Chapter 1

SYSTEMS AND SOFTWARE REQUIREMENTS SPECIFICATION (SSRS) TEMPLATE

Version A.3a, October 2013

FOREWORD

This document was written to provide software development projects with a template for generating a System and Software Requirements Specification (SSRS). This document is based on a template originally written by the U.S. Navy Research, Development, Test and Evaluation Division in June 1997 in accordance with the MIL-STD-498 DID (DI-IPSC-81433). The template was updated by the University of Idaho's Center for Secure and Dependable Systems (CSDS) in June 2008 to adhere to IEEE Std. 830-1998, *IEEE Recommended Practice for Software Requirements Specifications* [1], and IEEE Std. 12207-2008, *Systems and Software Engineering – Software Life Cycle Processes*[2]. It was then adapted in September 2008, 2010, 2013, and 2014 for use in UI CS 383.

The SSRS template begins on the next page. Just throw away this page and enter your project specifications into the following template. Don't forget to change the headers and footers as necessary.

DOCUMENT CONVENTIONS

[Text] Replace this text with your project specification text.

text in italics Notes or instructions to the author. **Delete in submitted document.**

[1] IEEE Std. 830-1998, *Recommended Practice for Software Requirements Specifications*, Institute of Electrical and Electronics Engineers, 345 East 47 th St. New York, NY, USA, 10017-2394.

[2] ISO/IEC 12207, IEEE Std. 12207-2008, *Systems and software engineering – Software life cycle processes*, 2nd ed., Institute of Electrical and Electronics Engineers, 445 Hoes Lane, Piscataway, NJ, USA, 08854.

SYSTEMS AND SOFTWARE REQUIREMENTS SPECIFICATION (SSRS) FOR

[insert your project name]

[replace image above with a cooler logo]

University of Idaho
Open Space. Open Minds.

Version [[insert version number]]
[[insert date]]

Prepared for:
[insert company name, address, and contact info]

Prepared by:
[insert your name(s)]
University of Idaho
Moscow, ID 83844-1010

[insert your project name] SSRS

RECORD OF CHANGES

Change number	Date completed	Location of change (e.g., page or figure #)	**A** **M** **D**	Brief description of change	Approved by (initials)	Date Approved

*__A__ - ADDED **M** - MODIFIED **D** - DELETED

[INSERT YOUR PROJECT NAME] SSRS
TABLE OF CONTENTS

Section Page

1 Introduction

This section the document should introduce the project, customer, audience, etc., without delving into too much detail, because those details are provided in subsequent sections.

1.1 IDENTIFICATION

This paragraph shall contain a full identification of the system and the software to which this document applies, including, as applicable, identification number(s), title(s), abbreviation(s), version number(s), and release number(s).

The software system being considered for development is referred to as [insert name and or id number]. The customer providing specifications for the system is [insert name and contact info]. The ultimate customer, or end-user, of the system will be [insert name and contact info]. This is a [new | redesign] project effort, so the version under development is version [insert version and release number].

1.2 PURPOSE

This paragraph shall contain a brief statement on the purpose of the system and software being developed, and the intended audience for this document.

The purpose of the system under development is to [insert your text here]. While the system will be used by [insert intended users], this document is intended to be read and understood by UICS software designers and coders. [Optional: The document will also be vetted or approved by [insert approval people]].

1.3 SCOPE

This paragraph shall briefly summarize the history of system development, operation, and maintenance; identify the project sponsor, acquirer, user, developer, and support agencies; identify current and planned operating sites; and list other relevant documents.

[insert your text here]

1.4 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

This section shall list and define all special terms, acronyms and abbreviations used throughout this document. A tabular form is preferable, but not mandatory.

Term or Acronym	**Definition**
Alpha test	Limited release(s) to selected, outside testers
Beta test	Limited release(s) to cooperating customers wanting early access to developing systems
Final test	aka, Acceptance test, release of full functionality to customer for approval
DFD	Data Flow Diagram
SDD	Software Design Document, aka SDS, Software Design Specification
SRS	Software Requirements Specification
SSRS	System and Software Requirements Specification

1.5 REFERENCES

This section shall list full bibliographic citations of all documents referenced in this report. This section shall also identify the source for all materials not available in printed form (e.g., web-based information) and list the complete URL along with owner, author, posting date, and date last visited.

