

Guide for Implementing IEEE Std 1512TM

Using a Systems Engineering Process



**Published by
Standards Information Network
IEEE Press**

Guide for Implementing IEEE Std 1512TM Using a Systems Engineering Process

Prepared for

Federal Highway Administration (FHWA)

&

Institute of Electrical and Electronics Engineers (IEEE)

Prepared by

Klotz Associates, Inc.

**Consensus Systems Technologies
SubCarrier Systems Corporation**

4 November 2008

Trademarks and Disclaimers

IEEE believes the information in this publication is accurate as of its publication date; such information is subject to change without notice. IEEE is not responsible for any inadvertent errors.

Library of Congress Cataloging-in-Publication Data

IEEE 1512 Implementation Guide
deliverable 5 / prepared for Federal Highway Administration (FHWA) &
Institute of Electrical and Electronics Engineers (IEEE) ;
prepared by Klotz Associates, Inc. in association with Consensus Systems Technologies,
SubCarrier Systems Corporation.

p. cm.
Includes bibliographical references.
ISBN 978-0-7381-5674-3

1. Emergency communication systems--United States--Standards. 2. Traffic safety--United States--Data processing--Standards. 3. Emergency management--United States--Data processing--Standards. I. United States. Federal Highway Administration. II. Institute of Electrical and Electronics Engineers. III. Klotz Associates.

TK5105.55.I33 2007
621.382021'873--dc22

2007062029

The Institute of Electrical and Electronics Engineers, Inc.
3 Park Avenue, New York, NY 10016-5997, USA

Copyright © 2008 by Institute of Electrical and Electronics Engineers, Inc.
All rights reserved. Published 4 November 2008. Printed in the United States of America.

No part of this publication may be reproduced in any form, in an electronic retrieval system, or otherwise, without the prior written permission of the publisher.

IEEE is a registered trademark of the Institute of Electrical and Electronics Engineers, Incorporated (www.ieee.org/).

IEEE Standards designations are trademarks of the Institute of Electrical and Electronics Engineers, Incorporated.

IEEE Press/Standards Information Network publications are not consensus documents. Information contained in this and other works has been obtained from sources believed to be reliable, and reviewed by credible members of IEEE Technical Societies, Standards Committees, and/or Working Groups, and/or relevant technical organizations. Neither the IEEE nor its authors guarantee the accuracy or completeness of any information published herein, and neither the IEEE nor its authors shall be responsible for any errors, omissions, or damages arising out of the use of this information.

Likewise, while the author and publisher believe that the information and guidance given in this work serve as an enhancement to users, all parties must rely upon their own skill and judgment when making use of it. Neither the author nor the publisher assumes any liability to anyone for any loss or damage caused by any error or omission in the work, whether such error or omission is the result of negligence or any other cause. Any and all such liability is disclaimed.

This work is published with the understanding that the IEEE and its authors are supplying information through this publication, not attempting to render engineering or other professional services. If such services are required, the assistance of an appropriate professional should be sought. The IEEE is not responsible for the statements and opinions advanced in the publication.

Review Policy

The information contained in IEEE Press/Standards Information Network publications is reviewed and evaluated by peer reviewers of relevant IEEE Technical Societies, Standards Committees and/or Working Groups, and/or relevant technical organizations. The authors addressed all of the reviewers' comments to the satisfaction of both the IEEE Standards Information Network and those who served as peer reviewers for this document.

The quality of the presentation of information contained in this publication reflects not only the obvious efforts of the authors, but also the work of these peer reviewers. The IEEE Press acknowledges with appreciation their dedication and contribution of time and effort on behalf of the IEEE.

To order IEEE Press Publications, call 1-800-678-IEEE.

PDF: ISBN 978-0-7381-5674-3 STDSP1161

See other IEEE standards and standards-related product listings at:

<http://standards.ieee.org/standardspress/>

Acknowledgement

Grateful acknowledgement is made to the authors of this Implementation Guide, including Sriram Natarajan, P.E. of PTOE and Michael Ogden, P.E. of Klotz Associates, Inc., Manny Insignares, of Consensus Systems Technologies, and David Kelley of SubCarrier Systems Corporation. In addition, special thanks to Tom Stout, FHWA and the IEEE 1512 Incident Management Working Group (IMWG) for their support and guidance throughout the development process of this Implementation Guide.

Guide for Implementing IEEE 1512 Using a Systems Engineering Process
Contents

Table of Contents

1	Introduction.....	1
1.1	Purpose of This Guide.....	1
1.2	IEEE 1512 Overview	2
1.3	IEEE 1512 System Interfaces.....	2
1.4	System Interface Life Cycle Process.....	4
1.4.1	The Systems Engineering Process (SEP).....	4
1.4.2	Standards That Support the Systems Engineering Process.....	5
1.5	Tailored System Interface Life Cycle Process	6
1.6	System Acquisition	11
1.7	Document Organization	12
2	Project Initiation Phase	13
2.1	Project Management Plan.....	13
2.2	Systems Engineering Management Plan	13
2.3	Configuration Management Plan	14
2.4	Verification and Validation Plan.....	14
2.5	Risk Management Plan.....	15
3	Concept Definition Phase	17
3.1	Documenting User Needs and Developing a Concept of Operations (ConOps)	17
3.2	Standards that Support ConOps Development.....	17
3.3	ConOps Content for Systems Interfaces	18
3.3.1	Current Situation / Problem Scope	19
3.3.2	Identification of User Classes and Roles	19
3.3.3	Operational Scenarios	23
3.4	User Needs Wording	31
3.5	Summary	33
4	Requirements Elicitation Phase	34
4.1	System Interface Requirements Identification	34
4.1.1	What are Requirements?	34
4.1.2	Attributes of Good Requirements	34
4.1.3	Standards That Support System Requirements Specification Development	35
4.1.4	Requirements Wording	36
4.1.5	Example System Interface Requirements	36
4.2	Requirements Traceability	38
4.2.1	Needs to Requirements Traceability Matrix (NRTM).....	38
4.2.2	Example NRTM.....	39
4.2.3	Forward Traceability.....	40
4.3	Developing a System Interface Specification Document from the Requirements.....	40
4.3.1	Compliance Statement	41
4.3.2	Dialog Descriptions	42
4.3.3	Generic Dialogs	43
4.3.4	Data Concept Definition: Dialogs, Messages, Data Frames, and Data Elements... ..	46
4.3.5	Tailoring Project Specific Data Concepts from the IEEE 1512 Standard	46
4.3.6	Select Required Data Concepts.....	47

Guide for Implementing IEEE 1512 Using a Systems Engineering Process
Contents

4.3.7	Example Tailored Incident Description Message	48
4.3.8	Example Tailored Request Information (RIN) Message	52
4.4	Handling Extension of the Standard.....	53
4.4.1	ASN.1 Module Approach	53
4.4.2	XML Schema Approach	53
4.5	System Interface Specification Requirements Traceability Matrix (RTM).....	54
4.6	Summary	58
5	Design Phase.....	59
5.1	Software Design Documentation and Standards.....	59
5.1.1	What is a Software Design Description (SDD)?.....	59
5.1.2	Purpose of the SDD	59
5.1.3	SDD Content and Organization	59
5.2	Design Process	61
5.3	System Interface Requirements to Software Design Description Traceability Matrix ..	62
6	Implementation Phase.....	64
6.1	Documenting the Software Engineering Environment	64
6.2	Technical Issues in System Interface Design and Implementation.....	64
6.2.1	Establishing a Communications Framework	64
6.2.2	Communications Dialogs Related Technologies	69
6.2.3	Integrating Legacy Systems.....	72
7	System Interface Test Phase	74
7.1	Test Documentation	74
7.1.1	Standards that Support Test Documentation.....	74
7.1.2	Tailored System Interface Compliance Testing Concept	75
7.2	Test Plan.....	77
7.3	Test Design Specifications	79
7.4	Test Case Specifications.....	80
7.4.1	Example Requirements to Test Case Traceability Matrix (RTCTM).....	81
7.4.2	Example Test Case Specification.....	81
7.5	Test Procedure Specifications	88
7.5.1	Example Test Procedure	89
7.6	Test Reports.....	90
7.7	Summary	90

1 Introduction

1.1 Purpose of this guide

IEEE Std 1512™, an Intelligent Transportation Systems (ITS) standard, deals with information exchanges between public safety centers and traffic management centers.

The purpose of the IEEE 1512 Implementation Guide is to provide guidelines for and assistance with the implementation process of IEEE Std 1512. This guide presents information and examples on how to facilitate the development of needs, requirements, and design content that can be used during the procurement, implementation, and verification phases of the system life cycle for projects that use IEEE Std 1512.

The guide has been developed to provide instruction to a knowledgeable reader on how to plan a deployment of an IEEE 1512 system interface through the use of the systems engineering process. It also provides guidance to help clarify the acquisition process.

The guide provides the following information:

- a) IEEE Std 1512 implementation process overview
- b) Development of operational scenarios leading to well-defined user needs
- c) Development of well-formed requirements traceable to user needs
- d) Preparation of a system interface specification traceable to the requirements
- e) How to deal with interoperability issues, for example, legacy systems
- f) Creation of test plans, test cases, and procedures
- g) Verification and validation (V&V)

The intended audience is composed of deployers (system planners, system operators, project managers, requirements engineers, specification developers, designers, programmers, system integrators, and testers) whose goal is to incorporate IEEE Std 1512 at their respective public safety center or transportation management center.

