problem and preparing for a re-test, while the agency may want to press on to see if there are other issues or problems with the system. The specifications should place this decision with the agency and require that once the test has started it is the judgment of the agency as to whether the test continues after such a failure or is terminated and re-scheduled. Note that in some instances, continued testing may be impossible due to such occurrences as a corrupted database, failure of a server, or failure of some other mission-critical device.

Another issue that must be considered is "how many times" the vendor is allowed to fail the testing before terminating the contract or forcing some other drastic action. While such conditions are not anticipated or expected to occur, project specifications need to address possible scenarios and place limits on the number of re-tests allowed and the timing of such re-testing. Then are also issues of who bears the cost of such re-testing as well as how many re-tests are allowed. While it should be clear from the contract that the contractor is responsible for the test environment and all costs associated with performing the tests (including laboratory personnel, test equipment, consumables, utilities, space, etc.), there may be a cost to the agency for consultant services to observe and monitor the testing as well as possible travel costs if the testing is performed at another facility some distance from the agency's offices. Such issues need to be addressed in the contract. Examples include a requirement that the contractor prepay all travel expenses (using government per diem allowances) for all testing (which tends to place distant vendors at a financial disadvantage), or a limit of 3 "free" testing attempts. In some instances, the contractor may be required to reimburse the agency for the expense of its consultants for retesting after a limited number of "free" attempts. How this is dealt with will depend on the number of units involved, the contract provisions, and the agency's procurement policies and procedures.

9.7 Testing Timeframes

Defining the timeframes for testing is a critical function of the procurement specification. Be prepared to address test interruptions by planning for them early and defining their handling.

A test's timeframe should be defined in the procurement specification. Usually this timeframe is set in terms of *contiguous* calendar days. This terminology is important to set a maximum time for conducting the tests and to avoid terminology that allows contractors to interrupt or suspend tests in order to make corrections and then resume the tests.

When defining operational tests of longer durations (30-90 days), the procurement specification must be realistic about the probability that external forces will impact system operations. For example, outages with leased communications facilities will impact overall system operation but are beyond the control of the contractor. Also, field facilities may be damaged by vehicles causing knock-downs of equipment. It is not realistic to hold a contractor liable for events beyond their control. There will be small maintenance issues that occur, but these need to be put into perspective and dealt with without impacting the operational test program. For example the operational test should not be failed due to a report not working because the printer is out of paper, but the test should be concerned about a loss of communications due to the failure of a power supply in the custom central communications equipment.

One also needs to be realistic in understanding that equipment does fail and that during a 30 or 60-day “observation” period, it is possible that of 250 field devices, one or more may experience a failure or anomaly. Restarting such an observation period at day one for each such failure will almost guarantee that the test will never be completed. While agencies may see this as an extension of their warranty, such expectations are unrealistic. Instead, the agency should establish a failure management and response approach that recognizes this possibility and establishes criteria for determining that the test has failed and must be re-started vs. continued without suspension, or suspended and continued once the failure has been corrected. Factors such as the severity of the failure and the time to repair should be incorporated into the decision. For example, if during the period of observation, a DMS panel experiences a failure, the contractor might be allowed 48 hours to correct the problem without affecting the test; however, if more than 3 signs (of 25) experience such a failure within the 60 day test, the test may be considered to have failed and must be restarted. The decision should be based on whether there appears to be a symptomatic problem or a random failure of the device. For system software this may become more problematic. For example, if there is a “memory leak” that seems to be causing the system to crash and need to be re-booted about once per week, does one continue to move on, suspend, or terminate? If the problem can be quickly diagnosed and repaired, a restart is probably in order, but if the problem appears half way into the test, what is the best approach? Should this be noted and corrected under the system warranty? Or, should the test be halted, the problem corrected, and the test restarted?

There are no easy answers to these issues; the agency needs to ensure that their system and devices are reliable, while the project needs to move on to final acceptance so that the contract can be closed out. Be prepared to deal with these issues and be sure that they are addressed in the project specifications.

9.8 Testing Organization Independence from Developers

Testing should be developed and conducted by an organization that is independent of the product development organization. For test procedures, this helps to ensure that the test procedures verify the requirements as stated, not what was assumed to be wanted or needed by the developer, or, for that matter, intended by the acquiring agency. If the requirement is unclear, vague, or ambiguous, the test organization will not be able to develop a test procedure to verify it and will ask that the requirement be revised or rewritten such that it can be verified. This needs to happen early during product design and development, not after the product has been delivered for acceptance testing. Therefore, the test organization needs to start test planning and test procedure development in the requirements definition and analysis phase of the project.

For test execution, test personnel from an independent organization will not overlook or ignore unexpected test results that a developer might. Often the developer will overlook anomalies because he can explain them or knows that those unexpected results are not related to what is being tested (i.e., they could be caused by an unforeseen interaction with another process or device not directly related to the test). If there are problems or unexpected results that occur during the test, they need to be recorded and reported so they can be corrected or resolved before accepting a potentially flawed product.

While this section recommends an “independent” test organization, it is likely that the contractor will handle the testing from test plan generation to test execution as well. Within most organizations, an independent test group will take on this responsibility and this should be permissible as long as the personnel are independent from the developers and designers. Review the contractor’s organization chart and determine the degree of independence of the testing group.

9.9 Testing Relevancy and Challenges

Some challenges that must be met in a testing program relate to the relevancy of the test with respect to the actual test conditions and test limitations and constraints. For example, if the stated test condition requires a period of high traffic volume, testing at night or during an off-peak period will probably not achieve the desired test result, compromising the ability to verify the requirement which depended on the existence of that condition for demonstration. Make sure the expected test conditions are relevant for the requirements being verified. For the example cited, one may need to develop calibrated simulators that are installed in the remote cabinets to actually demonstrate that the specific requirements have been met.

Test limitations and constraints must also be considered to ensure that the test is relevant and the test results will demonstrate compliance to the requirements being tested. For example, if the test is limited to the CCTV camera subsystem, it should not have any test steps that verify requirements for the DMS subsystem. However if camera selection for control is accomplished by clicking a mouse pointer on the camera’s icon on the GIS map display, requirements for that control action and related GIS display are relevant and should be also verified in the CCTV camera subsystem test. Further, where the GIS display is active, it may be prudent to ensure that all map “layers,” which would include the DMS, be shown.

A typical test constraint might limit DMS test messages to a pre-defined fixed set, even though a much larger set of both pre-defined and user-generated messages will ultimately be required and verified in a different test. In this example, the test is limited to a pre-defined set, so the DMS software needed to support the test does not have to be the final version. More precisely, where the final version would typically allow the user to both edit existing messages and create new ones, the test software would only allow the selection of pre-coded messages. Here, the test relevancy has been purposely limited to verifying the ability of the DMS subsystem to access stored messages and display them. This test limitation allows early verification of a critical portion of the DMS requirements while design and development of software to satisfy the complete set of requirements continues.

Such a situation might be useful for conducting a 30 or 60 day test message burn-in where later tests will fully verify the central system capabilities.

9.10 Issues Affecting System Reliability

When establishing a burn-in or extended system test, it is important to consider what might happen, how systems fail, and what steps the designers may wish to consider to mitigate the effects of such failures. Criteria for acceptable system performance and the calculations for system reliability are also discussed, again, as background when considering how to manage extended testing.

9.10.1 System Failure Modes and Effects

TMS subsystem failures can result from a large number of different causes, and a particular failure event can have a variety of effects. This section examines some of the more common failure events and the design and operational factors that can mitigate the effects of those failure events. It also examines failures of typical critical communication infrastructure components and addresses their failure modes and effects.

Table 9-1 presents some of the more common events that can cause failures and the factors that can mitigate their occurrence and/or severity. Note that redundant capabilities are listed as mitigating factors for cable plant damage and power outage events only. While it could be argued that some form of redundancy could mitigate the effects of all of these causal events, it would be true only when that redundant capability is geographically separated or provided by different means or methods other than the primary capability. That is, the causal event would not affect both the primary and redundant capability in the same way at the same time. Since this is typically not the case, the mitigating factors become very important and must be considered as well as possible redundancy options when developing the project specifications and requirements.

Table 9-1. Common Failure Causes and Mitigating Factors

Causal Event	Mitigation Factors
Lightning	Lighting Arrestor Attenuators/Filters Non-Conducting Materials Proper Bonding and Grounding/Isolation
Fire	Material Selection Elimination of Ignition Sources Fire Suppressant/Extinguisher
Flood	Site Prep/Drainage Equipment Location Enclosures/Seals Alarms
Wind	Structure/Support/Strain Relief Mounting Enclosure

Causal Event	Mitigation Factors
Temperature	• Component Selection • Ventilation • Insulation • Mounting (Expansion/Compression) • Heating Ventilation and Air Conditioning
Humidity	• Component Selection • Coatings • Enclosures/Seals • Heating Ventilation and Air Conditioning
Shock and Vibration	• Component Selection • Mounting and Isolation
Vandalism	• Access Controls • Surveillance
Animals/Insects	• Site Prep/Clear Vegetation • Cover/Close Access Ports • Screen Vents • Remove Debris and Refuse • Regular Inspections
Power Outage	• Notification/Coordination of Activities • Utility Locates • Redundant/Secondary Feed (Long Term) • On Site Generator (Days to Weeks) • Uninterruptible Power System (Short Term)
Cable Plant Damage	• Utility Locates • Redundant Cable • Notification/Coordination of Activities
Improper or Incorrect Maintenance	• Staffing/Training • Management Oversight • Diagnostic Tools • Logistics (Spares/Provisioning and Deployment) • Preventive Maintenance/Inspections • Upgrades/Process Improvement • Communication/Coordination of Activities
Improper or Incorrect Operation	• Management Oversight • Staffing/Training Communication/Coordination of Activities • Upgrades/Process Improvement

Table 9-2 lists the typical critical components, their network locations, failure modes and effects, and how those failures would be detected and isolated for a typical TMS fiber optic communications infrastructure. As shown in the table, most failures would be automatically detected by the network management system and isolated to the component level by either the network management system or maintenance staff and in some cases with the assistance of the network administrator and software staff. An optical time domain reflectometer (OTDR) is the primary tool used by the maintenance staff to locate a problem with the fiber optic media and verify its repair. The mean time to restore (MTTR) includes the time to detect and isolate the failure as well as test the repair needed to restore full functionality. MTTR is estimated based on having spare critical components strategically pre-positioned and a well-trained technical staff skilled in the use of the network management system, OTDR, and other necessary tools.