[insert your citations here]

1.6 OVERVIEW AND RESTRICTIONS

This paragraph shall describe the organization of this document and shall describe any security or privacy considerations associated with its use.

This document is for limited release only to UI CS personnel working on the project and [state others who will receive the document].

Section 2 of this document describes the system under development from a holistic point of view. Functions, characteristics, constraints, assumptions, dependencies, and overall requirements are defined from the system-level perspective.

Section 3 of this document describes the specific requirements of the system being developed. Interfaces, features, and specific requirements are enumerated and described to a degree sufficient for a knowledgeable designer or coder to begin crafting an architectural solution to the proposed system.

Section 4 provides the requirements traceability information for the project. Each feature of the system is indexed by the SSRS requirement number and linked to its SDD and test references.

Sections 5 and up are appendices including original information and communications used to create this document.

2 OVERALL DESCRIPTION

This section of the document should describe the general factors that affect the product and its requirements. This section does not state specific requirements. Instead, it provides the background for those requirements, which are defined in detail in Section 3.

2.1 PRODUCT PERSPECTIVE

This subsection of the document should put the product into perspective with other related products. If the product is independent and totally self-contained, it should be so stated here. If the document de[FB01?]nes a product that is a component of a larger system, then this subsection should relate the requirements of that larger system to functionality of the software and should identify interfaces between that system and the software. A block diagram showing the major components of the larger system, interconnections, and external interfaces can be helpful.

[insert your text here]

2.2 PRODUCT FUNCTIONS

This subsection of the document should provide a summary of the major functions that the software will perform. For the sake of clarity The functions should be organized in a way that makes the list of functions understandable to the customer or to anyone else reading the document for the first time. Textual or graphical methods can be used to show the different functions and their relationships. Such a diagram is not intended to show a design of a product, but simply shows the logical relationships among variables.

[insert your text here]

2.3 USER CHARACTERISTICS

This subsection of the document should describe those general characteristics of the intended users of the product including educational level, experience, and technical expertise. It should not be used to state speci[FB01?]c requirements, but rather should provide the reasons why certain speci[FB01?]c requirements are later speci[FB01?]ed in Section 3 of this document.

[insert your text here]

2.4 CONSTRAINTS

This subsection of the document should provide a general description of any other items that will limit the developer's options. These include: a) Regulatory policies; b) Hardware limitations (e.g., signal timing requirements); c) Interfaces to other applications; d) Parallel operation; e) Audit functions; f) Control functions; g) Higher-order language requirements; h) Signal handshake protocols; i) Reliability requirements; j) Criticality of the application; k) Safety and security considerations.

[insert your text here]

2.5 ASSUMPTIONS AND DEPENDENCIES

This subsection of the document should list each of the factors that affect the requirements stated in the document. These factors are not design constraints on the system and/or software but are, rather, any changes to them that can affect the requirements in the document. For example, an assumption may be that a speci[FB01?]c operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not available, the document would then have to change accordingly.

[insert your text here]

2.6 SYSTEM LEVEL (NON-FUNCTIONAL) REQUIREMENTS

This subsection of the document should identify system level (whole, not functional) requirements that impact the construction, operation, packaging and delivery of the system and software.

2.6.1 Site dependencies

This paragraph shall specify site-dependent operational parameters and needs (such as parameters indicating operation-dependent targeting constants or data recording). . The requirements shall include, as applicable, number of each type of equipment, type, size, capacity, and other required characteristics of processors, memory, input/output devices, auxiliary storage, communications/ network equipment, and other required equipment or software that must be used by, or incorporated into, the system. Examples include operating systems, database management systems, communications/ network software, utility software, input and equipment simulators, test software, and manufacturing software. The correct nomenclature, version, and documentation references of each such device or software item shall be provided.