As part of the deployment process, it is necessary to develop user needs, requirements, traceability of the requirements, and applicable dialogs to facilitate exchange of information. These elements are not included in IEEE Std 1512; however, they are essential for the successful implementation of this standard.

1.2 IEEE 1512 Overview

IEEE Std 1512 was developed to standardize data communications between public safety and traffic management centers. IEEE Std 1512 is not a single standard, but a series of five standards. Here is an overview of each volume:

- **IEEE Std 1512TM-2006, IEEE Standard for Common Incident Management Message Sets for Use by Emergency Management Centers.** This document is the foundation of the series and includes general introductory material about IEEE Std 1512 and a message set that focuses on the exchange of information about traffic incidents.
- **IEEE Std 1512.1TM-2003, IEEE Standard for Traffic Incident Management Message Sets for Use by Emergency Management Centers.** This volume focuses on the exchange of information about traffic and public safety agency resources used during traffic incident response. It specifically addresses the assignment of multi-jurisdictional resources and covers how to request a traffic management center to perform functions with their roadside devices.
- **IEEE Std 1512.2TM-2004, Public Safety Incident Management Message Sets for Use by Emergency Management Centers.** This volume covers the exchange of information necessary to support traffic incident response across public safety agencies, including law enforcement, emergency medical services (EMS), and fire and rescue.
- **IEEE Std 1512.3TM-2006, Hazardous Material (HAZMAT) Incident Management Message Sets for Use by Emergency Management Centers.** This volume covers the exchange of information where hazardous materials have been released on or near a roadway. It deals specifically with the information content necessary to describe the type of cargo and placards.
- **IEEE P1512.4TM (Draft), Common Traffic Incident Management Sets for Use in Entities External to Centers.** Currently under development, this volume specifies message sets which support the exchange of information between a mobile asset and the center.

Familiarity with IEEE Std 1512 is a prerequisite for understanding this guide. This standard can be purchased from IEEE. An excellent primer is the IEEE publication, *Guide to the IEEE 1512 Family of Standards*.

1.3 IEEE 1512 System Interfaces

The goal of IEEE Std 1512 is to support the development of system interfaces between centers for real time traffic incident information exchange.

IEEE Std 610TM, IEEE Standard Glossary of Software Engineering Terminology, defines an interface as a shared boundary across which information is passed. The specification of this boundary, the system interface, is the focus of this guide.

The purpose of an interface standard, such as IEEE Std 1512, is to achieve interoperability. *Interoperability* is defined as the ability of two or more systems or components to exchange information and to use the information that has been exchanged [IEEE Std 610, p. 42].

IEEE Std 1512 contains a definition of the system interface concept. As a system interface standard, IEEE Std 1512 provides a detailed specification for:

- Data content of information
- Format and sequence of information exchanges (inputs and outputs)

The system interface concept, as described in IEEE Std 1512, is shown in Figure 1.

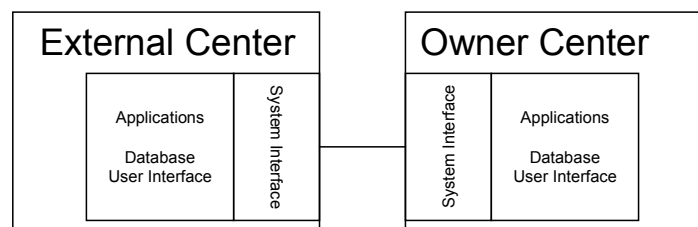


Figure 1. System Interface Concept

This diagram sets an environmental context for the information presented in this guide. The diagram shows an external center and an owner center joined through a common system interface. Also, a logical view of the system components (i.e., system interface, user interface, and database) is shown in this diagram.

An *owner center* is a center, such as a TMC, that provides information developed or stored within it (e.g., event information) to another center. The owner center can also have direct control of field devices. In the context of the most common dialogs used by this standard, the owner center publishes information or provides responses to a request from an external center. An external center is a unique combination of an organization name and center name that uses the C2C services provided by another center. Typically, each center has its own applications.

The system interface is a separate system entity from the applications, databases, servers, user interface, and in fact, any hardware and software of the system. When the term *system interface* is used, it refers only to the sequence of information exchanges (called a *dialog*) and content.

For example, in Chapter 7, the compliance of the system interface specification is being tested, not the software implementation. Database and user interface software may be used to test the system interface. The test is for compliance with the specification to ensure that the dialogs and data content of message exchanges are implemented per a specification (which will be based upon the standard).

1.4 System Interface Life Cycle Process

This section will provide discussion of a high-level process to be used in the development of a system interface. This process, called *the system interface life cycle process*, describes how to define, build, and test a system interface. Central to the definition of the system interface life cycle process is the use of the Systems Engineering Process, which will be discussed in the following sections.

1.4.1 The Systems Engineering Process (SEP)

A Systems Engineering Process (SEP) is a structured approach to technical management, system design, product realization, and technical evaluation. The SEP propagates through a series of steps which lead to a verifiable system solution. The VEE diagram, shown in Figure 2, is a representation of the life cycle process as it relates to systems engineering. Scoping, design, and specification development, progress down the left side of the VEE. Construction (of hardware and software) sits at the bottom of the VEE, while testing and validation proceed up the right side.

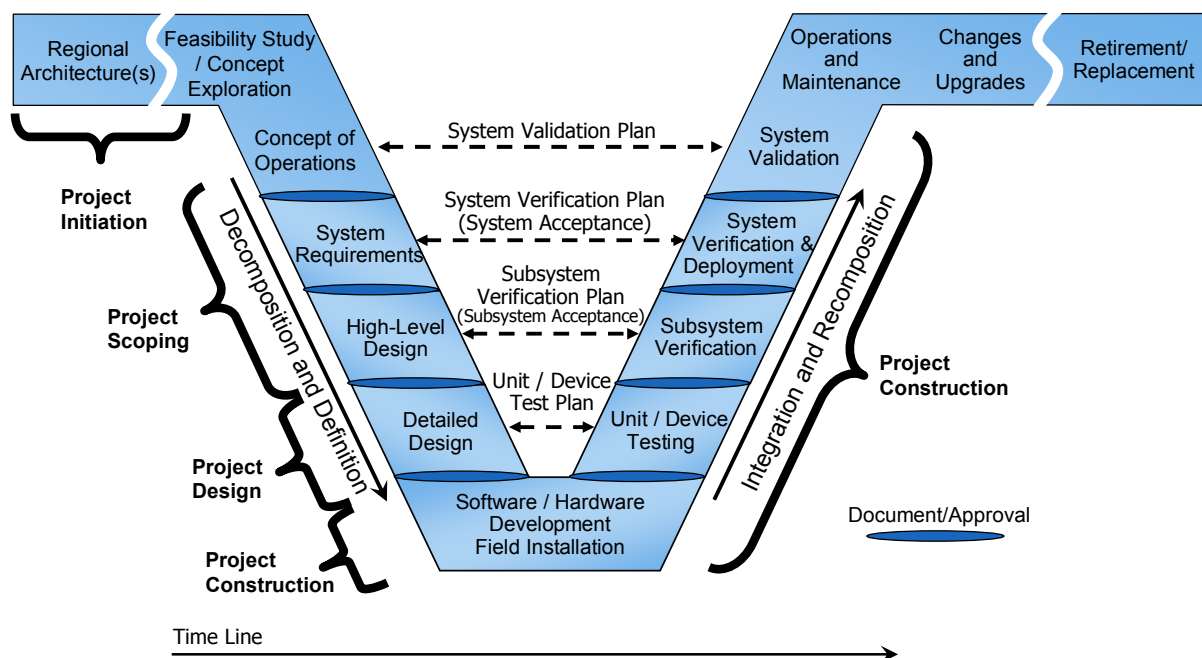


Figure 2. Integrating Systems Engineering into the Project Development Process

An excellent primer on the use of the Systems Engineering Process for ITS systems is *Systems Engineering for Intelligent Transportation Systems—An Introduction for Transportation Professionals*, which has been developed by the U.S. Department of Transportation to provide a high-level overview of the subject.

1.4.2 Standards That Support the Systems Engineering Process

Many standards provide a more in-depth treatment of the SEP and the stages defined. These standards, listed in Table 1, are referenced and used throughout this guide.