Table 9-2. Failure Modes and Effects

Critical Component	Failure or Fault	Location	Effect	Detection	Isolation	Mean time to Restore
Fiber Distribution Center	Fiber Optic Pigtail/ Fiber Optic Connector	Network Node	Link Loss/ Multiple Link Loss	NMS	Maintenance Staff	< 4 Hrs.
Fiber Optic Jumper	Fiber Cable/ Fiber Optic Connector	Network Node	Link Loss	NMS	Maintenance Staff	< 1 Hr.
Splice	Single Fiber/ Multiple Fibers	Field Splice Box	Loss of 2-way Comm/ Multiple Link Loss	NMS	OTDR	4 Hrs to 2 Days
Fiber Backbone	Elongation/ Bend Radius/ Abrasion/ Partial Cut/ Sever	Turnpike Mainline	Performance. Loss/ Multiple Link Loss/ Dn. Stream Comm Failure	NMS	OTDR	1 to 4 Days
Fiber Drop	Partial Cut/ Sever	TMC or Equipment Site	Loss of 2-way Comm/ Comm Failure	NMS	OTDR	1 to 2 Days
Network Repeater	Input Port/ Output Port/ Power Supply/ CPU	Network Hub Site	Node and/or Dn. Stream Comm Failure	NMS	NMS/ Maintenance Staff	< 4 Hrs.

Critical Component	Failure or Fault	Location	Effect	Detection	Isolation	Mean time to Restore
Network Switch/ Router	Input Port/ Output Port/ Power Supply/ CPU/ Routing Table/ Software	TMC		NMS	NMS/ Maintenance Staff/ Network Administrator	2 to 4 Hrs.
Hub	Input Port/ Output Port/ Power Supply/ CPU	TMC or Equipment Site	Link or Multiple Link Loss	NMS	NMS/ Maintenance Staff	< 2 Hrs.
Network Management Host	Network Interface Card/ Power Supply/ CPU/ Operating System/ NMS Software	TMC	Loss of Comm. Subsystem Health and Status Data and Reconfig. Capability	Network Admin.	Network Administrator/ Maintenance Staff/ Software Staff	< 1 Hr. (Switch-over to Hot Standby)
Network Server	Network Interface Card/ Power Supply CPU/ Operating System/ Application Software	TMC or Equipment Site	Loss of System/ local Functionality	NMS	Network Administrator/ Maintenance Staff/ Software Staff	< 4 Hrs.

NMS = Network Management System

An examination of table 9-2 suggests that meeting a high (e.g., 99 percent) availability goal would be difficult to achieve with MTTRs exceeding the 8 hours that would be allowed in a 30 day period to meet a 99 percent availability goal for some failure events, unless these events have a low probability of occurrence. One way to mitigate these failures is to provide redundancy for the critical components with potentially high MTTRs. For example, if a backup TMC is implemented that will provide a hot standby capability for the primary TMC network management host and network servers, the estimated MTTR can be much less than one hour (maybe seconds) assuming a switchover to the backup TMC occurs following a failure at the primary TMC. Without this capability, it could take a day or more to restore full functionality at the TMC even if the necessary spares were available. Note that once a switchover to a redundant capability at the backup TMC is accomplished, a subsequent failure of that capability would result in an outage that could take a full day to recover, unless there are multiple levels of redundancy or the primary TMC repair occurs before failure at the backup TMC.

Since the fiber backbone, fiber drops, and associated splices have high estimated MTTRs, it would be prudent to implement some type of redundancy to mitigate the impact of a failure event for these elements as well.

If backup and redundant elements are part of the overall project requirements and specifications, it is important that the testing program, at all levels, verify the switch-over times, the recovery process, and the system's ability to detect and alert the operators to the failure(s). Such testing should include multiple and compound failures of all systems and components. This type of testing should be representative of the failures that will occur; i.e., simply using the computer console to "halt" a process is not the same is shutting the power down and observing the result.

9.10.2 System Reliability, Availability and Redundancy

The intent of a system availability requirement is to set a standard for acceptable performance for the system as a whole to avoid installing a system that does not meet operational needs or, worse, is not reliable (as defined in the requirements). Requiring a system to meet a specific performance standard with respect to reliability and availability at the outset typically comes with a very high initial cost. This is primarily due to over design and over specification coupled with the attendant analysis and testing needed to verify that a specific performance standard has been met. Because reliability and availability are related, setting a *goal* (rather than a hard requirement) for system availability may allow both to be achieved over time through a process of continuous improvement and can result in a significantly lower overall cost. For this approach to work, however, it is essential that system operational performance and failure data be collected to determine whether the availability goal is being met and thus whether and where improvements are necessary.

Defining an acceptable level of availability for a large, complex system can be a daunting task. There are two key aspects to this task:

- Identifying those functions (hence components and processes) that are deemed critical to system operation and the stated system mission. It is assumed that the loss or interruption of these functions for longer than some pre-determined time interval is defined to be system failure.

- Determining the duration of operation without failures (i.e., failure-free operation).

Ideally, one would like to have a very long period of failure free operation, particularly for the critical functions. The reality is that the system components or a software process will fail or that some aspect of the system's performance will eventually degrade below an acceptable level. All that one can reasonably expect is that the failure is quickly detected and diagnosed, and the repair or replacement of the failing item is completed as soon as possible, thus restoring normal operation.

If one cannot tolerate the loss or interruption of a critical function (even for a short interval), some form of active redundancy is required. That is, some alternate means of accomplishing the critical function must be instantly available. Several levels of redundancy might be required to reduce the probability of a loss or interruption to near zero. If a failure can be tolerated for a short period of time, then there is the possibility that the critical function can be restored within that time interval, either by switching to a standby redundant capability or by repairing or replacing the component or process causing the loss of functionality. The longer the outage can be tolerated, the greater the likelihood that the critical function can be restored without relying on redundancy. Hot standby redundancy is always an expensive solution and is usually not necessary or required unless the critical function has a life safety aspect or is considered to have other mission critical real-time dependencies.

In order to set a system availability goal that is both meaningful and reasonable for the TMS, it is necessary to define some terms and discuss some mathematical relationships.

Availability (A) is the probability that an item will operate when needed. Mathematically, it is defined at the ratio of the failure free service interval to the total in-service interval typically expressed as:

$$A = MTBF/(MTBF+MTTR)$$

Where:

Mean Time Between Failures (MTBF) is the average expected time between failures of an item, assuming the item goes through repeated periods of failure and repair. MTBF applies when the item is in its steady-state, random-failure life stage (i.e., after the infant mortality and before the wear-out period), and is equal to the reciprocal of the corresponding constant failure rate, the *Mean Time To Failure* (MTTF).

Mean-Time-To -Restore (MTTR) is the average expected time to restore a product after a failure. It represents the period that the item is out of service because of the failure and is measured from the time that the failure occurs until the time the item is restored to full operation. MTTR includes the times for failure detection, fault isolation, the actual repair (or replacement), and any re-start time needed to restore full operation.

Reliability (R) is the probability that an item will perform a required function under stated conditions for a stated period of time. Mathematically, reliability is typically defined as:

$$R = e^{-T / MTBF}$$

Where:

e is the base of the natural logarithm (2.718….)

T is the time of failure free operation

MTBF is mean time between failures or 1/MTTF.

For example, if an item had a demonstrated MTBF of 2000 hours, what is the probability of achieving 200 hours or failure free operation?

$$R = e^{-200/2000} = 0.905 \text{ or } 90.5\%$$

Thus, there is a 90.5 percent probability that 200 failure free hours or operation could be achieved. Continuing with this example: if the item can be repaired or replaced and returned to service in 4 hours what is the expected availability during the failure free interval?

$$A = 2000/(2000+4) = .998 \text{ or } 99.8\%$$

With a 4-hour restoration time, the item can be expected to be available for service 99.8 percent of the time.

The above examples are very simplistic and only apply to a single item. For large, complex systems, reliability is typically assessed for the critical path, i.e., the series of components and processes when taken together provide critical system functionality. It is computed as the product of the reliabilities of the components/processes on that path. In practice, estimating a system's reliability and availability would be a very difficult task and require an enormous effort even if all of the necessary failure and repair statistics were available, including the appropriate usage assumptions, confidence levels, weighting factors, and a complete understanding of all the failure modes and effects for each of the system’s components. The operating agency can, however, impose a system-level availability goal, define critical functions, and collect operational performance data with respect to failure free operation time and time to restore operations for those critical functions. This information can be used to compute actual system availability for comparison against the proposed goal. The information collected will be useful in determining whether the current operational performance level is acceptable and what needs improvement.