[insert your text here]

2.6.2 Safety, security and privacy requirements

This paragraph shall specify the system requirements, if any, concerned with maintaining safety, security and privacy. These requirements shall include, as applicable, the safety, security and privacy environment in which the system must operate, the type and degree of security or privacy to be provided, and the criteria that must be met for safety/security/privacy certification and/or accreditation.

[insert your text here]

2.6.3 Performance requirements

This paragraph should specify both the static and the dynamic numerical performance requirements placed on the soft ware or on human interaction as a whole. Static numerical requirements may include the following: a) The number of terminals to be supported; b) The number of simultaneous users to be supported; c) Amount and type of information to be handled. Dynamic numerical requirements may include, for example, the numbers of transactions and tasks and the amount of data to be processed within certain time periods for both normal and peak workload conditions. All of these requirements should be stated in measurable terms. For example, "95% of the transactions shall be processed in less than 1msec."

[insert your text here]

2.6.4 System and software quality

This paragraph shall specify the requirements, if any, concerned with hardware and software quality factors identified in the contract. Examples include quantitative requirements regarding the system's functionality (the ability to perform all required functions), reliability (the ability to perform with correct, consistent results), maintainability (the ability to be easily corrected), availability (the ability to be accessed and operated when needed), flexibility (the ability to be easily adapted to changing requirements), portability (the ability to be easily modified for a new environment), reusability (the ability to be used in multiple applications), testability (the ability to be easily and thoroughly tested), usability (the ability to be easily learned and used), and other attributes.

[insert your text here]

2.6.5 Packaging and delivery requirements

This paragraph shall specify the requirements, if any, for packaging, labeling, handling and delivery of the system being developed to the customer.

The executable system and all associated documentation (i.e., SSRS, SDD, code listing, test plan (data and results), and user manual) will be delivered to the customer on CD's and/or via email, as specified by the customer at time of delivery. Although document "drops" will occur throughout the system development process, the final, edited version of the above documents will accompany the final, accepted version of the executable system.

2.6.6 Personnel-related requirements

This paragraph shall specify the system requirements, if any, included to accommodate the number, skill levels, duty cycles, training needs, or other information about the personnel who will use or support the system under development. These requirements shall include, as applicable, considerations for the capabilities and limitations of humans; foreseeable human errors under both normal and extreme conditions; and specific areas where the effects of human error would be particularly serious. Examples include requirements for color and duration of error messages, physical placement of critical indicators or keys, and use of auditory signals.

The system under development has no special personnel-related characteristics.

2.6.7 Training-related requirements

This paragraph shall specify the system requirements, if any, pertaining to training. Examples include training software, tutorials, or help information to be included in the system.

No training materials or expectations are tied to this project other than the limited help screens built into the software and the accompanying user manual.

2.6.8 Logistics-related requirements

This paragraph shall specify the system requirements, if any, concerned with logistics considerations. These considerations may include: system maintenance, software support, system transportation modes, supply-system requirements, impact on existing facilities, and impact on existing equipment.

[Insert a description of the minimum hardware requirements and OS and application software dependencies here]

2.6.9

2.6.10 Other requirements

This paragraph shall specify additional system level requirements, if any, not covered in the previous paragraphs.

[insert your text here]

2.6.11 Precedence and criticality of requirements

This paragraph shall specify, if applicable, the order of precedence, criticality, or assigned weights indicating the relative importance of the requirements in this specification. Examples include identifying those requirements deemed critical to safety, to security, or to privacy for purposes of singling them out for special treatment. If all requirements have equal weight, this paragraph shall so state.

[insert your text here]

3 SPECIFIC REQUIREMENTS

This section of the document should contain all of the software requirements to a level of detail sufficient to enable designers to design a system to satisfy those requirements, and testers to test that the system satisfies those requirements. Throughout this section, every stated requirement should be externally perceivable by users, operators, or other external systems. These requirements should include at a minimum a description of every input into the system, every output from the system, and all functions performed by the system in response to an input or in support of an output. As this is often the largest and most important part of the document, all requirements should be uniquely identifiable and careful attention should be given to organizing the requirements to maximize readability.