Table 1. Standards that Support the Systems Engineering Process

SEP Activity	Document Name
Define the SEP	INCOSE Systems Engineering Handbook v3.1, 2007
Define the SEP	IEEE Std 1220 TM -2005, IEEE Standard for Application and Management of the Systems Engineering Process
Software Life Cycle	IEEE Std 1074 TM -2006, IEEE Standard for Developing Software Life Cycle Processes
Glossary of Terms	IEEE Std 610.12 TM -1990, IEEE Standard Glossary of Software Engineering Terminology
Project Management Plan	IEEE Std 1058 TM -1998, IEEE Standard for Software Project Management Plans
Quality Assurance Plan	IEEE Std 730 TM -2002, IEEE Standard for Software Quality Assurance Plans
Configuration Management Plan	IEEE Std 828 TM -2005, IEEE Standard for Software Configuration Management Plans
Concept of Operations	ANSI/AIAA G-043-1992, Guide for the Preparation of Operational Concept Documents
Concept of Operations	IEEE Std 1362 TM -1998, IEEE Guide for Information Technology—System Definition—Concept of Operations (ConOps) Document
Software Requirements	IEEE Std 1233 TM -1998, IEEE Guide for Developing System Requirements Specifications
Software Requirements	IEEE Std 830 TM -1998, IEEE Recommended Practice for Software Requirements Specifications
Design	IEEE 1016 TM -1998, IEEE Recommended Practice for Software Design Descriptions
Implementation	IEEE Std 1028 TM -1997, IEEE Standard for Software Reviews
Testing, Verification and Validation	IEEE Std 829 TM -1998, IEEE Standard for Software Test Documentation
Testing, Verification and Validation	IEEE Std 1012 TM -2004, IEEE Standard for Software Verification and Validation

1.5 Tailored System Interface Life Cycle Process

Thus far the concept of a system interface has been presented. This section presents a tailoring of the system engineering life cycle process to focus on the definition and deployment of the system interface. The resulting system interface life cycle process will:

- a) Illustrate how to make acquisitions of interfaces easier and traceable to needs
- b) Illustrate how to set up a requirements-based verification and validation program
- c) Inform the integrator as to how and when verification and validation will be performed

System interface life cycle phases include:

- **Project Initiation Phase.** During this phase the project is planned, and documents created to show project milestones and the expected process that will be followed.
- **Concept Definition Phase.** This phase documents how an organization and its people will use the system. Organizational and user needs are developed.
- **Requirements Elicitation Phase.** Based on the user needs from the previous phase, a description of what the system interface shall do is developed.
- **Design Phase.** Based on the requirements from the previous phase, system software and hardware design can be developed
- **Implementation Phase.** Based on the system design, software and hardware products are purchased or fabricated.
- **System Interface Test Phase.** Based on the requirements, a test plan shall be developed and executed to verify that the system complies with the system interface requirements.

Defining and deploying a system interface requires a team. Figure 3 illustrates team member roles in relation to the phases of the system interface life cycle. Note that a project could have fewer persons on the team than shown in the figure because individuals could possibly have multiple roles during the development process. Conversely, another project in a different jurisdiction might require additional roles.

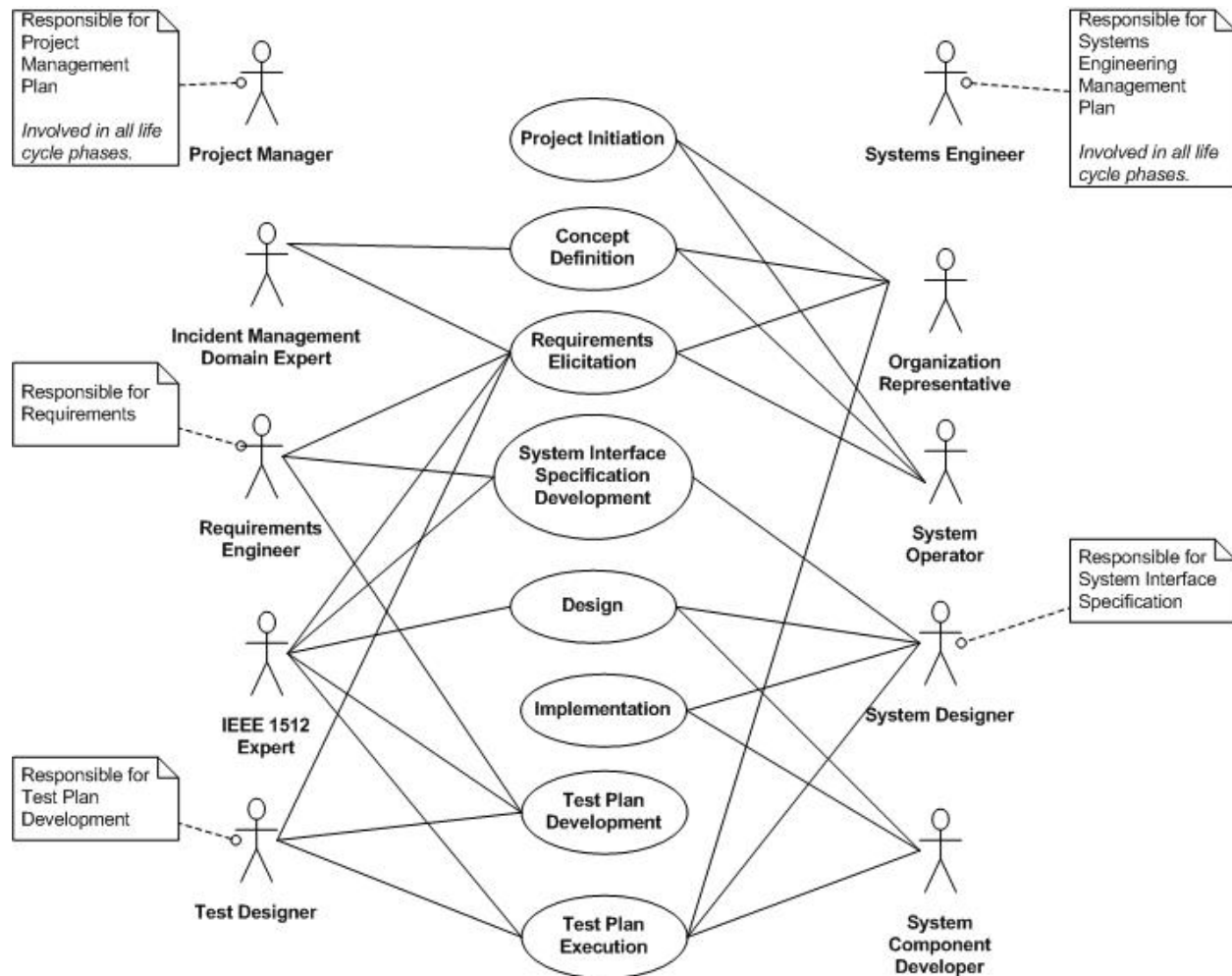


Figure 3. System Interface Development Life Cycle Roles and Responsibilities

A description of each team member role follows:

- **Organization Representative.** Represents the organization procuring the system, typically a public agency, and is knowledgeable of the policies, goals, and objectives related to the system.
- **System Operator.** This role represents the users who will interact with the system. This role includes system operators, users of information (reports, for example) generated by the system, and system maintainers.
- **Project Manager.** Responsible for the overall day-to-day management for the scope, budget, and schedule of the project.
- **System Engineer.** Responsible for the management of the system engineering life cycle and life cycle products, documents, and artifacts of the project.

- **Incident Management Domain Expert.** Ensures that the project needs and requirements are based on valid incident management principles, practices, and experience.
- **IEEE 1512 Expert.** Ensures that the project requirements, specifications, and design are based on a thorough understanding of IEEE Std 1512 principles, industry practices, and experience.
- **Requirements Engineer.** Responsible for gathering, analysis, and verification of the system interface requirements.
- **System Designer.** Responsible for translating the requirements into a solution (a specification or design).
- **System Component Developer.** Represents the persons responsible for construction and integration of the system elements into a working system capable of being tested, based on the system design. Persons responsible include programmers, database administrators, hardware installers, and their supervisors.
- **Test Designer.** This role is responsible for the development and execution of the system interface test plans.

Table 2 summarizes the roles, responsibilities, deliverables, and the major activities related to the system interface life cycle and includes the following information:

- **Project Phase.** The system interface life cycle phase as it relates to a project with the objective of defining, implementing, and testing the system interface.
- **Project Documents and Artifacts.** Deliverable documents, products, or portions.
- **Suggested Project Document Reference Guide.** The name of a standard or reference guide for the content of the project documents and artifacts.
- **Roles.** Team roles that are involved in the phase to develop the deliverable documents or product.
- **Responsibility.** Role responsible for development of the deliverable document or product.
- **Chapter in this Guide.** Chapter in this guide to read for more information about this project phase.

Guide for Implementing IEEE 1512 Using a Systems Engineering Process

Introduction

Table 2. System Interface Development Life Cycle Roles, Responsibilities, and Artifacts

Project Phase	Project Documents & Artifacts	Project Document Reference	Roles	Responsibility	Chapter in Guide
1. Project Initiation	• Project Management Plan • Systems Engineering Management Plan • Configuration Management Plan • Verification and Validation Plan • Risk Management Plan	• IEEE Std 1058, IEEE Standard for Software Project Management Plans • IEEE Std 1220, IEEE Standard for Application and Management of the Systems Engineering Process • IEEE Std 828, IEEE Standard for Software Configuration Management Plans • IEEE Std 1012, IEEE Standard for Software Verification & Validation	• Organization Representative • Project Manager • Systems Engineer • System Operator	• Project Manager • Systems Engineer	Chapter 2
2. Concept Definition	• Concept of Operations • User Needs	• IEEE Std 1362, IEEE Guide for Information Technology—System Definition—Concept of Operations (ConOps) Document • ANSI/AIAA G-043, Guide for the Preparation of Operational Concept Documents	• Organization Representative • System Operators • Incident Management Domain Expert • Project Manager • Systems Engineer	• Project Manager • Systems Engineer	Chapter 3
3. Requirements Elicitation	• System Interface Requirements • Needs to Requirements Traceability Matrix • Requirements Traceability Matrix	• IEEE Std 1233, IEEE Guide for Developing System Requirements Specifications • IEEE Std 830, IEEE Recommended Practice for Software Requirements Specifications	• Organization Representative • System Operators • Incident Management Domain Expert • IEEE 1512 Expert • Requirements Engineer • Project Manager • Systems Engineer • Test Designer	• Requirements Engineer	Chapter 4