Suppose that a service outage of 12 hours during a 24-hour by 5-day operational period were tolerable, the system availability goal would be:

$$A= (24*5-12/(24*5) = 108/120 = 0.90 \text{ or } 90\%$$

A 90 percent availability goal may not be high enough initially, but this value does allow for a significant failure restoration time and accounts for single as well as multiple failure events. The 12 hours allotted includes the time to detect the failures, dispatch a maintenance technician and/or software engineer, diagnose and isolate the problem, repair or replace the problem component or process, test, and, if necessary, re-start the operation. If the operating agency finds that the 90 percent availability goal results in an unacceptable operational service level, it can be raised to force improvements to be made.

Note that service outages can be caused by both unplanned events and scheduled maintenance and upgrade events. The effects that determine the duration of the outage include whether or not there is any redundancy and the switchover time, as well as failure containment (i.e., minimizing the propagation of a failure once it occurs). Effects that influence recovery time include failure detection and fault isolation times, repair or replacement, and functional test or workaround times. Hardware and software upgrades and plans to minimize service outages through provisioning of spares, critical replacement components, and diagnostic tools are all part of a contingency recovery plan that can be implemented to accomplish availability improvements.

An availability goal forces the operations agency to identify the system's critical functions and collect actual system performance and failure data with respect to those critical functions to determine whether that goal is being met. Without a goal, poor system performance will still be noticed and be unacceptable, but there won't be any hard information as to what needs improvement and by how much. An availability goal also requires the operations agency to address contingency recovery planning which might otherwise be overlooked.

Reliability goals are often used during the “observation” period to determine pass/fail criteria. In this manner, the level of availability can be measured and the determination of when to suspend, terminate, or continue the observation period can be established and measured.

9.11 Testing Myths

The following are two commonly held beliefs concerning testing that in reality, are myths.

9.11.1 Elimination of Software Errors

The belief that software errors, or bugs, can be eliminated by extensively testing the final product is a myth. Well-written software requirements can be verified at least to the functional and operational level. However, one of the unique problems that testing software has is establishing a test environment and developing appropriate test stimuli that are both sufficiently robust and directly comparable to the real-world operational environment. In addition, because of the nearly infinite number of possible paths through the software code that are created by the necessary conditional statements and code modules, testing each possible path takes an infinite amount of time or resources. Only after long operational periods under a variety of conditions and performance stress will most software errors be detected.

Once detected, they can be fixed, or operational procedures changed, to avoid problem conditions. When an error condition or anomalous event occurs or is suspected, a specific test can usually be developed to capture triggering conditions or circumstances and allow the investigation and resolution of the problem. The key is identifying the conditions that allow the test team and the software developer to produce a test case that reliably reproduces the problem. This is true only when the problem is repeatable or can be reliably triggered by some stimulus.

Such tests or test requirements are rarely developed beforehand since the anomalous behavior is not contemplated or expected. If it were, the design should have included a means for avoiding or detecting the problem and taking some corrective action. Moreover, if included in the design or operational requirements, an acceptance test (design review, code inspection and walk through, unit test, build test, integration test or system test) would have verified the desired behavior.

When a problem is not repeatable, i.e., it appears to occur randomly under a variety of different conditions or circumstances, it is most often a software problem rather than a hardware problem. Hardware tends to exhibit intermittent failures under similar conditions or stress and circumstances related to the physical environment. Finding and fixing a problem that cannot be readily triggered by a specific set of conditions or stimulus requires a tremendous amount of luck and technical skill. Over time and as other problems are resolved, these seemingly intractable problems sometimes resolve themselves (because they were caused by interactions with other problems), or they become repeatable such that they can be found and fixed, or simply become less bothersome and easier to live with. A software component or operating system upgrade may ultimately fix this class of problems as there are often obscure bugs in the operating system software that only become active under a specific set of circumstances, which may not be repeatable. A word of caution which was also noted earlier: one method often used to track the cause of intermittent software problems includes the use of debugging tools provided by the operating system, COTS products, or compilers. The introduction of these debugging aids can also perturb the inter-process timing relationships so that the problem “disappears” when the debugging aids are present, and re-appears when they are turned off.

A practical solution is to retain competent software development and test personnel throughout the operational life of the system to deal with the inevitable software errors. It is recommended that software maintenance contracts be executed with all of the COTS and software providers. Most software developers will be continuing to test, enhance, and fix bugs as they are discovered. The software maintenance contract provides a mechanism for introducing those changes into existing systems. However, this approach may also have its share of problems and needs to be part of a configuration management program. Upgrades can have both positive and negative results – it is important that upgrades be tested in an isolated environment and that roll-back procedures be in place in case the upgrade is worse than the existing system or is not applicable to a specific platform or operating environment.

9.11.2 Software Version Control

The belief that a software bug found, fixed, and cleared by verification testing will stay fixed is not necessarily true. The problem was resolved and testing verified that it had been fixed; yet it magically appears again. How is this possible? There could be a number of reasons; perhaps the problem was not really fixed or perhaps the problem was simply masked by other “fixes” or features. In some cases, a new release has replaced the previous release and somehow, during the development of the new release, the old problem reappeared. The reason is that the fix was not incorporated in the newer releases – it was effectively lost when the new release was created.

This typically results from a version control problem (or software configuration management lapse). All the elements that were suppose to be in the new release (including the subject fix) were not properly documented and accounted for when the new release was built, and because it was documented improperly, the regression test that should have incorporated the test for the fix did not occur. Hence, the new release was installed and passed the regression testing, but that testing failed to test for the old problem. In subsequent installation testing with the new release, the old problem resurfaced. The subject fix will now have to be incorporated into a new release and re-tested.

Software version control is extremely important, particularly for a large development project that may have multiple release versions in various stages of integration testing at the same time. The agency needs to periodically audit the CM program to ensure that all problems have been noted, and included in subsequent releases.

9.12 Testing Tradeoffs

There are a number of testing tradeoffs that can have a favorable impact to cost, scheduling or resources required to conduct the overall testing program. Three examples are provided here.

9.12.1 Accepting Similar Components

Whether or not to require the full test suite when similar components are added to expand an already accepted and operational system can be a difficult question. The risk lies in how similar the new components are to the ones already accepted (e.g., are the electrical and communication interfaces the same, will the component fit in the available space, etc.). The safest, least risky course of action is to subject each new component to the same level of testing (if not the same test procedures) used for the currently accepted components. If, however, there are some risk-mitigating circumstances, such as the fact that the product(s) in question are from the same vendor, are listed on a QPL, or are in wide use by others in similar environments, then consideration should be given to abbreviating the testing requirements for these components and accepting vendor test results or a certificate of requirements compliance from the vendor for at least the lower level unit and factory acceptance testing in order to reduce the impact on the testing budget.

9.12.2 Using COTS

Commercial-off-the-shelf (COTS) products, if found to meet the necessary requirements, can save a great deal of money in both development and testing costs that would otherwise be associated with a new or custom product. The tradeoff here is in the continuing licensing costs to use the product and the cost of product maintenance (including product upgrades to remain eligible for vendor maintenance) over the product's useful lifetime vs. reduced testing costs as compared to cost to develop, test, and maintain a custom product. COTS products will usually save money in the short run and will allow needed functionality to be achieved more quickly; however, the longer they are used, the more expensive they become. Eventually, COTS may cost more than a custom product over which you would have had complete control to upgrade, modify, and otherwise use as you see fit. In addition, if you choose a COTS product, you will have to tailor your requirements and operations to meet those of the product and accept the fact that some desired features will not be implemented. For some class of products such as operating systems, relational data base management software, computers, servers, routers, etc. the choice is clear: choose a COTS product. You can't afford to develop these nor should you need to.

For the TMS application software, the choice will depend on the available budget, schedule, and the specific operational needs of the system. The agency needs to carefully review the proposed solution and be comfortable with the "adaptations" required to use the product in their environment. Be mindful that the benefits of using a COTS product can be lost when significant customization is contemplated. Some companies have spent more to modify an existing system to meet their needs than a whole new system might have cost. With today's modular software, it may be possible to assemble a system from well-known and tested modules that minimize the new development required.

Another consideration is the ongoing software maintenance where your choice is a COTS TMS application vs. a custom developed application. If your implementation is unique, you can expect that your agency must bear the full cost of all software support, including upgrades when required to replace hardware that has become obsolete. If your implementation relies on a vendor's "standard" software, then the maintenance costs are likely being shared amongst all of the clients using this software. When it comes to testing new releases or changes, each approach has its own challenges. The use of COTS application software generally means that the vendor must simply update their previous test procedures to demonstrate the new features and functions; with custom software, it is likely that the agency will need to develop the revised test procedures. Further, with custom software, it is likely that there will be no track record of previous testing with the new release which will require that the agency be more rigorous in its testing program.

9.12.3 Testing Legacy Components

Legacy components, i.e., those leftover from a previous system incorporated in the new TMS (unless operated and maintained as a stand-alone subsystem) will have to be integrated with the new system hardware and software. If all that you have for these components is operations and maintenance manuals, i.e., the original requirements, design documents and as-built installation drawings are either non-existent or inadequate; you will be faced with having to reverse engineer the components to develop the information you need to successfully incorporate them into you new system. Testing of these components will be impossible unless a requirements baseline can be established and a “black box”[38] approach used. In addition, unless spares and replacement parts are available, maintenance may also be challenging. It may make sense to operate a legacy system as a stand-alone subsystem until it can be functionally replaced by components in the new system. The tradeoff here is that some initial cost, schedule, and resources may be saved by using the legacy components as a stand-alone subsystem, but, for the long-term, legacy components should be replaced by functionality in the new system.