3.1 EXTERNAL INTERFACE REQUIREMENTS

This subsection should be a detailed description of all inputs into and outputs from the software system. It should complement the constraints and dependencies defined in earlier sections, but not repeat that information. Hardware, software, user, and other communication interfaces need to be specified. Use the four subsections listed below or the table on the next page, or some combination of both.

3.1.1 Hardware Interfaces

[insert your text here]

3.1.2 Software Interfaces

[insert your text here]

3.1.3 User Interfaces

[insert your text here]

3.1.4 Other Communication Interfaces

[insert your text here]

External Interface Requirements

Hardware Interfaces

Name	Source/Destination	Description	Type/range	Dependencies

Software Interfaces

Name	Source/Destination	Description	Type/range	Dependencies

User Interfaces

Name	Source/Destination	Description	Type/range	Dependencies

Other Communication Interfaces

Name	Source/Destination	Description	Type/range	Dependencies

3.2 SYSTEM FEATURES

Functional requirements should define the fundamental actions (i.e., features) that must take place in the software in accepting and processing the inputs and in processing and generating the outputs. These requirements are given in the form of ***Use Cases*** *where possible, denoting a concrete use (discrete user-performable task) of the system. Use case diagrams are followed by use case descriptions, followed by any non-task features. Non-task features are generally listed as "shall" statements starting with "The system shall. . . " These include: a) Validity checks on the inputs; b) Exact sequence of operations; c) Responses to abnormal situations, including error detection, handling and recovery; d) Parameter specification and usage; e) Relationship of outputs to inputs, including formulas for input to output conversion.*

It may be appropriate to partition the functional requirements into sub functions or subprocesses, but that decomposition (here) does not imply that the software design will also be partitioned that way. You should repeat subsections 3.2.i for every specified feature defined for the system or software.

3.2.1 Use Case Diagrams

[insert 1+ use case diagrams here]

3.2.2 System feature 1: [insert feature name here]

For each feature, you should either provide a Use Case Description ***or*** *a Non-task feature description, whichever is more appropriate.*

Use Case Description	**Non-task feature description**
Name Actors Goals Preconditions **Summary** Related use cases **Steps** 1. ... 2. ... Alternatives Postconditions	**Introduction/Purpose** [insert your text here] **Input/Output sequence** [insert your text here] **Design constraints** [insert your text here] **Performance requirements** [insert your text here] **Detailed functional requirements** **Functional requirement 1.1** [insert your text here] **Functional requirement 1.2** [insert your text here] . . . **Functional requirement 1.[n]** [insert your text here]

3.2.3 System feature 2: [insert feature name here]

Write another feature description, as either a use case or a non-task feature description

. . . do as many as necessary, probably a lot

3.2.4 System feature [m]: [insert feature name here]

Write a final feature description, as either a use case or a non-task feature description

4 REQUIREMENTS TRACEABILITY

This section shall contain traceability information from each system requirement in this specification to the system (or subsystem, if applicable) requirements it addresses. A tabular form is preferred, but not mandatory.

Feature Name	**Req No.**	**Requirement Description**	**Priority**	**SDD**	**Alpha Release**		**Beta Release**	
					Test Case(s)	**Test Res.**	**Test Case(s)**	**Test Res.**
	1.1							
	1.2							
	...							
	1.[n]							
	2.1							
	2.2							
	...							
	2.[n]							
	3.1							
	3.2							
	...							
	3.[n]							
...	...							
	[m].1							
	[m].2							
	...							
	[m.n]							

Priorities are: **M**andatory, **L**ow, **H**igh

SDD link is version and page number or function name.

Test cases and results are file names and **P**ass/**F**ail or % passing.

5 APPENDIX A. [insert name here]

Include copies of specifications, mockups, prototypes, etc. supplied or derived from the customer. Appendices are labeled A, B, . . . n. Reference each appendix as appropriate in the text of the document.