Guide for Implementing IEEE 1512 Using a Systems Engineering Process

Introduction

Table 2. System Interface Development Life Cycle Roles, Responsibilities, and Artifacts *(continued)*

Project Phase	Project Documents & Artifacts	Project Document Reference	Roles	Responsibility	Chapter in Guide
4. Design	• System Interface Design Description • System Interface Requirements to Design Traceability Matrix • Design Reviews and Walkthroughs	• IEEE Std 1016, IEEE Recommended Practice for Software Design Descriptions	• IEEE 1512 Expert • System Designer • System Component Developer • Project Manager • Systems Engineer	• System Designer	Chapter 5
5. Implementation	• Code Development Environment • Code Reviews and Walkthrough	• IEEE Std 1028, IEEE Standard for Software Reviews and Audits	• System Designer • System Component Developer • Project Manager • Systems Engineer	• System Component Developer	Chapter 6
6. System Interface Test	• Test Plan • Test Design Specifications • Test Cases • Test Procedures • Test Reports • Requirements to Test Case Traceability Matrix	• IEEE Std 829, IEEE Standard for Software Test Documentation	• Organization Representative • IEEE 1512 Expert • Test Designer • System Component Developer • Project Manager • Systems Engineer	• System Designer • Test Designer	Chapter 7

1.6 System Acquisition

The purpose of the system acquisition process is to procure a system that possesses a potential for high quality and is well-documented; i.e., documentation that is complete, correct, and consistent. Table 2 presents a list of project documents required to ensure a defined level of quality from procured software products. Which documents are included in a procurement package depends on which phase of development the system is in. Procurements (one or more) could occur at various points during the system development life cycle.

The key results of the system acquisition process as outlined in this guide will be systems that:

- a) Are based on appropriate requirements
- b) Are traceable to needs
- c) Are well-documented and based on standards
- d) Support verification and validation activities at each stage of the development process
- e) Specify dialogs, messages, and contents for the system interface

The approach for determining which system documents to include in a procurement package should be independent of the method of procurement (for example, design-build or system integrator). What is important is that regardless of which organization is responsible for a particular project phase—whether public agency or contractor—that the resulting system documents properly reference, use, and trace to documents and artifacts from previous project phases, and that the documents and artifacts created in a given phase are usable and traceable from project documents and artifacts developed in a future project phase. This approach will maintain traceability and support verification and validation at each project phase.

When a public agency or contractor develops the system documents, they shall maintain the tailored systems engineering process used in this guide; for example, the system documents shall keep the traceability intact, and they shall allow verification and validation at each phase in the project.

The suggested approach for using the information in Table 2 depends on the current project phase. The documents should be developed up to that phase in the procurement package, and then specify that the contractor provide the system interface and project level documents as required deliverables for the phase (or phases) being procured. The procurement package needs to be specific and should describe the content of the documents required for the phases being procured, and what relationships with any previous phase documents need to be documented and maintained, for example, traceability matrices.

For example, if a phase of development after the requirements phase is being procured, and the design, implementation, and test phases are also being procured, then it can be assumed that the following project documents and artifacts listed already exist:

- a) Project Management Plan, Systems Engineering Management Plan, Configuration Management Plan, Verification and Validation Plan, and Risk Management Plan
- b) Concept of Operation (ConOps) and User Needs

- c) System Interface Requirements and Needs to Requirements Traceability Matrix
- d) System Interface Specification and Requirement Traceability Matrix

The following brief list provides the contents of a procurement package:

- All documents listed previously in items 1 through 4
- Any other project documents and artifacts to be provided by the contractor
- Detailed information on the roles that the agency expects to play during implementation
- A list the documents and descriptions required in each of the design, implementation, and test phases
- Description of the relationships between the documents that are required to support Verification and Validation activities

1.7 Document Organization

This section provides information on the content of the remaining chapters of the guide.

- **Chapter 2: Project Initiation Phase.** This chapter provides an overview of the project planning phase and identifies key documents that will allow an organization to manage the system life cycle from the perspectives of project management, systems engineering, and verification and validation.
- **Chapter 3: Concept Definition Phase.** This chapter provides an overview of the Concept Definition Phase and a key document, the Concept of Operations (ConOps). Guidance on creating and refinement of operational scenario to support identification of one or more user needs is also included in this chapter.
- **Chapter 4: Requirements Elicitation Phase.** This chapter provides information on developing IEEE 1512-based System Interface Requirements. This chapter also provides guidance on system requirements identification, attributes of good requirements, and assistance with creating well-formed requirements that satisfy user needs.
- **Chapter 5: Design Phase.** This chapter describes what should be in a software design description document and explores how to develop elements of a software design that will fulfill the system interface requirements.
- **Chapter 6: Implementation Phase.** This chapter presents an approach for tailoring IEEE Std 1512 based on system interface requirements, and discusses technical issues that affect IEEE 1512 deployments.
- **Chapter 7: System Interface Test Phase.** This chapter provides guidance on system interface testing. This chapter also includes information on development of test plans, test specifications, and test reports.

2 Project Initiation Phase

The project initiation phase is a planning phase. Project planning lays out the activities, resources, budget, and timeline for the project. This project planning effort, which begins early in the project life cycle, results in the creation of two major plans, the Project Management Plan (PMP) and the Systems Engineering Management Plan (SEMP). Both of these documents should be written in such a way that a newcomer to the project team can understand the scope of the project, the responsibilities of the major players, the staffing, the schedule and budget, and the processes that will govern the project. The three additional plans that will also be discussed in this section—Risk Management Plan, Configuration Management Plan, and Verification and Validation Plan—will either become appendices to the PMP or SEMP or will stand on their own, depending on the size and complexity of the project [SE Guide for ITS, p. 82].

2.1 Project Management Plan

The Project Management Plan (PMP) documents how the project will be managed and controlled from a programmatic standpoint. It identifies the detailed work plans for both administrative and technical tasks. For each project task, the PMP documents what is to be done, by whom, with what funds, when, how (processes to be used), and dependencies.

The PMP should include the purpose and overview of the project, task descriptions, resources and budget allocated to each task, deliverables, and project schedule. It should also include a budget plan that estimates annual / monthly costs and identifies the source of funds, a project organization, as well as roles and responsibilities relative to project execution. It is helpful to create a table that identifies the owner or lead for each process and document as well as those organizations that provide a supporting role [SE Guide for ITS, p. 83].

2.2 Systems Engineering Management Plan

The Systems Engineering Management Plan (SEMP) is the top-level plan for managing the systems engineering effort to produce a final operational system from initial requirements. Just the way a PMP defines how the overall project will be executed, the SEMP defines how the engineering portion of the project will be executed and controlled. It describes how the efforts of system designers, test engineers, and other engineering and technical disciplines will be integrated, monitored, and controlled during the complete life cycle. For a small project, the SEMP can be included as part of the PMP document, but for any project of greater size or complexity, a separate document is recommended [SE Guide for ITS, p. 83].

The information contained within a SEMP can be organized in different ways, but in general, it should include sections for introduction (including system description, top-level schedule, and relevant documents), technical planning and control, systems engineering processes tailored specifically for the project, and plans for coordinating the efforts of multiple engineering disciplines to accomplish the project tasks [SE Guide for ITS, p. 83]. A key recommendation is to make sure that the SEMP and the PMP are consistent.

2.3 Configuration Management Plan

The Configuration Management Plan documents processes and outputs of configuration management activities. Configuration management may be addressed in the SEMP. The purpose of configuration management is to keep the physical implementation of a product consistent with the documentation that describes how to build it and what it is supposed to do. By keeping the product and all its associated documentation synchronized throughout the development cycle, manufacturers reach the production stage of a product life cycle ready to begin mass-producing it.

This involves having a document (or series of documents) that describes the components that make up any version of a product (hardware or software), along with the requirements that the version of the product is intended to satisfy. The documentation includes diagrams or descriptions of the product's design, its operating parameters, and anything else required to ensure that interchangeable copies of the product can be produced at any time [CM Guide for ITS, p. 5].

The configuration management process consists of five major activities:

- **Configuration Management Planning.** Planning for what needs to be controlled to completely define the configuration of the system, how to change a controlled configuration, how to keep track of those changes, and how to verify that the configuration management processes are working
- **Configuration Identification.** Identifying the functional and physical characteristics of a configuration item
- **Configuration Change Management.** Controlling change to those characteristics
- **Configuration Status Accounting.** Keeping track of the status of changes to the configuration items (e.g., proposed, approved, or implemented)
- **Configuration Auditing.** Verifying that configuration management procedures are being followed as well as the consistency of documentation against the configuration item [SE Guide for ITS, p. 90]

2.4 Verification and Validation Plan

Verification and Validation (V&V) may be addressed in the SEMP. The V&V plan describes the process and documentation used to determine whether the requirements for a system or component are complete, consistent, and correct, whether the products of each development phase fulfill the requirements or conditions imposed by the previous phase, and whether the final system or component complies with specified requirements [IEEE Std 610, p. 85]. The purpose of V&V is to help the development team build quality (satisfies user needs and ensures that requirements are fulfilled throughout the system life cycle) into the system interface life cycle. V&V shall be accomplished at each phase of the life cycle. This allows identification and correction of defects early in the process when it is most cost-effective to do so. The process of V&V is facilitated by requirements traceability analysis, an important tool used to perform

V&V. Other tools include reviews, walkthroughs, and inspections at each stage of the life cycle: Concept Definition, Requirements Elicitation, Design, Implementation, and Testing.