9.13 Estimating Testing Costs

The test location, test complexity, number and types of tests, and the test resources required (including test support personnel, system components involved, and test equipment) impact testing costs. Testing is expensive, and estimating test costs is a difficult and complex task that won’t be attempted here, except as an example for a hardware unit test given below. What is important and is stressed here is that these costs, while a significant portion of the overall TMS acquisition budget, should not dissuade you from conducting a thorough and complete test program that verifies that each of your requirements has been met. You ultimately control testing costs by the number and specificity of your requirements. A small number of requirements with minimum detail will be less costly to verify than a large number of highly detailed requirements. Both sets of requirements may result in similar systems, but the smaller, less complex set takes less time and resources to verify. Be very careful of what you put in the specification requirements—less requirement detail, unless absolutely necessary, does two things: (1) it allows a more flexible approach to design and implementation, and (2) it reduces the cost to verify the requirement. This approach should be balanced with the agency’s expectations since there may be various means by which the requirement *could* be satisfied by a vendor.

Hardware unit testing can be especially expensive and can significantly add to the cost of a small number of devices. Consider the actual cost of the testing; as a minimum, the agency should send at least two representatives to a planned test—typically these include an inspector and a technical expert. Most testing should also include the project manager—which increases the number to three people, one of whom is typically a consultant (technical expert). The NEMA testing will typically require a minimum of 4 days to complete, and product inspection can easily add an additional day unless additional product is available for inspection. Given the above, the cost is typically 120 hours plus preparation and report generation (add an additional 32 hours) with 5 days each for per diem expenses as well as airfare and local transportation. The costs can easily range from $12,000 to $15,000.[39] In addition to these direct agency costs, the vendor will incur the cost for laboratory facilities, vendor personnel to conduct the test, and the preparation of the test procedure. One needs to consider these costs when specifying a “custom” product, as they are real costs to the agency and the vendor’s costs will be reflected in the cost of the product.

[38] Black box testing is based only on requirements and functionality without any knowledge of the internal design or code.

[39] This is the estimated cost for all 3 people (inspector, consultant, and project manger) and assumes a typical per diem of $150 per day, $650 airfare, $70/hour average labor costs (includes consultant hours) and an allowance of $200 for local transportation.

9.14 Summary

The above testing considerations address specific issues that the acquiring agency has control of at the outset of the testing program. Do not neglect these issues; most will have to be dealt with at some point in your testing program. It is better to plan for tem and deal with them early in the project life cycle rather than reacting to them later under pressure.

10. Additional Resources

This chapter provides a list of web sites that are intended as additional testing resource material. They address TMS relevant standards, organizations and associations that provide training in the testing discipline, and organizations involved with the ITS standards setting process. These resources are starting points and may appear in the results of web searches regarding testing, systems and software engineering, or ITS standards.

Federal Highway Administration (FHWA)

ITS Procurement Workshop materials

http://www.fhwa.dot.gov/resourcecenter/teams/operations/procure.cfm

Systems Engineering

http://ops.fhwa.dot.gov/int_its_deployment/sys_eng.htm

ITS Architecture

http://ops.fhwa.dot.gov/its_arch_imp/index.htm

http://ops.fhwa.dot.gov/its_arch_imp/training.htm

ITS Standards

http://www.standards.its.dot.gov/

National Highway Institute (NHI)

http://www.nhi.fhwa.dot.gov

Institute of Transportation Engineers (ITE)

http://www.ite.org

American Association of State Highway and Transportation Officials (AASHTO)

http://www.transportation.org/

Institute of Electrical and Electronics Engineers (IEEE)

http://www.ieee.org/

IEEE Test Technology Technical Council (TCTT)

http://tab.computer.org/tttc/

IEEE Computer Society (CS)

http://www.computer.org

IEEE Software Engineering Body of Knowledge

http://www.swebok.org

National Electrical Manufacturers Association (NEMA)

http://www.nema.org

National Institute of Standards and Technology (NIST)

Software Diagnostics & Conformance Testing

http://www.nist.gov/public_affairs/guide/software_diagnostics.htm

http://www.itl.nist.gov/div897/index.html

The Open Group (open standards)

http://www.opengroup.org/overview/

International Council on Systems Engineering (INCOSE)

http://www.incose.org

Software Engineering Institute

http://www.sei.cmu.edu

Association for Computing Machinery (ACM)

http://www.acm.org/

Association for Software Testing

http://www.associationforsoftwaretesting.org/about.html

Quality.Org Software QA/Engineering/Testing Resources

http://www.quality.org/html/sw-eng.html

American Society for Quality

http://www.asq.org/

Quality Assurance Institute Worldwide (QAI)

Software Testing Certifications and courses

http://www.qaiworldwide.org/qai.html

Computer software testing newsgroup

http://groups.google.com/group/comp.software.testing / News Group

http://www.faqs.org/faqs/software-eng/testing-faq/ FAQ

Instrumentation, Systems, Automation Society

http://www.isa.org/

References

American Association of State Highway and Transportation Officials (AASHTO), Institute of Transportation Engineers (ITE) and National Electrical Manufacturers Association (NEMA), *National Transportation Communications for ITS Protocol, The NTCIP Guide*, NTCIP 9001, Version 3, v03.02b (Washington, DC: October 2002).

American Association of State Highway and Transportation Officials, *Testing and Conformity Assessment User Guide: A User Comment Draft*, NTCIP 9012 (Washington, DC: December 2005).

U.S. Department of Transportation, *Intelligent Transportation Systems: Critical Standards*, (Washington, DC: June 1999).

U.S. Department of Transportation, Federal Highway Administration, *Developing Intelligent Transportation Systems Using the National ITS Architecture: An Executive Edition for Senior Transportation Managers* (Washington, DC: July 1998).

U.S. Department of Transportation, Federal Highway Administration, *The Road to Successful ITS Software Acquisition* (Washington, DC: 1998).

U.S. Department of Transportation, Federal Highway Administration, *The National Architecture for ITS: A Framework for Integrated Transportation into the 21st Century* (Washington, DC: 1997).

U.S. Department of Defense, *MIL-STD-490A Specification Practices,* June 4, 1985.

Edwards and Kelcey, National Cooperative Highway Research Program (NCHRP) Report, *Guide to Contracting ITS Projects* (Baltimore, MD: 2005).

Appendix A. Example Verification Cross Reference Matrix

The following table is an example of a verification cross reference matrix (VCRM) from a system test plan. The requirements listed in the example are for the video subsystem only and include the system specification requirement statement, the paragraph reference number (with "shall" statement number in parenthesis if more than one requirement is included in the reference paragraph), and the verification method(s) (Inspection, Certification of Compliance, Analysis, Demonstration, and Test). These first three columns of the matrix appear in the system specification and are referred to in that document as the requirements verification matrix. The last three columns: test level (1, 2, 3, 4 or 5), test responsibility (who is responsible for verifying the requirement), and test identification number (i.e., which specific test(s)—in this case at system level 4 or 5—will verify the requirement) are added in the system test plan to complete the VCRM.

Table A-1 Verification Cross Reference Matrix

System Requirement	**Para.** **No.**	**Verif.** **Method**	**Test Level**	**Test Resp.**	**Level 4&5 Test ID**
The video function of the ITS shall consist of the subsystem and components required to collect video information from facilities and roadway, and distribute this video to operators and managers at the TMCs and Highway Patrol Dispatch Center.	3.3.3.1(1)	I/C	2	Installation Contractor/ Vendor	
Video functional element shall include a dedicated surveillance closed circuit television (CCTV) network.	3.3.3.1(2)	I/C	2	Installation Contractor/ Vendor	
The CCTV network shall accept and distribute video signals originating outside the dedicated network.	3.3.3.1(3)	D	4	ITS Consultant	2
Components of the CCTV network shall include the roadway deployed cameras, communication, video switching, camera controls, facility security cameras, video recorders and video monitors.	3.3.3.1(4)	I/C	2	Installation Contractor/ Vendor	

Table A-1 Verification Cross Reference Matrix

System Requirement	Para. No.	Verif. Method	Test Level	Test Resp.	Level 4&5 Test ID
The ITS shall include CCTV cameras along the roadways and other areas of interest to provide system operators a visual indication of traffic and incidents on the roadways.	3.3.3.2(1)	I/C D D	2 3 4	Installation Contractor/ Vendor ITS Consultant ITS Consultant	2
Camera shall support full motion video capability.	3.3.3.2(2)	I/C	2	ITS Consultant	
Video from roadway cameras shall be transmitted to the TMC with the primary operational responsibility for the section of the turnpike on which that camera is located.	3.3.3.3(1)	D	4	ITS Consultant	2
Any subset of video images (minimum of 8) transmitted to TMC shall also be available for monitoring at both control centers.	3.3.3.3(2)	I/C D D	2 3 4	Installation Contractor/ Vendor ITS Consultant ITS Consultant	2
A subset (minimum of 8) of video images transmitted to TMC shall be available for monitoring at the Highway Patrol Dispatch Center.	3.3.3.3(3)	I/C D D	2 3 4	Installation Contractor/ Vendor ITS Consultant ITS Consultant	2
Each TMC and the Highway Patrol Dispatch Center shall have individual switching control of these video images.	3.3.3.3(4)	D D	3 4	ITS Consultant ITS Consultant	2