[insert appendix A here]

6 APPENDIX B. [insert name here]

[insert appendix B here]

Chapter 2

SYSTEM AND SOFTWARE DESIGN DESCRIPTION (SSDD) TEMPLATE

(Incorporating Architectural Views and Detailed Design Criteria)
Version A.2, November 2010
FOREWORD

This template was created to provide system and software development projects with a model System and Software Design Description (SSDD) that incorporates both architectural views and detailed design criteria. The template is based on work compiled by Dr. Paul Oman from a large collection of software engineering design document standards discussed in Section 1.5. It has been edited and updated by Dr. Clint Jeffery for use in UI CS 383.

The SSDD template begins on the next page. Just throw away this page and enter your project specifications into the following template. Don't forget to change the headers and footers as necessary. The following conventions are used to guide you in developing your SSDD:

[Text] **Replace** this text with your project design text.

text in italics Notes/instructions to the author. **Delete in your finished document.**

SYSTEM AND SOFTWARE DESIGN DESCRIPTION (SSDD): Incorporating Architectural Views and Detailed Design Criteria FOR

[state program/system name here]

Version [[insert version number]]
[[insert date]]

Prepared for:
[state customer name(s) here]

Prepared by:
[insert your name(s)]
University of Idaho
Moscow, ID 83844-1010

CS383 SSDD

RECORD OF CHANGES (Change History)

Change Number	Date completed	Location of change (e.g., page or figure #)	A M D	Brief description of change	Approved by (initials)	Date approved

A - ADDED M - MODIFIED D – DELETED

[put program /system name here]
TABLE OF CONTENTS

Section Page

1 INTRODUCTION

This section of this document should introduce this document and its audience, and the project, the system, and the software object of this SSDD. For compliance with ISO/IEC 42010:2007 (Section 5.1) (and ISO/IEC 12207:2008) at a minimum the following information shall be included in this SSDD document: Date of Document Issue, Document Status, Document Issuing Organization, Document Change History, Document Summary, Document Scope, Document Context, Glossary, and References.

1.1 IDENTIFICATION

This subsection shall contain a full identification of this document, and the system and software to which this document applies, including, as applicable, identification number(s), title(s), abbreviation(s), version number(s) for the document, the system, and the software, and software release number(s).

[Insert text here.]

1.2 DOCUMENT PURPOSE, SCOPE, AND INTENDED AUDIENCE

This subsection shall contain a statement on the purpose of this document, the scope or context of the document, and the intended audience.

1.2.1 Document Purpose

[Insert text here.]

1.2.2 Document Scope and/or Context

[Insert text here.]

1.2.3 Intended Audience for Document

[Insert text here.]

1.3 SYSTEM AND SOFTWARE PURPOSE, SCOPE, AND INTENDED USERS

This subsection shall contain a statement on the purpose of this system and software, the scope or context of the system and software, and the intended users of the system and software. The subsection shall describe the system boundaries and the software boundaries, both with respect to their containing system and other systems of interest. It shall be clear from this section: 1) what is the relationship of other systems-of-interest with the system described in this document and 2) what is the relationship between the system and the particular software described in this SSDD.

1.3.1 System and Software Purpose

[Insert text here.]

1.3.2 System and Software Scope/or Context

[Insert text here.]

1.3.3 Intended Users for the System and Software

[Insert text here.]

1.4 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

This section shall list and define all special terms, acronyms and abbreviations used throughout this document. A tabular form is preferable, but not mandatory.