2.5 Risk Management Plan

The Risk Management Plan documents the processes and outputs of risk management activities. Risk management is the identification and control of risks during all phases of the project life cycle. Risk management may be addressed in the PMP. The goal of risk management is to identify potential problems before they occur, plan for their occurrence, and monitor the system development so that early action can be taken if the risk occurs [SE Guide for ITS, p. 87].

Risk management is composed of the following general steps:

- a) **Risk Identification.** The objective of the risk identification step is to identify the key risks to project success at the beginning of the project. This will require that project managers, stakeholders, and subject experts brainstorm about where the risks lie. They should take a look at all potential risks, from initial development all the way out through operations and maintenance and eventually, retirement of the system. Areas of risk that might affect a project include technical, institutional, schedule, and funding [SE Guide for ITS, p. 87].
- b) **Risk Analysis and Prioritization.** After the risks have been identified, the next step in the process is to analyze and prioritize them. This step involves a determination of the impact, should the risk occur, and the probability of its occurrence. Risks typically fall into one or more of the following areas: technical (risks affecting the quality of the resulting system), schedule (risks that cause schedule slippage), and cost (risks that cause actual cost to exceed allocated budget). Depending on the nature of the project and the nature of the risks, this assessment of severity can be qualitative (e.g., high, medium, or low) or quantitative (e.g., weeks of schedule slippage or amount of cost overrun) [SE Guide for ITS, p. 88].
- c) **Risk Mitigation.** The objective of risk mitigation is to identify and evaluate alternatives for handling the risks identified previously. From a project management standpoint, there are several ways to address risks: avoid the risk altogether (e.g., change in requirements or design), take actions to reduce the likelihood or severity of the risk (e.g., prototype a new technology or develop and deliver software incrementally), accept the risk and do nothing, which is often the approach for risks with low impact and low probability of occurrence. A risk mitigation strategy that will describe the steps to be taken to lessen the severity of the risk, and the probability of its occurrence should be created for each risk [SE Guide for ITS, p. 89].
- d) **Risk Monitoring.** Risks should be monitored throughout the life cycle to determine whether the mitigation steps are actually lessening the severity or probability of each risk. It is also possible that the nature of the risk has changed (e.g., the due date for delivery of the software code was extended because another project was delayed). One way to monitor risks is to develop metrics that will give an indication that a risk is occurring. These metrics (e.g., cost versus schedule to identify if cost overruns are beginning to occur) should be easy to track and should provide some

signal of a potential problem. Risk status should be reviewed periodically during progress meetings and other reviews [SE Guide for ITS, p. 90].

3 Concept Definition Phase

To define a system interface that creates value for the agencies looking to implement IEEE Std 1512, it is necessary to understand the internal operations (people and processes) of the organization, what information needs to be exchanged, and for what purpose.

This chapter includes the development of the key Concept of Operations elements that is necessary to successfully develop a system interface definition.

3.1 Documenting User Needs and Developing a Concept of Operations (ConOps)

A Concept of Operations (ConOps) is a document, written from the user and organization's point of view, that clearly defines the situation or problem scope, identifies user needs, and the operational context for the information exchanges that the system interface will support. The ConOps, therefore, should be developed with participation from all users that benefit from or are affected by the system.

A team should be brought together to develop the ConOps. With a focus on the system interface, team members should have expertise in the following: center management and operations, incident management, and user needs assessment and definition.

Describing how to write a ConOps for a system or a project is beyond the scope of this document. Hence, this guide will focus on the ConOps elements (problem scope, identification of users, operational scenarios, and user needs) that will support successful definition of a system interface. A general ConOps developed for a project might not include all of the ConOps elements needed to successfully define the system interface requirements that will be used to tailor IEEE 1512 requirements. In this case, a team that can develop the missing ConOps elements should be brought together.

3.2 Standards that Support ConOps Development

Two standards to guide ConOps development are currently available. They are:

- IEEE Std 1362, IEEE Guide for Information Technology System Definition—Concept of Operations (ConOps) Document
- ANSI/AIAA G-043-1992, Guide for the Preparation of Operational Concept Documents

Both documents agree that a ConOps is written from a user and organizational perspective.

IEEE Std 1362 for ConOps states:

A ConOps is a user-oriented document that describes system characteristics for a proposed system from the users' viewpoint. The ConOps document is used to communicate overall quantitative and qualitative system characteristics to the user, buyer, developer, and other organizational elements (for example, training, facilities, staffing, and maintenance) [IEEE Std 1362, abstract].

A suggested IEEE 1362 ConOps outline is as follows:

- a) Scope
- b) Referenced Documents
- c) Current System or Situation
- d) Justification for and Nature of Changes
- e) Concepts for the Proposed System
- f) Operational Scenarios
- g) Summary of Impacts
- h) Analysis of the Proposed System

The ANSI G-043 standard says that the purposes of an Operational Concept Document are to:

- “1) Describe the system characteristics from an operational perspective.
 - 2) Facilitate understanding of the overall system goals with users, buyers, implementers, architects, testers, and managers.
 - 3) Form an overall basis for long-range operations, planning and provide guidance for development of subsequent system definition documents such as system and interface specifications.
 - 4) Describe the user organization and mission from an integrated user / system point of view.”
- [ANSI G-043, p. 1]

A suggested ANSI G-43 outline is as follows:

- a) Scope
- b) Referenced documents
- c) Operations
- d) Operational Needs
- e) System Overview
- f) Operational Environment
- g) Support Environment
- h) System Operational Scenarios

3.3 ConOps Content for Systems Interfaces

The previous two outlines show several key ConOps elements that must be documented to successfully define a system interface:

- a) Current Situation/Problem Scope
- b) Identification of User Classes and Roles

- c) Operational Scenarios
- d) User Needs

The remainder of this chapter will discuss these four key ConOps elements.

3.3.1 Current Situation/Problem Scope

This is a statement of the problem or situation. This section of the ConOps provides an introduction to the problem domain. It enables readers to understand the rationale behind the changes and/or improvements desired—the motivation behind the system. The motivation might be one derived from a stated organizational strategy, tactic, goal, or objective. For example, improved coordination with other organizations during emergency situations could be an agency goal. Furthermore, the agency might have identified that improved coordination with other emergency responders depends on sharing of incident information.

3.3.2 Identification of User Classes and Roles

The ConOps includes a description of who the system's users are and who is affected by or benefits from the system. Users might include the Operations Department (users that operate the system), Maintenance Department (users that maintain the system), and public safety agencies (users that provide information and resource requests). The techniques shown in this section represent one way of assessing users and the operational capabilities desired in a system.

3.3.2.1 Identification of User Classes

Table 3 provides a sample list of user classes that might be documented in the ConOps. This list is the starting point for the examples developed in this guide. The user class table shows the following information:

- **User Class ID.** A unique identifier for each user class
- **User Class.** Distinguishes the ways in which users interact with the system. Factors that distinguish a user class include common responsibilities and work activities [IEEE Std 1362, p. 8]
- **User Class Description.** Descriptive text outlining the overall operational or maintenance responsibilities of the user

Table 3. Example of User Classes and Descriptions

User Class ID	User Class	User Class Description
1	Motorist	<i>Motorist</i> is a term used to represent the traveling public.
2	Traffic Management Center (TMC) Operator	TMC operator controls ITS devices, monitors the status of roadways, and provides information about roadway status to other organizations and travelers. Depending on local TMC operations, TMC operator may dispatch first response also.
3	Emergency Management (EM) Operator	EM Center Operators control the dispatch function for responders and coordinate incident verification with TMCs.
4	EM Responder	Emergency personnel operate emergency vehicles that respond to incidents and coordinate incident site logistics.
5	Maintenance Section Operator	Maintenance and construction personnel operate heavy machinery when performing road maintenance or when clearing an incident or providing repairs to the transportation network due to an incident.
6	Regional Transit Operator	Transit operators manage fleets of transit vehicles throughout the region and operate evacuation buses for residents during a natural disaster or hazardous materials incidents.

3.3.2.2 Documenting User Classes and Roles

Table 4 shows information that is useful to capture in documenting user class roles. This information includes:

- **User Class Identifier and User Class.** Used to cross reference with the initial list of system users shown in Table 3.
- **Role.** A user plays one (or possibly more) roles in daily operations of the center that will exchange information across the system interface. Other users might not have a daily role but might fill-in during special situations. A user playing a specific role creates a need.
- **Desired Operational Capability.** This is the statement of a desired operational capability. It could be a description of what operational or user need is lacking, or system features that are missing.
- **Expected Benefits.** Define the desired outcome the user wants from the system that is of value to the user in a particular role, for example, improved travel time, incident response time, or other valued operational improvement.
- **Performance Measures.** These measures help to quantify the organization/user expectations of the systems. Ideally, performance measures can be used to determine the value of the system or service to the user, vis-à-vis cost to deploy.

Assessment of user classes and roles (per the examples in Tables 3 and 4) and development of the operational scenarios and narratives (which are discussed in the next section) will usually occur in an iterative fashion until a clear set of operational capabilities of the system and sequence of operational or maintenance activities are defined.