Table A-1 Verification Cross Reference Matrix

System Requirement	Para. No.	Verif. Method	Test Level	Test Resp.	Level 4&5 Test ID
Access controls shall be implemented to deny unauthorized access to video switching functions identified herein.	3.3.3.3(5)	A/T	3	ITS Consultant	
			4	ITS Consultant	2
The ITS shall support the installed camera base for the TMCs.	3.3.3.4(1)	A	1	ITS Consultant	
Capability shall be designed into the video function element to support a minimum of 20% growth in the installed camera base.	3.3.3.4(2)	I/C	2	Installation Contractor/ Vendor	
Each console workstation shall have a video monitor.	3.3.3.5(1)	I	2	Installation Contractor/ Vendor	
The workstation shall provide the operator with the capability to monitor or display a single video source or multiple (minimum of 4) video sources.	3.3.3.5(2)	I/C	2	Installation Contractor/ Vendor	
		D	3	ITS Consultant	
		D	4	ITS Consultant	2
The Operator Workstation shall include all necessary controls for the operator to select the video source(s) to be viewed and define the position on the video monitor for each selected source to be displayed.	3.3.3.5(3)	I/C	2	Installation Contractor/ Vendor	
		D	3	ITS Consultant	
		D	4	ITS Consultant	2
A capability to select and display blank screen (null video) source(s) shall be provided.	3.3.3.5(4)	D	3	ITS Consultant	
		D	4	ITS Consultant	2

Table A-1 Verification Cross Reference Matrix

System Requirement	Para. No.	Verif. Method	Test Level	Test Resp.	Level 4&5 Test ID
The operator shall also have the capability to select and control the pan-tilt-zoom (PTZ), iris and focus functions of any camera available to the operator for control.	3.3.3.5(5)	D D	3 4	ITS Consultant ITS Consultant	2
Selection of a camera for PTZ control shall automatically select that camera input for display on the operator workstation monitor.	3.3.3.5(6)	D D	3 4	ITS Consultant ITS Consultant	2
The controlling jurisdiction for the camera shall have PTZ control for each camera, but the ITS shall make provisions to allow remote PTZ control only with the approval and authorization of that controlling jurisdiction.	3.3.3.5(7)	A/D D	3 4	ITS Consultant ITS Consultant	2
Relinquishing control shall be through a method that prevents accidental passing of camera control.	3.3.3.5(8)	A/T	3 4	ITS Consultant ITS Consultant	2
Operational protocols shall limit control for camera to one operator at a time.	3.3.3.5(9)	A/T D	3 4	ITS Consultant ITS Consultant	2
Other monitors within a management or control center shall be able to simultaneously display operator workstation video.	3.3.3.5(10)	D	4	ITS Consultant	2
The ITS shall provide for additional video monitors to support crisis management, day-to-day operation and group viewing.	3.3.3.6(1)	I	2	Installation Contractor/ Vendor	
This monitor only capability shall include standard monitors and large screen displays and may include integrated computer screen monitors.	3.3.3.6(2)	I/C	2	Installation Contractor/ Vendor	
Video inputs and outputs shall allow the ITS video subsystem to connect to external video interfaces.	3.3.3.7(1)	D	4	ITS Consultant	2

Table A-1 Verification Cross Reference Matrix

System Requirement	Para. No.	Verif. Method	Test Level	Test Resp.	Level 4&5 Test ID
Video from external inputs shall be selectable internal to the ITS for viewing on monitors within the system.	3.3.3.7.1(1)	D	4	ITS Consultant	2
The external video inputs shall selectable and viewable from any location having the capability to switch and monitor the roadway camera video.	3.3.3.7.1(2)	D	4	ITS Consultant	2
ITS shall provide the capability to accept, switch, and monitor local television and cable and/or direct broadcast satellite channels for weather, traffic, and incident information in compliance with the constraints of agreements and rights of these sources.	3.3.3.7.1.1(1)	A/D	3	ITS Consultant	
			4	ITS Consultant	2
Audio from television channels shall route to the display location(s) of the corresponding video that have an associated audio output capability.	3.3.3.7.1.1(2)	A/D	3	ITS Consultant	
		D	4	ITS Consultant	1,2
The ITS video switching network shall provide connections to accept video "feeds" transmitted from mobile camera equipment mounted on helicopters, airplanes, or motorized vehicles for localized surveillance of areas not covered by the fixed camera installations.	3.3.3.7.1.2(1)	I/C	2	Installation Contractor/ Vendor	
		D	3	ITS Consultant	
		D	4	ITS Consultant	2
The ITS shall provide video output connections for use by the broadcast media for dissemination external to the ITS.	3.3.3.7.2(1)	I/C	2	Installation Contractor/ Vendor	
		D	3	ITS Consultant	
		D	4	ITS Consultant	2

Table A-1 Verification Cross Reference Matrix

System Requirement	Para. No.	Verif. Method	Test Level	Test Resp.	Level 4&5 Test ID
Access controls shall be established to prevent unauthorized access.	3.3.3.7.2(2)	I/C D D	2 3 4	Installation Contractor/ Vendor ITS Consultant ITS Consultant	2
Video recording capability shall be provided in each TMC and the Highway Patrol Dispatch Center.	3.3.3.8(1)	I/C	2	Installation Contractor/ Vendor	
Each operator workstation shall have access to a video recording capability.	3.3.3.8(2)	I/C D D	2 3 4	Installation Contractor/ Vendor ITS Consultant ITS Consultant	2
The operators shall have the capability to record and playback any video available at the workstation.	3.3.3.8(3)	I/C D D	2 3 4	Installation Contractor/ Vendor ITS Consultant ITS Consultant	2

Table A-1 Verification Cross Reference Matrix

System Requirement	Para. No.	Verif. Method	Test Level	Test Resp.	Level 4&5 Test ID
The recording capability shall provide control and accessibility determined through an access control mechanism.	3.3.3.8(4)	I D D	2 3 4	Installation Contractor/ Vendor ITS Consultant ITS Consultant	2
All video source recording shall be performed in compliance with the constraints of agreements and rights of these sources.	3.3.3.8(5)	A/D D	3 4	ITS Consultant ITS Consultant	2
A minimum of two (2) record and playback units shall be provided at each TMC.	3.3.3.8(6)	I/C	2	Installation Contractor/ Vendor	
The Highway Patrol Dispatch Center shall have a minimum of one (1) record and playback unit.	3.3.3.8(7)	I/C	2	Installation Contractor/ Vendor	

Appendix B. Sample Test Procedure

This appendix contains a sample system level test procedure for Dynamic Message Signs.

Intelligent Transportation System

System Test Procedure

Test 3 - Data Functional Element

Technical Test Procedure (TTP) - 3.2 Dynamic Message Signs (DMS)

1 Objective and Scope

This test procedure shall verify, by demonstration, that the ITS provides data for display by the DMS, monitors the DMS to detect failures, logs failures to the database, reports failures to the operator, takes the failed equipment off-line, and alerts maintenance personnel of failures.

This technical test procedure (TTP) shall be executed from a control position within the TMC constituting a single test case.

2 Requirements Traceability

This test shall partially verify the following system level requirements:

> *3.3.4.1.1(1) The ITS shall provide data such that DMS can disseminate information advising the traveler.*
>
> *3.3.4.4.1(2) DMS shall provide periodic feedback as to operational status, diagnostic messages, and displayed message.*
>
> *3.3.4.4(3) The system shall also monitor all of the field equipment under its control, and shall identify any component failures.*
>
> *3.3.4.6(6) At a minimum, the following events shall be logged:*
>
> - *Traffic Management.*
>   - *Incident detection and clearance.*
>   - *Events scheduled.*
>   - *Message and/or control and status changes for DMS, HAR, HAT, (future) TCS, (future) RMS, etc.*
> - *System Management.*
>   - *System restart/shutdown.*
>   - *Equipment on-line/off-line.*
>   - *User log-in/log-out.*
>   - *Data file archive, or retrieval.*
>   - *System initialization.*
> - *Equipment Status Changes.*
>   - *Power failure recovery.*
>   - *Computer system failure.*
>   - *Any system hardware failure or restoration.*
>   - *Any field hardware failure or restoration.*
>   - *Any system parameter modifications or re-configurations.*
>
> *3.3.4.6.1(5) The database shall provide accurate and current information for dissemination to the motoring public through ATIS, the broadcast media, and other users.*
>
> *3.10(1) The ITS shall have the ability to monitor the operation of those components which provide status, detect failures, log failures to the database, report failures to the operator, take the failed equipment offline, and alert maintenance personnel of failures.*

> *3.10(2) Equipment within the ITS shall be monitored to detect any potential errors or reduced performance (provided the equipment will support bi-directional communication).*

Completion of this procedure and TTPs- 1.3, 1.4, 3.3, 3.5 and 4.5 shall fully verify the requirements.

3 Prerequisites/Special Requirements

Level 3 verification testing shall have been successfully completed on the applicable portions of the DMS control functionality prior to this test.

4 Limitations

This test shall be limited to a demonstration of the DMS capability only.

5 Special Test Equipment

No special test equipment is required for this test.

6 Test Configuration

6.1 Equipment

The following equipment shall be configured to support this test.

At the TMC:

- Application Server
- Database Server
- Operator Workstation

At the Highway Patrol Dispatch Center:

Not applicable to this test.

In the Field:

- Representative DMS shall be utilized to demonstrate the capability of displaying messages to the motorist.

6.2 Data Requirements

A list of approved DMS messages shall be pre-loaded into the TMC database prior to the start of this test. At least two distinct messages shall be available for each DMS involved in the test configuration. DMS test messages shall include single- and two-phase messages where appropriate for the DMS sign type being tested. Two-phase messages shall demonstrate specified blanking intervals between message phases and message updates. The message content shall identify the particular DMS and clearly indicate that it is a test message.

7 Resources

7.1 Facilities

Use of the following facility is required to perform this test procedure:

- TMC.

7.2 Personnel

The following personnel are required to accomplish this test procedure (exclusive of support personnel required to configure the system to support the test):

- Test Conductor.
- Test Operator.
- Test Witness.
- Maintenance Technician (single DMS location).
- Test Observer (DMS Locations).

8 Test Description and Procedure Steps

This test shall demonstrate the capability to select messages from a library of messages, download them to a DMS for display, monitor the DMS, log and report failures. This test will also demonstrate the system's ability to bring a DMS back on line after a failure has been detected.