Guide for Implementing IEEE 1512 using a Systems Engineering Process

Concept Definition Phase

Table 4. Example Analysis of User Class Roles

User Class ID	User Class	Role	Desired Operational Capability	Expected Benefit	Performance Measure
1	Motorist	The traveling public	To plan timely routes.	Improved travel time.	Travel time between an origin and destination, by time of day.
			To be notified of incident and road closures.	Improved safety <i>en route</i> to destination.	Number of incidents reported.
2	TMC Operator	The TMC operator controls TMC ITS devices, monitors the status of roadways and provides information about such to motorists.	To provide warnings and alerts to the traveling public.	More timely provision of information, warnings and alerts to Motorists so that they can change travel plans.	Timeliness of warnings and alerts. Geographic scope of coverage (given the dimensions) of alerts and warnings.
			To provide incident information to emergency responders.	More accurate information to emergency responders so they dispatch properly to incidents.	Response times of incident responders
3	EM Operator	The EM Operator controls the dispatch function for responders and coordinates incident verification with TMC operators.	To provide immediate response to incidents.	More timely provision of incident responders at the incident site.	Response times of incident responders.
			To provide the proper response to incidents.	More effective communication and coordination between EM providers and TMCs.	Response times of incident responders.
			To coordinate traffic operations with TMCs for incidents.	More effective communication and coordination between EM providers and TMCs.	Response times of incident responders. Travel delays associated with the incident.
			To provide timely response to incidents, especially EMS and HAZMAT. To coordinate traffic operations with TMCs for large scale incidents and with other EM Centers for evacuation plans.	More timely provision of incident responders to incident site for EMS or HAZMAT, including clean-up. More timely provision of secondary incident responders to incident site for evacuations.	Response times of incident responders. Number of fatalities associated with HAZMAT or EMS providers. Total time associated with evacuations. Number of fatalities and injuries during evacuations.

Guide for Implementing IEEE 1512 using a Systems Engineering Process

Concept Definition Phase

Table 4. Example Analysis of User Class Roles (*continued*)

User Class ID	User Class	Role	Desired Operational Capability	Expected Benefit	Performance Measure
4	EM Responder	Emergency personnel operate emergency vehicles that respond to incidents and coordinate incident site logistics.	To provide accurate information to EM Centers for dispatch of secondary responders. To effectively coordinate the incident at the incident site with traffic, maintenance and other emergency providers.	More timely provision of incident information to secondary responders. Improved travel times for travelers with effective mitigation procedures at the incident (including traffic and maintenance coordination).	Number of fatalities at incident site. Time delay associated with incident. Time between initial responder notification and clear time at incident site.
		Emergency Personnel operate secondary emergency vehicles that respond to incidents (HAZMAT) and coordinate incident site logistics.	To provide accurate information to EM Centers for proper mitigation procedures for HAZMAT. To effectively coordinate the incident at the incident site with traffic, maintenance and other emergency providers for evacuation in surrounding areas.	More timely provision of incident information to the public. Improved travel times for travelers with effective mitigation procedures at the incident (including traffic and maintenance coordination).	Number of fatalities at incident site and in the surrounding areas potentially affected by a HAZMAT. Time to evacuate surrounding areas. Total delay associated with incident.
5	Maintenance Section Operator	Maintenance and construction personnel operate heavy machinery when performing road maintenance or when clearing an incident or providing repairs to the transportation network due to an incident.	To receive accurate and timely information about services needed to clear an incident and about the status of the transportation network and requirements for maintenance or reconstruction (to bridges, roads, guard rail, etc.).	Shorter lag times in incident removal. More efficient construction and maintenance activities. Improved travel times and reduced delays for construction and maintenance activities.	Maintenance response time. Time between initial maintenance notification and clear time at incident site. Total delay associated with construction or maintenance of roadways.
6	Regional Transit Operator	Transit operators operate transit fleets during normal operations and incidents.	To improve efficiency of transit routes by avoiding incidents and traffic.	More timely provision of transit routes and adhering to transit schedules.	Schedule adherence when there is an incident.
		Transit operators operate transit fleets during a natural disaster or HAZMAT incident.	To provide timely response to transit needs during evacuation.	Faster and more coordinated evacuation and reentry of residents.	Time required to enact an emergency evacuation plan.

3.3.3 Operational Scenarios

Why focus on operations? Because system users do not always view operations from the same perspective, nor do users always agree upon operations. In other words, users involved in the same operational activities will not always agree on what the system should do or how the system should work. For example, operations and maintenance users will view an incident from a different perspective. Operations staff such as first responders will focus on getting to the incident scene as quickly as possible, treating victims and traffic control, while maintenance users will focus on ensuring that response equipment is working properly and on repairing the roadway infrastructure. Yet how a system works for all of its users during all defined modes of operations and maintenance is a key determinant of success for system projects. Different and potentially conflicting views of system use will result in a system that only partially meets the user's expectations.

Operational scenarios describe operational activities, user interactions with the system, and help to identify one or more user needs. For example, a scenario might include the procedures on how public safety agencies make requests for action or how maintenance requests are monitored and made. Operational scenarios are used in the concept of operations to help visualize what is needed and to establish intent. User needs identified in one or more operational scenarios will then be written to capture the full intent. There are many ways that these scenarios can be documented. The following examples provide one approach.

3.3.3.1 Operational Scenarios Example

This section examines the development of operational scenarios by showing an initial draft and a refined final draft to illustrate that multiple iterations and review with users and experts is necessary to arrive at a concise, nonoverlapping, complete, and consistent set of user needs agreed upon by all user classes. Moreover, describing the operational scenarios will help resolve conflicting information provided by users and provide for missing or incomplete descriptions provided by users.

First a use case diagram is developed that shows all system users and documents which users participate in which operational activities. An example of a use case diagram is shown in Figure 4. This is the initial draft. The use case diagram shows a graphical representation of user roles, written as a stick figure, and what use cases (defined in the next paragraph) a user role participates in. A use case is a description of a sequence of actions that a system performs that yields an observable result of value to a user role [UML User Guide, p. 468]. A single use case is indicated as an oval. The use case diagram captures the relationship of user roles and operational capabilities desired in a system. The use case now becomes a single point of view among several user roles.

To develop an initial use case diagram one can begin with the information from Table 4. First, each user class and role should be written as an actor (a stick figure that represents a role) with appropriate text representing the combination of user class and role underneath. Second, each desired operational capability can be represented as a use case (an oval) with appropriate text representing the operation. Finally, lines can be drawn that link an actor to a use case, thus making the connection between a user class-role and desired operational capability.

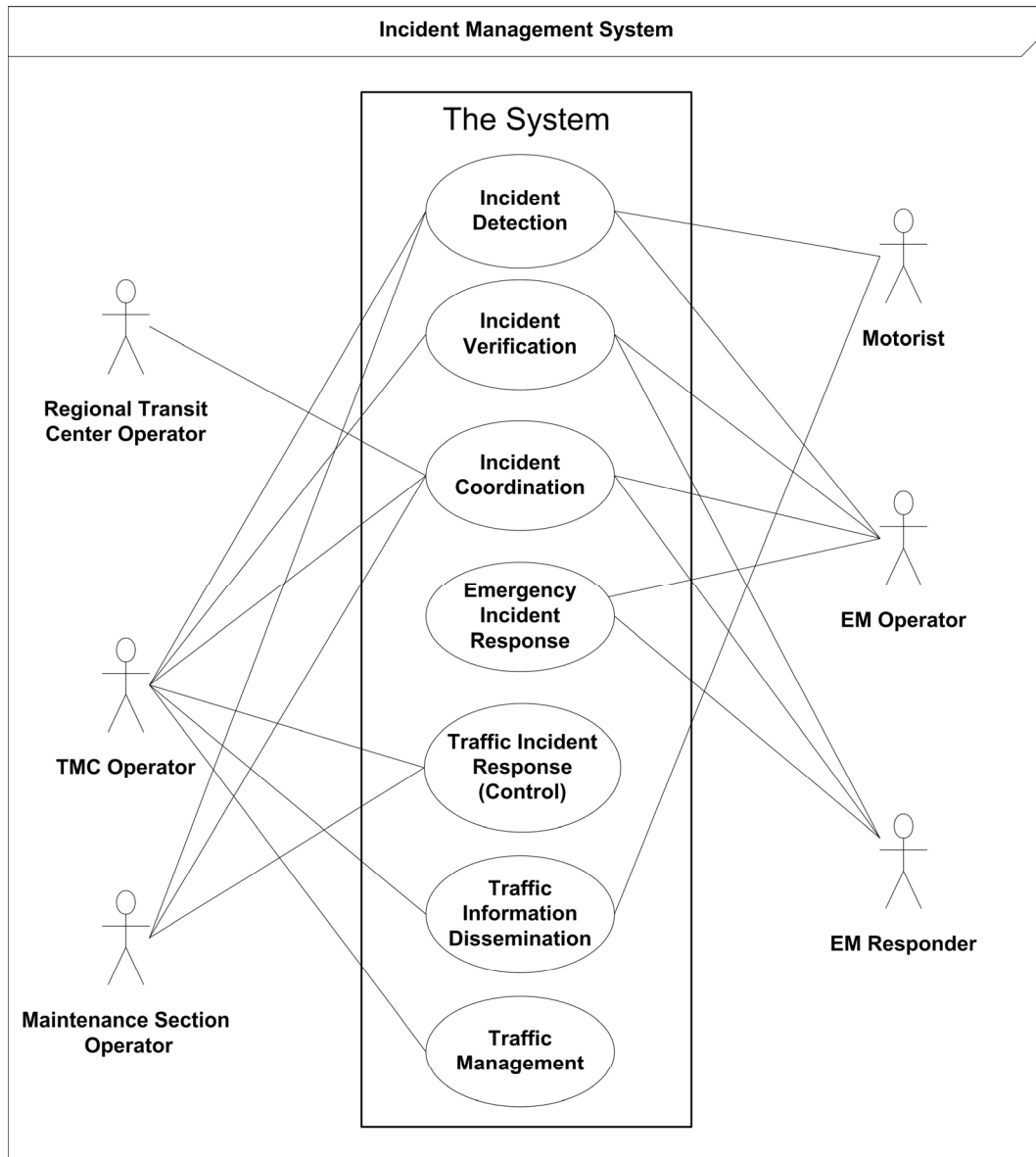


Figure 4. Example Use Case Diagram

Figure 5 illustrates an operational scenario, related to exchange of information between TMC and other centers. This is the initial draft. The example comes from the previous use case named Incident Coordination. This example is only one of several operational scenarios that might occur related to the exchange of information between TMCs and other centers. The use case and activity diagram represent one of several ways to graphically represent operational scenario development.

For each use case, an activity diagram and a corresponding narrative that defines the sequence of steps in the operational scenario can be developed. The activity diagram illustrates the role of each user and is organized on the basis of partitions placed vertically (also called *swim lanes*) to provide information to the reader. At the top of each swim lane, the name of each user role taking action is identified. For example, Figure 5 shows three swim lanes wherein the user roles are TMC Operator, Regional Transit Center Operator, and Maintenance Section Operator.

Each activity diagram has a start and finish to indicate the beginning and end of the operational scenario. Individual activities are shown as rounded rectangles and contain a name. The activity step, shown as a number inside a circle next to the activity, correlates an element in the activity diagram with the narrative. The fork or join symbol indicates synchronization of activities and is shown as a solid line. Activities can be split (fork) or be brought together (join) in a sequence step before continuing with other activities. One last element of an activity diagram, not shown on this diagram, is a decision symbol (shown as a diamond) used to allow the sequence of activities to branch based on a decision and outcomes.

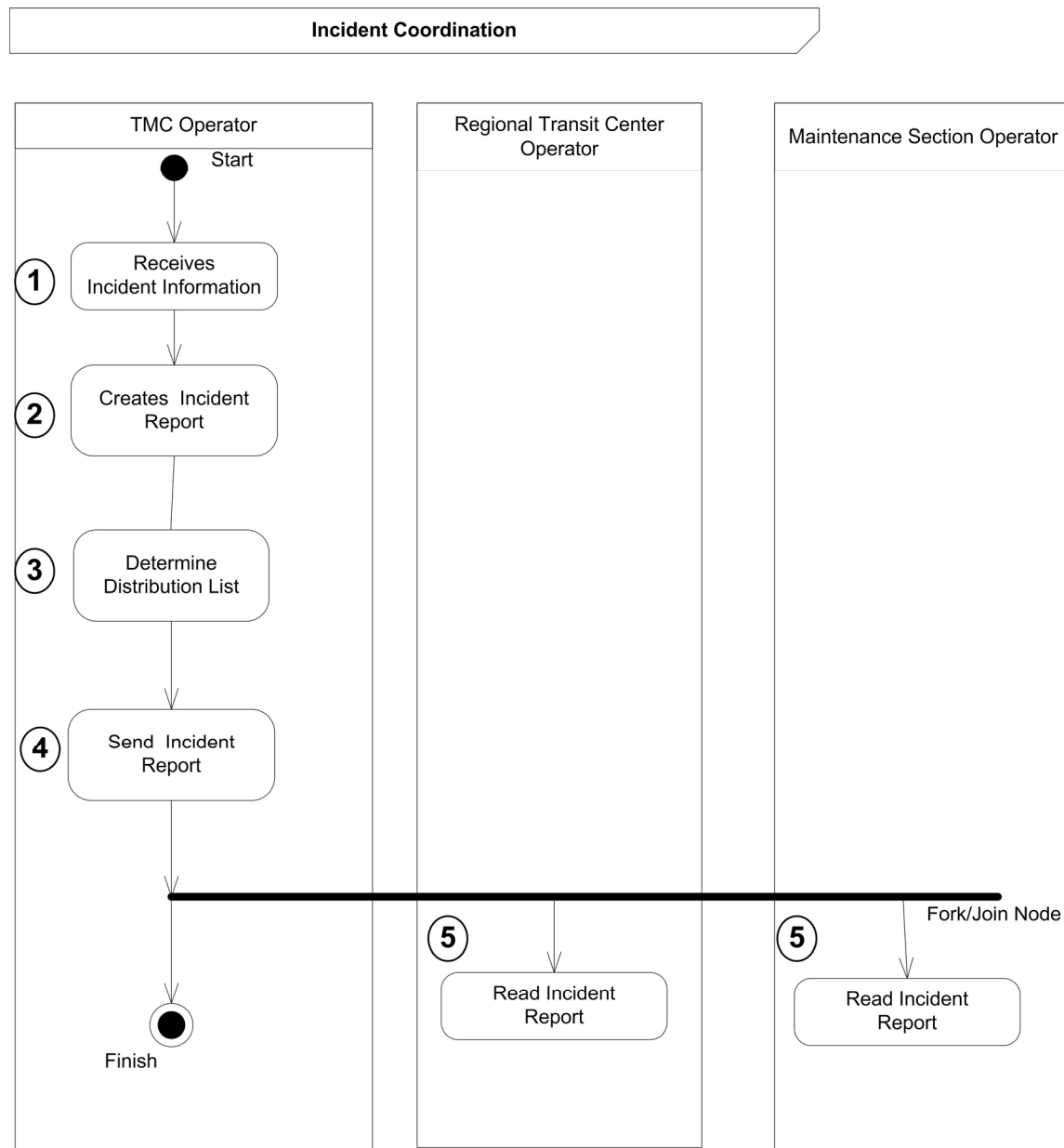


Figure 5. Example Operational Scenario Activity Diagram: Incident Coordination

A narrative of the sequence of activities can be developed by examining each step of the activity diagram from start to finish. Also, it is important to identify any preconditions or assumptions about the state of operations in effect before the scenario start, for example, whether the center is in a normal operations mode or emergency mode. An operational scenario narrative describing the example shown in Figure 5 follows.

Operational Scenario Narrative

Activity: Incident Coordination

Preconditions: This operational scenario begins after an incident has already been detected and responders have already been dispatched to the incident site.

Steps:

1. This operational scenario begins when a TMC operator receives information regarding an incident (traffic flow, traffic images, etc.) and determines to send the incident information to other regional providers of transportation or emergency services.
2. TMC operator creates an incident report.
3. TMC operator determines the list of centers that should receive this incident report.
4. TMC operator sends the incident report to regional transit center operators and maintenance section operators.
5. The regional transit center operator and maintenance section operator read the incident report.

3.3.3.2 Refining the Operational Scenarios

Through interaction with users and subject matter experts, the operational scenarios description can be refined to eliminate redundancy, resolve conflicts, identify gaps in operations that must be filled, and account for any additional operations activities that had not been considered previously. For example, in a review of the previous operational scenarios, the users identified the following issues:

- a) The Incident Coordination activity needed to be renamed Incident Notification to more properly describe the activity.
- b) The Maintenance Section and Regional Transit Operators (Users) could be generalized into a Relevant Authorized Center Operator to eliminate redundancy.
- c) Activities were added to account for verification of incident information sent and received (verification of who was sending and what was sent).
- d) An activity was added to allow receiving centers to confirm to the originating center (TMC Operator) that they had received the incident report.

Figure 6 shows the updated (final draft) use case. Figure 7 shows the final draft operational scenario activity diagram and narrative.