A test observer will be used to view different messages displayed at selected DMS sites. This observer will use a cellular phone or a radio to communicate DMS display results.

The detailed steps to be performed for each test case of this procedure follow. The procedure worksheets are to be completed and signed by the test witness(es). The test conductor will also sign the completed worksheet.

ITS TEST PROCEDURE

TTP-3.2 Dynamic Message Signs

Start Date: ___________ Time: ___________

Test Case No. __1___ of __1__ Cases for this Procedure.

Test Site(s) Involved:
[x] Primary TMC
[ ] Secondary TMC
[ ] Highway Dispatch Center
[x] Field
[ ] Other (specify) __________________________

Test Conductor:
____

Test Witness(es): __________________ __________________ __________________ __________________

Test Observer(s): __________________ __________________ __________________ __________________

Test Operator(s): __________________ __________________ __________________ __________________

Step No./ Check Off	Procedure Steps	Expected Results/ Comments/ Test data	Results Pass (P) or Fail (F)
1 [ ]	Record the console identifier for this operator position.		
2 [ ]	On the display screen enter the username and password of the test user with access to all system commands.	The User Interface consisting of a Main Menu, Alarm Status Window and GIS map is displayed. The GIS map also contains a menu of commands.	
3 [ ]	Select Freeway > Dynamic Message Sign > DMS Unit Control from the Main Menu pull down.	A Dynamic Message Sign Status and Unit Control Screen appears.	
4 [ ]	Click on the Browse Button	Dynamic Message Sign Selection Screen Appears displaying a list of available DMSs.	
5 [ ]	Record identifier for Dynamic Message Sign Selection.	Dynamic Message Sign Selection Unit # ____________ Location _________________	

Step No./ Check Off	**Procedure Steps**	**Expected Results/ Comments/ Test data**	**Results Pass (P) or Fail (F)**
6 [ ]	Select the Dynamic Message Sign Unit and click on the OK button.	Dynamic Message Sign Selection Screen disappears.	
7 [ ]	View Unit # and Location in the Selected DMS on the Dynamic Message Sign Control Screen.	A DMS matching the Unit # in step 5 is displayed.	
8 [ ]	Click on the Sign Control Button.	A Sign Control Screen opens.	
9 [ ]	Click Yes and OK.	The DMS is in a manually controlled mode.	
10 [ ]	Select and Read Message to be sent to the DMS.		
11 [ ]	Send the Message		
12 [ ]	Verify the message sent by having a field technician to communicate the display.		
13 [ ]	Repeat steps 10 - 12 four additional times.		
14 [ ]	Repeat steps 4 - 13 two additional times at other DMS sites.		
15 [ ]	Request Maintenance Technician to disconnect the communication interface that corresponds to the last DMS Unit # recorded in step 5.		
16 [ ]	Click the Configuration Parameters button and view the Loss Default Message # text field.	A Loss Default Message number is displayed.	
17 [ ]	Click the ... button next to the Loss Default Message # text field.	A Text Message Display Screen opens with the Message Name and the Phase 1 and Phase 2 Text Messages.	
18 [ ]	Request a reading of DMS display from Test Observer.	The DMS message is the same as the Loss Default Message in the above step.	
19 [ ]	View the Status/Alarm Management list box under the Main Menu.	An alarm appears in the Status/Alarm Management window. This alarm displays as a category 2.	

Step No./ Check Off	Procedure Steps	Expected Results/ Comments/ Test data	Results Pass (P) or Fail (F)
20 [ ]	Record the identifier for the new alarm displayed in the Status/Alarm Management window of the Main Menu.	Time Alarm Received ______ Category _________________ Priority _________________	
21 [ ]	In the Status/Alarm Management Window of the Main Menu, click on the new alarm.	The Alarm Details/Comments Display Window appears.	
22 [ ]	Input Alarm comments information and click OK.	The Alarm Details/Comments Display Window closes.	
23 [ ]	Select Events > On Line Viewer from the Main Menu.	An Event Log Viewer Screen opens with a list box of Events.	
24 [ ]	View List Box.	An Event with an alarm time matching that in step 11 and a category 2 should be present in the list. Also, the event type ID should be 403.	
25 [ ]	Click on the Close Button.	Event Log Viewer closes.	
26 [ ]	Select Freeway > Dynamic Message Sign > DMS Unit Control from the Main Menu pull down.	A Dynamic Message Sign Status and Unit Control Screen appears.	
27 [ ]	Click on the Browse Button	Dynamic Message Sign Selection Screen Appears displaying a list of available DMSs.	
28 [ ]	Select the Dynamic Message Sign Unit # that was disconnected in step 15 and click on the OK button.	Dynamic Message Sign Selection Screen disappears.	
29 [ ]	View Unit #, Location and Status on the Dynamic Message Sign Control Screen.	The Unit # and Location should match the Unit # and Location recorded in Step 5. The Status should indicate that the Device is Not Active.	
30 [ ]	Select Maintenance > Inventory > Set Status from the Main Menu.	The Inventory - Set Status Screen opens.	
31 [ ]	Select the Inventory ID matching the failed DMS and click on the Out of Service toggle button in the Status Form and the Save button on the Inventory - Set Status screen.	The Inventory - Set Status screen disappears.	

Step No./ Check Off	**Procedure Steps**	**Expected Results/ Comments/ Test data**	**Results Pass (P) or Fail (F)**
32 [ ]	Request Maintenance Technician to reconnect the communication interface to the DMS.		
33 [ ]	Select Freeway > Dynamic Message Sign > DMS Unit Control from the Main Menu pull down.	A Dynamic Message Sign Status and Unit Control Screen appears.	
34 [ ]	Click on the Browse Button	Dynamic Message Sign Selection Screen Appears displaying a list of available DMSs.	
35 [ ]	Select the Dynamic Message Sign Unit # that was disconnected in step 10 and click on the OK button.	Dynamic Message Sign Selection Screen disappears.	
36 [ ]	View Unit #, Location and Status on the Dynamic Message Sign Control Screen.	The Unit # and Location should match the Unit # and Location recorded in Step 5. The Status should indicate the Device is Active.	
37 [ ]	Select and Send a message to the DMS.		
38 [ ]	Request a reading of the DMS display from the field Test Observer.	Message is the same as the message sent to the DMS.	
39 [ ]	Click the Sign Control Button on the DMS Control Screen.	A Sign Control Screen opens.	
40 [ ]	Click No and OK	The Sign Control Screen closes.	
41 [ ]	Click the Close Button on the Dynamic Message Sign Control Screen.	The Dynamic Message Sign Control Screen Closes.	
42 [ ]	Select Maintenance > Inventory > Set Status from the Main Menu.	The Inventory - Set Status Screen opens.	
43 [ ]	Select the Inventory ID matching the failed DMS and click on the In Service toggle button in the Status Form and the Save button on the Inventory - Set Status screen.	The Inventory - Set Status screen disappears.	
44 [ ]	Select Admin > Exit from the Main Menu.	End of Test.	

Completion Date: ____________ Time: ____________

____

Test Conductor

____

Test Witness(es)

Notes:

Appendix C. Sample System Problem / Change Request (SPCR) Form

This appendix contains a sample System Problem/Change Request (SPCR) form. This form is used to formally record and describe a system problem and/or to request a change to a Configuration Item (CI) (i.e., document, drawing, hardware and software) and track its review, assessment and approval or rejection by the CCB.

SPCR Form Responsibility

The SPCR originator, Change Assessment and Resolution (CAR) Leader (if necessary), CM Manager, and CCB Chair all have responsibilities in completing different sections/items on the SPCR form. These responsibilities are detailed below.

- The Originator (SPCR form white area items 1 through 15) – the originator starts the SPCR process. After determining that the problem/proposed change affects a CM controlled item, the originator completes items 1 through 15 on the SPCR form. If an assessment is required or no specific action is being recommended, the SPCR needs to be assigned to a CAR leader and items 7, 8 and 15 should be left blank. The originator should complete items 9, 10, 11, 12 and 13, if known. If the originator cannot determine if the SPCR affects a CM controlled item or if the SPCR needs to be assigned to a CAR leader, the originator should contact the CM Manager.
- The CAR Leader (SPCR form white area items 1 through 15) – the CAR leader (assigned by the CM Manager) is responsible for the accuracy of the information provided by the originator and for the completion and/or validation of items 7 through 15.
- The CM Manager (SPCR form shaded area items 16 through 27) – the CM Manager has the overall responsibility for the integrity of the form including the accuracy of the original information on the form and for completing or revising information during the SPCR life cycle. The CM Manager assigns the SPCR numbers (items 25 and 26) and maintains the SPCR History (item 27).
- The CCB Chair (SPCR form items 28 and 29) – the CCB Chair signs and dates the SPCR form validating the CCB Action (approval or rejection) for SPCR disposition and authorizes the recommended action for approved SPCRs.

SPCR Item Description

Listed below is a description of each individual item on the SPCR form that follows. The number of the item corresponds with the number in each box of the form.

1. **Originator: –** Name of individual initiating the requested change.
2. **Organization: –** The name of the organization/group the originator represents.
3. **Date: –** Date that originator completes the SPCR.
4. **Subject: –** An appropriate subject title depicting the problem or requested change.
5. **Type of Change:**
   a) **Stand Alone -** The “stand alone” SPCR is one that is normally a simple change. This could be a wording change to a document, a view change on a drawing or a simple change to software that normally does not affect any other area and requires no additional assessment by others. In this type of change the originator knows exactly what document number, drawing number or software version is affected.

b) **Assessment Required** - Used when investigation is required for total change impact on all areas. This will require an assignment of a CAR Leader to study the problem and prepare supporting information for a recommended action.

6. **Reason for Change:**
   a) **Defect/Error** - Reason for change is due to a defect or error to the original baseline item or past revision or version.
   b) **Improvement -** An improvement to existing hardware, software, etc.
   c) **Addition** - This reason results in change to existing CM data due to additional new requirements.
7. **Affects: –** A change may affect one or more of the following. Check all that apply.
   a) **Software -** Indicate if requested change affects software.
   b) **Hardware -** Indicate if requested change affects any hardware. Note that all hardware and cabling requested changes must be depicted in drawing concepts.
   c) **Documents** - Indicate if requested change affects any documents.
8. **Priority:**
   a) **Emergency** - This is authorized if a problem or change request must be resolved quickly to avoid a major impact to a particular operation, design, method, etc. Depending on when the problem is found in relationship to a scheduled CCB meeting, this priority SPCR may be acted upon by the CM Manager and Program Manager in advance of meeting. If an SPCR is handled as an emergency, it still must be brought before the CCB to establish results and closure.
   b) **Urgent** - An urgent request is one that can normally can wait for the next scheduled CCB meeting but the originator feels that it has a high impact to operations, design, methods, etc. An urgent request is one that should be acted on quickly and perhaps as a priority item.
   c) **Routine** - A routine request is one that should be implemented to best-fit schedule.
9. **CSCI(ver): -** The name (and version) of the affected Computer Software Configuration Item. If it is found that the originator or CAR Leader has missed a CSCI or entered an incorrect number or version, the CM Manager will make the appropriate change or addition as a result of the CCB meeting.
10. **HWCI(ver):** - The name (and revision) of the affected Hardware Configuration Item. If it is found that the originator or CAR Leader has missed a HWCI or entered an incorrect number or revision, the CM Manager will make the appropriate change or addition as a result of the CCB meeting.
11. **Other Impacts: -** If the problem or requested change will impact other areas such as operational or maintenance procedures, software build, training, personnel, cost, schedule, etc. indicate here and include a description of the impact(s) in item 14.
12. **Related SPCR(s):** - If there are other related SPCRs enter their identification numbers. If this is an SPCR that was generated as CM Manager action to track different parts (e.g. hardware, software and/or documentation) of the original problem/change request separately leave this item blank. The CM Manager will complete this item and items 27 and 28 to indicate related (child and parent respectively) SPCRs.
13. **DOC/DRW No.(rev):** - Enter all affected document and drawing numbers (and revisions). If it is found that the originator or CAR Leader has missed a document or drawing number or entered an incorrect number or revision, the CM Manager will make the appropriate change or addition as a result of the CCB meeting.
14. **Description of Problem/Change: –** The description of the problem or change request should be complete enough for the CCB and/or the CAR Leader to understand. If the space provided is not adequate, or if additional documentation is required, a continuation sheet should be used. In this case, a general paragraph must still be entered with the "**Additional Data Attached**" block checked. If the description of problem/change is continued on additional sheets, include the

heading "Description of Problem/Change Cont'd. – SPCR No. ____ Sheet ___ of ___" on each additional sheet. The CM Manager will add the SPCR No. to the continuation sheet(s). Number the Description of Problem/Change continuation sheets consecutively starting with one and include the total number of continuation sheets used.

15. **Recommended Action: –** A paragraph must be entered for the recommended action. If the space provided is not adequate, or if additional supporting information is required, a continuation sheet should be used. In this case, a general paragraph must still be entered with the "**Supporting Data Attached**" block checked. Supporting data is any information that the CCB may need to validate the recommended action. This may be information such as cost impact (hours or dollars), schedule delays, other alternate approaches and their results. If the recommended action is continued on additional sheets, include the heading "Recommended Action Cont'd. – SPCR No. ____ Sheet ___ of ___" on each additional sheet. The CM Manager will add the SPCR No. to the continuation sheet(s). Number the Recommended Action continuation sheets consecutively starting with one and include the total number of continuation sheets used.
16. **CAR Leader: –** The name of the CAR Leader assigned by the CM Manager.
17. **Date Due: –** The date set by agreement between the CM Manager and the CAR Leader to have analysis complete. This date should be a minimum of 3 days prior to a scheduled CCB meeting.
18. **CCB Mtg: –** The date of the CCB meeting.
19. **Name: –** If the CCB requires additional information to be submitted prior to change approval, the name(s) of the individual(s) assigned to provide the information are listed.
20. **Assignment: –** A brief explanation of the individual's task assignment is listed.
21. **Due Date: –** The date that the task assignment is due.
22. **CCB Action:** - SPCR disposition. Not valid without the signature of the CCB Chair's signature in item 28 and the date in item 29.
    a) **Pre Approved** – Checked if the CCB gives tentative approval based on the future submittal of additional information.
    b) **Approved** – Checked when CCB authorizes changes to be made to existing CM controlled items.
    c) **Rejected** – Checked if the CCB determines that the proposed SPCR is not valid.
23. **Notes: –** The CM Manager will enter any pertinent information as a result of the CCB meeting.
24. **Update Assignments:**
    a) **Doc/Dwg/Software No. –** Everything under CM control has a number assigned to it. The number of the document, drawing or software to be updated is listed here.
    b) **Old Rev/Ver –** For documents and drawings the letter entered will be the current revision. For software, the current version number will be entered.
    c) **New Rev/Ver –** For documents and drawings the letter entered will be next letter higher than the old letter. For software, the version number will be assigned by the CCB. Note- There may be times when a software version cannot be assigned for a future build. In this case the word "open" will be used.
    d) **Update By –** The name of the individual responsible for the actual update. If an individual cannot be named, then the organization/group's name will be entered.
    e) **QA By –** The name of the individual responsible to ensure the integrity of the updated item(s).
    f) **Due Date –** The date that all data is due back from QA for re-entry into CM Library/Repository. This will be determined by the CM Manager.
25. **SPCR No.: –** The SPCR tracking number issued by the CM Manager. A four-digit number starting with 0001. For a parent SPCR, the CM Manager should also enter in item 12 (Related SPCRs) the list of child SPCRs created. If this is a child SPCR, enter the parent SPCR number followed by the child's letter designator, e.g. 0001A.

26. **Parent No.: –** Use for child SPCRs only. When an original SPCR is split, resulting in two or more SPCRs the number of the original (parent SPCR is entered).
27. **SPCR History**
    a) **Log Date –** The date when the SPCR was received by the CM Manager and entered into the status log.
    b) **Ready CCB –** The date that the SPCR is ready for the CCB.
    c) **CCB Date –** The CCB action date for the disposition of the SPCR.
    d) **Close Date –** The date that all work is completed, validated and returned to CM Library/Repository.
    e) **CM Manager –** The name of the individual that validated the SPCR and reviewed updated item(s).
28. **CCB Chair Signature:** – Signature of CCB Chair (or designee) validating the CCB Action (item 22) for the SPCR disposition.
29. **Date:** - Date signed by CCB Chair.

SH 1 of ____ (bottom right of form) – If attachments are included as part of the SPCR, add their number and enter the total number of sheets.

ITS – System Problem/Change Request (SPCR) Form

25. SPCR NO.

26. PARENT NO.

ORIGINATOR/CAR LEADER

1. ORIGINATOR:	**2. ORGANIZATION:**	**3. DATE:**	**27. SPCR HISTORY**
4. SUBJECT:			A LOG DATE:

5. TYPE OF CHANGE	**6. REASON FOR CHANGE**	**7. AFFECTS**	**8. PRIORITY**	B READY CCB:
A □ STANDALONE	A □ DEFECT/ERROR	A □ SOFTWARE	A □ EMERGENCY	C CCB DATE:
B □ ASSESSMENT REQUIRED	B □ IMPROVEMENT	B □ HARDWARE	B □ URGENT	D CLOSE DATE:
	C □ ADDITION	C □ DOCUMENT(S)	C □ ROUTINE	E CM MGR:

9. CSCI (VER):	**10. HWCI (REV):**	**11. OTHER IMPACTS:**	**12. RELATED SPCR(S)**
13. DOC/DRW NO. (REV):			

14. DESCRIPTION OF PROBLEM/CHANGE □ **ADDITIONAL DATA ATTACHED**

15. RECOMMENDED ACTION □ **SUPPORTING DATA ATTACHED**

CM MANAGER

16. CAR LEADER:	**17. DATE DUE:**	**18. CCB MTG:**	**22. CCB ACTION:**
19. NAME	**20. ASSIGNMENT**	**21. DUE DATE**	A □ PRE APPROVED
			B □ APPROVED
			C □ REJECTED

23. NOTES:

24. UPDATE ASSIGNMENTS

A. DOC/DWG/SW NO.	**B. OLD REV/VER**	**C. NEW REV/VER**	**D. UPDATE BY**	**E. QA BY**	**F. DUE DATE**

28. CCB CHAIR SIGNATURE **29. DATE**

FORM NO. 03-001 REV. A DATE: 04/01/01 **SH 1 OF**

Appendix D. Example Application of NTCIP Standards

As an example of how NTCIP standards are applied, the following contains excerpts from Section 4, "Transportation Management Center Communications Specifications" from the FDOT draft specification for DMS devices titled *Dynamic Message Sign Specifications for Limited-Access Facilities*. The specification references specific NTCIP standards and the specific MIBs to be implemented for communication with the DMS field devices (to be deployed in Florida).

Dynamic Message Sign Specifications for Limited-Access Facilities

Version 5 (January 19, 2005)

This specification replaces the FDOT TERL Permanent Mount Dynamic Message Sign Minimum Specifications dated September 2001.