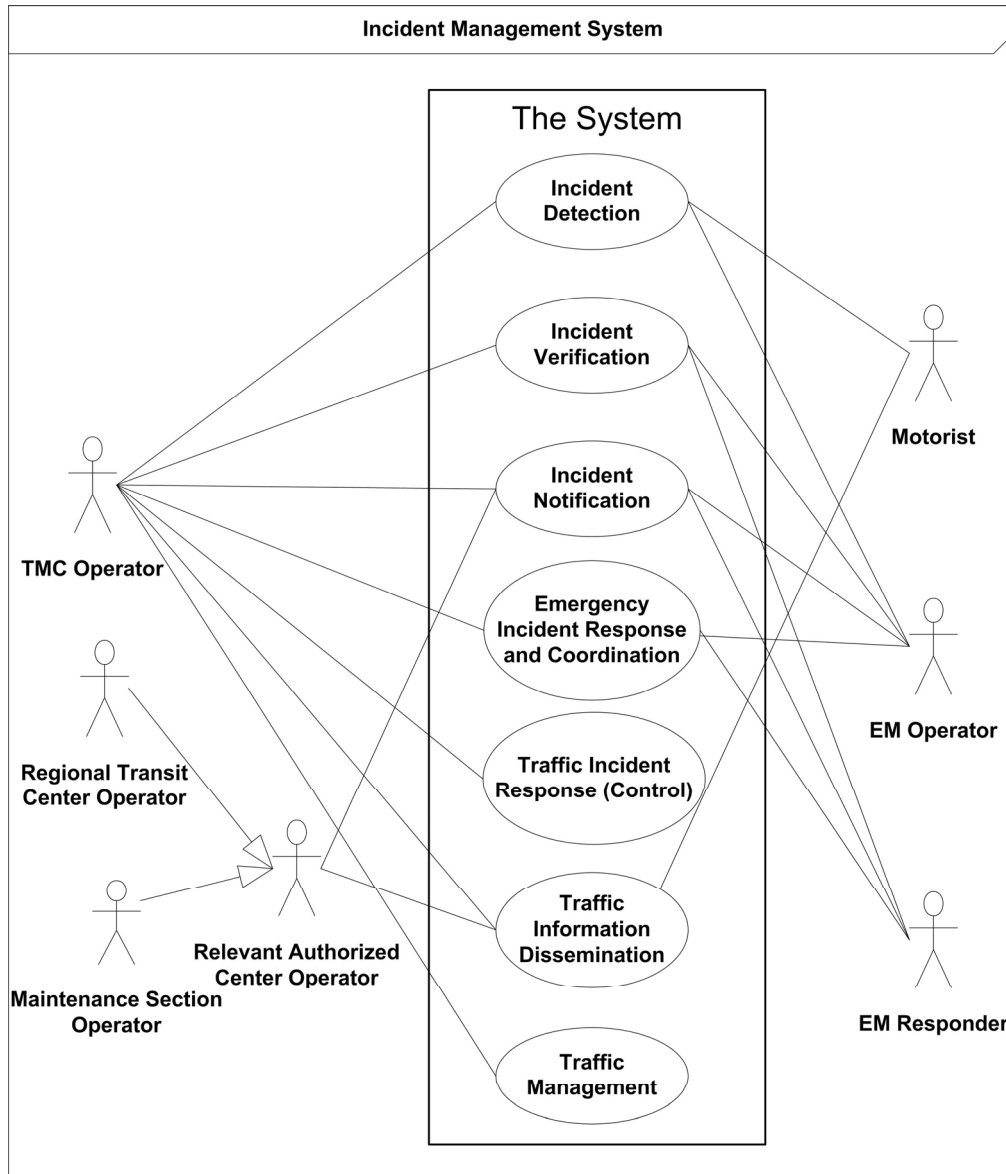


Figure 6. Example Use Case Diagram

Guide for Implementing IEEE 1512 Using a Systems Engineering Process Concept Definition Phase

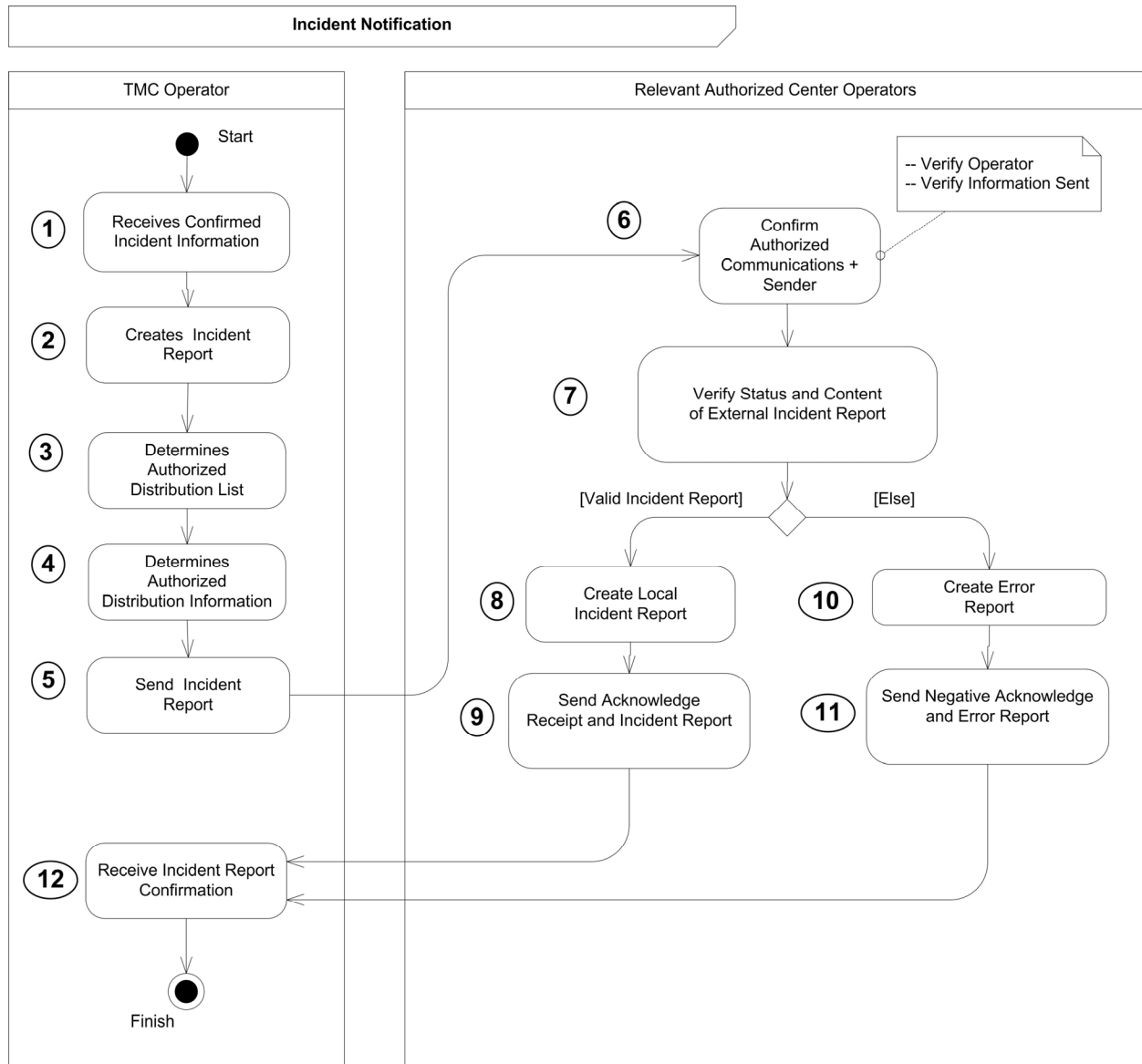


Figure 7. Example Operational Scenario Activity Diagram: Incident Notification

Operational Scenario Narrative

Activity: Incident Notification

Preconditions: This operational scenario begins after an incident has already been detected, first responders have been dispatched to the incident site, and the incident confirmed.

Steps:

1. TMC operator receives information about a confirmed incident.
2. TMC operator creates an incident report that includes information regarding an incident (and optionally traffic flow, traffic images, etc.).
3. TMC then determines which Relevant Authorized Centers (EMC, Maintenance Sections, and Transit Centers) to notify about the incident.
4. TMC determines what authorized information needs to be provided for each of the Relevant Authorized Centers.
5. TMC then sends the Relevant Authorized Centers an incident report.
6. Relevant Authorized Centers confirm that the TMC is an authorized provider of incident information. This step involves verification of the operator sending the information and verification that the information sent is complete.
7. Relevant Authorized Centers confirm that the TMC incident report is valid.

The following sequence occurs if the TMC incident report is valid:

8. The Relevant Authorized Centers will create or update their local incident report (this is the incident report that the Relevant Authorized Centers use at their respective centers).
9. The Relevant Authorized Centers send a confirmation to the TMC acknowledging receipt of the incident report.

The following sequence occurs if the TMC incident report is not valid:

10. The Relevant Authorized Centers create an error report documenting the reason the report is not acceptable and useable.
11. The Relevant Authorized Centers send a confirmation receipt with a negative acknowledgement and error report to the TMC.

The following step is always performed:

12. The TMC receives a confirmation receipt from the Relevant Authorized Centers.

3.4 User Needs Wording

Definition of user needs is one of the key aspects of the system engineering process. User needs tend to remain stable over time. (If needs changed frequently, it would be impossible to build a system interface to satisfy those needs.) It is this inherent stability in the user needs that bounds the scope of the system interface. Well-written needs describe one or more system features and the intent of that need to address a user problem or responsibility. User needs then drive the requirements for the system interface definition and allow development of complete and correct requirements. It is very important that the user needs be developed with all users involved. Creating user needs might involve several iterations to resolve conflicts. After it is complete, agreed upon, and approved, the user needs help manage the expectations of users and the rest of the development process.

Defining user needs is a critical step in the development of a system interface definition and a driver for the next step in the process, i.e., system interface requirements elicitation. If a ConOps for a system or project has been prepared but it does not contain needs, it is considered incomplete. Actions necessary to develop the needs thus completing the ConOps should be taken. Clearly stated user needs will improve the chances of having requirements that, upon realization, satisfy the stated user needs. Criteria for well-written user needs include:

- **Uniquely Identifiable.** Each need must be uniquely identified, i.e., each need shall be assigned a unique number and title
- **Major Desired Capability.** Each need shall express a major desired capability in the system, regardless of whether the capability exists in the current system or situation or is a gap
- **Solution Free.** Each need shall not have a state solution, thus giving designers flexibility and latitude to produce a feasible solution
- **Capture the intent and rationale.** Each need shall capture the rationale or intent as to why the capability is needed in the system

The final list of formal user needs for this example is shown in Table 5. In some cases redundant needs were eliminated or combined. The User Needs table includes a column that shows a unique identifier for each need, in this case, a concatenation of the User ID plus an enumeration (1, 2, 3, and so forth). A second column of information shows the User Need. In this example, a user's needs are organized by the user.