4. Transportation Management Center Communication Specifications

The sign controller shall be addressable by the TMC through the Ethernet communication network. Software shall comply with the NTCIP 1101 base standard (formerly the NEMA TS 3.2-1996 standard), including all amendments as published at the time the Request for Proposals (RFP) is released, and the NTCIP Simple Transportation Management Framework (STMF), and shall conform to Compliance Level 1. Software shall implement all mandatory objects as defined in the FDOT-standard Global MIB v01c in Appendix A of that document.[4] Software shall implement all mandatory objects as defined in the FDOT-standard DMS MIB v01c in Appendix B of that document. Software shall implement all mandatory objects as defined in the FDOT-specific DMS MIB v01c in Appendix C of that document. The DMS shall comply with the NTCIP 1201 v01, 1203 v01, 2101 v01.19, 2103 v01.13, 2201 v01.14, 2202 v01.05, and 2301 v01.08 standards. Software may implement additional objects, but the additional objects shall not interfere with the standard operation of any mandatory objects.

Each DMS shall provide full, standardized range support for all objects required by these specifications unless otherwise detailed in the plans. The standardized range is defined by a size, range, or enumerated listing indicated in the object's syntax field and/or through descriptive text in the relevant standard object description field. The DMS maximum response time for any object or group of objects shall be 200 milliseconds unless otherwise indicated in the plans, or unless approved by the FDOT Traffic Engineering Research Laboratory (TERL). Deviances from the full ranges for objects are detailed in Table 4.1.

Table 4.1 – Range Deviances for Objects

Object	Minimum Project Requirements
FDOT Global MIB v01c	
Maximum Event Log Configurations	50
Event Configuration Mode	2, 3, and 4
Maximum Event Log Size	200
Maximum Event Classes	7
Maximum Group Address	1

FDOT DMSMIB v01c	
Number of Fonts	4
Maximum Font Characters	255
Default Background Color	0
Default Foreground Color	2, 7, 8, or 9
Default Justification Line	2, 3, 4
Default Justification Page	2, 3, 4
DMS – Number of Permanent Messages	0
DMS – Maximum Changeable Messages	50
DMS – Maximum Volatile Messages	0
Nonvolatile Memory	5 KB
DMS – Control Mode	2, 3, 4, and 5
Number of Action Table Entries	15
Number of Brightness Levels	255

The software shall implement the tags (opening and closing where defined) of MULTI as detailed in Table 4.2 and as defined in the NTCIP 1203 standard.

Table 4.2 – NTCIP 1203 Standard Software Tags *

Opening Tag	Closing Tag	Explanation
cbx		Color – Background – The background color for a message.
cfx		Color – Foreground – The foreground color for a message.
fx,y		Field – The information to embed within a message that is based on data from some device, such as a clock, calendar, temperature sensor, detector, etc. The following field tag values (IDs) are REQUIRED to be supported: 1 – the time in a 12-hour format 2 – the time in a 24-hour format 4 – the ambient temperature in degrees Fahrenheit 7 – the day of the week 8 – the date of the month 9 – the month of the year 10 – the year in two digits 11 – the year in four digits
fltxoy	/fl	Flash – Activate flashing of the text; define the flash-on and flash-off times; and the flash order (i.e., on/off or off/on).

fox		Font – Selects a font number (as specified in the font table) for the message display.
jlx		Justification – Line – Specify line justification: left, center, right, or full. However, full justification is not required.
jpx		Justification – Page – Specify page justification: top, middle, or bottom placement.
mvtdw,s,r,text		Moving – Text – Specify the parameters of a horizontal moving (scrolling) text.
nlx		New – Line – Specify the start of a new line.
np		New – Page – Specify the start of a new page.
ptxoy		Page – Time – Specify the page times (t = on , o = off).
scx	/sc	Spacing – Character – Specify the spacing between characters.

* The letters “x” and “y” are character placeholders, usually for numbers, that specify the tag parameter(s). See the *NTCIP1203* standard and its amendments for further definitions.

Refer to Sections 5.9 and 5 .10 contained herein as they relate to software licenses and intellectual property. All center-to-field device communications shall be nonproprietary.

[4] Information on the DMS Management Information Base (MIB) is available online at http://www.dot.state.fl.us/trafficoperations/fdot_dms_mib_faq.htm

Each sign controller shall be provided with error detection and reporting features that will be used to guard against incomplete or inaccurate transmissions. These features shall include:

- Cyclic redundancy checking of all data received from the TMC, with positive acknowledgment for all valid transmissions
- Status monitoring for communication line malfunctions or breakages
- Content validation for all transmissions received for logic or data errors

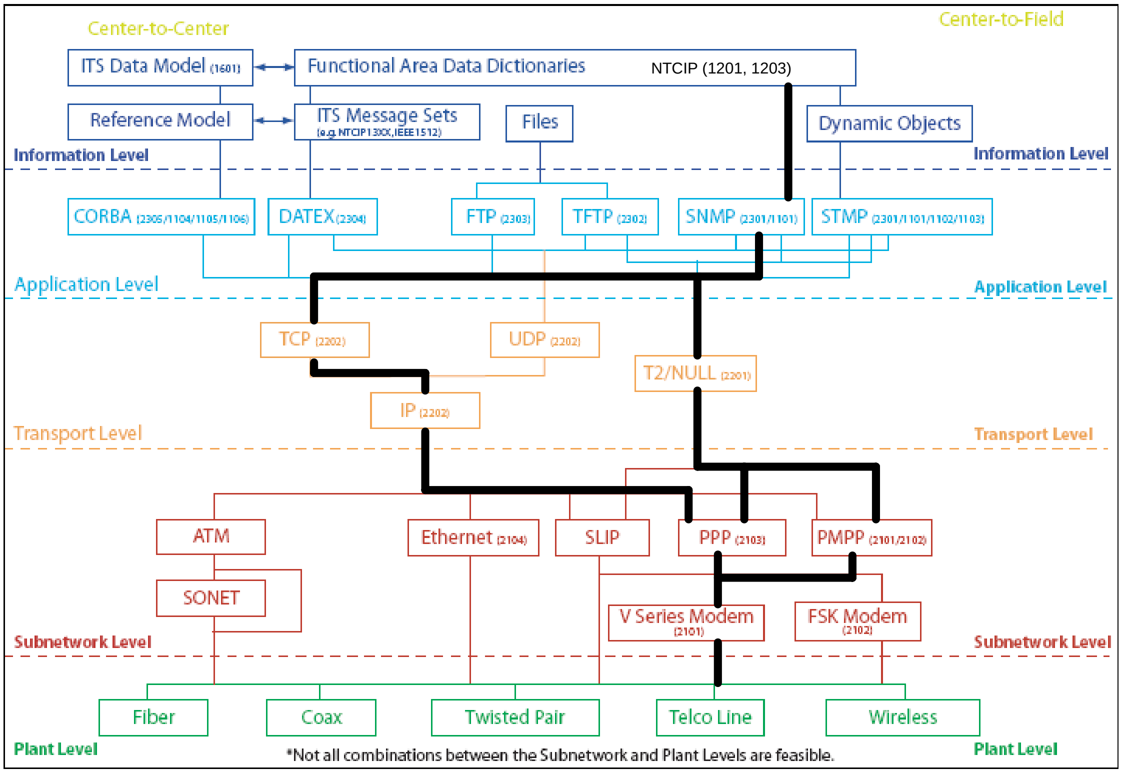
The communication line circuits shall be point-to-point or multipoint, and shall be full duplex asynchronous data transmissions at the rate directed by the Engineer.

Each sign controller shall be assigned a unique address in the circuit that the sign is connected to. Where applicable, all data transmitted between the TMC and the sign controller shall be encoded using 1 start bit, 8 data bits, and 1 stop bit.

The following NTCIP standards are shown in the center-to-field (C2F) NTCIP DMS standards framework diagram below. Those that are compliance standards in the FDOT DMS specification are listed in bold print below and shown on the highlighted paths in the diagram.

- **NTCIP 1101: Simple Transportation Management Framework (STMF).**
- NTCIP 1102: Base Standard: Octet Encoding Rules (OER).
- NTCIP 1103: Simple Transportation Management Protocol (STMP).
- **NTCIP 1201: Global Object Definitions.**
- **NTCIP 1203: NTCIP Object Definitions for Dynamic Message Signs (DMS).**
- **NTCIP 2101: Point to Multi-Point Protocol Using RS-232 Subnetwork Profile.**
- NTCIP 2102: Subnet Profile for PMPP Over FSK modems.
- **NTCIP 2103: Subnet Profile for Point-to-Point Protocol using RS 232.**
- NTCIP 2104: Subnet Profile for Ethernet.
- **NTCIP 2201: Transportation Transport Profile.**
- **NTCIP 2202: Internet (TCP/IP and UDP/IP) Transport Profile.**
- **NTCIP 2301: Application Profile for Simple Transportation Management Framework (STMF).**
- NTCIP 2302: Application Profile for Trivial File Transfer Protocol.
- NTCIP 2303: Application Profile for File Transfer Protocol (FTP).

Figure D-1 C2F NTCIP DMS Standards Framework



Based on the NTCIP Standards specified in the FDOT DMS specification, it is clear that the desired DMS C2F communications implementation is intended to use SNMP at the application level, support TCP/IP or T2/NULL at the transport level, and use point-to-point or point-to-multi-point protocol using a RS-232 interface to a V series modem at the subnetwork level.

Testing Programs for Transportation Management Systems: A Technical Handbook

Federal Highway Administration
U.S. Department of Transportation
400 7th Street, S.W. (HOP)
Washington, DC 20590
Toll-Free "Help Line" 866-367-7487
www.ops.fhwa.dot.gov

Publication No.: FHWA-HOP-07-088
EDL Document No.: 14353