Acquirer	The person, team, or organization that pursues a system or software product or service from a supplier. The acquirer may be a buyer, customer, owner, purchaser, or user. ISO/IEC 42010:2007 (Section 3.1).
AD	Architectural Description: "A collection of products to document an architecture" ISO/IEC 42010:2007 (Section 3.4).
Alpha test	Limited release(s) to selected, outside testers
Architect	"The person, team, or organization responsible for systems architecture" ISO/IEC 42010:2007 (Section 3.2).
Architectural Description	(AD) "A collection of products to document an architecture" ISO/IEC 42010:2007 (Section 3.4).
Architectural View	"A representation of a whole system from the perspective of a related set of concerns" ISO/IEC 42010:2007 (Section 3.9).
Architecture	"The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution" ISO/IEC 42010:2007 (Section 3.5).
Beta test	Limited release(s) to cooperating customers wanting early access to developing systems
Design Entity	"An element (component) of a design that is structurally and functionally distinct from other elements and that is separately named and referenced" IEEE STD 1016-1998 (Section 3.1).
Design View	"A subset of design entity attribute information that is specifically suited to the needs of a software project activity" IEEE STD 1016-1998 (Section 3.2).
Final test	aka, Acceptance test, release of full functionality to customer for approval
DFD	Data Flow Diagram
SDD	Software Design Document, aka SDS, Software Design Specification
Software Design Description	"A representation of a software system created to facilitate analysis, planning, implementation, and decision making, A blueprint or model of a software system. The SDD is used as the primary medium for communicating software design information" IEEE STD 1016-1998 (Section 3.4).
SRS	Software Requirements Specification
SSDD	System and Software Design Document
SSRS	System and Software Requirements Specification
System	"A collection of components organized to accomplish a specific function or set of functions" ISO/IEC 42010:2007 (Section 3.7).

Term or Acronym	**Definition**
System and Software Architecture and Design Description	An architectural and detailed design description that includes a software system within the context of its enclosing system and describes the enclosing system, the enclosed software, and their relationship and interfaces.
System Stakeholder	"An individual, team, or organization (or classes thereof) with interests in, or concerns, relative to, a system" ISO/IEC 42010:2007 (Section 3.8).

1.5 DOCUMENT REFERENCES

This subsection shall list full bibliographic citations of all documents referenced in this SSDD. This subsection shall also identify the source for all materials not available in printed form (e.g., web-based information) and list the complete URL along with owner, author, posting date, and date last visited.

1) CSDS, *System and Software Requirements Specification Template*, Version 1.0, July 31, 2008, Center for Secure and Dependable Systems, University of Idaho, Moscow, ID, 83844.

2) ISO/IEC/IEEE, *IEEE Std 1471-2000 Systems and software engineering – Recommended practice for architectural description of software intensive systems,* First edition 2007-07-15, International Organization for Standardization and International Electrotechnical Commission, (ISO/IEC), Case postale 56, CH-1211 Genève 20, Switzerland, and The Institute of Electrical and Electronics Engineers, Inc., (IEEE), 445 Hoes Lane, Piscataway, NJ 08854, USA.

3) IEEE, *IEEE Std 1016-1998 Recommended Practice for Software Design Descriptions*, 1998-09-23, The Institute of Electrical and Electronics Engineers, Inc., (IEEE) 445 Hoes Lane, Piscataway, NJ 08854, USA.

4) 3) ISO/IEC/IEEE, *IEEE Std. 15288-2008 Systems and Software Engineering – System life cycle processes,* Second edition 2008-02-01, International Organization for Standardization and International Electrotechnical Commission, (ISO/IEC), Case postale 56, CH-1211 Genève 20, Switzerland, and The Institute of Electrical and Electronics Engineers, Inc., (IEEE), 445 Hoes Lane, Piscataway, NJ 08854, USA.

5) ISO/IEC/IEEE, IEEE Std. 12207-2008, *Systems and software engineering – Software life cycle processes,* Second edition 2008-02-01, International Organization for Standardization and International Electrotechnical Commission, (ISO/IEC), Case postale 56, CH-1211 Genève 20, Switzerland, and The Institute of Electrical and Electronics Engineers, Inc., (IEEE), 445 Hoes Lane, Piscataway, NJ 08854, USA.

1.6 DOCUMENT OVERVIEW

This subsection shall provide an overview of the organization of this SSDD.

Section 2 of this document describes the system and software constraints imposed by the operational environment, system requirements and user characteristics, and then identifies the system stakeholders and lists describes their concerns and mitigations to those concerns.

Section 3 of this document describes the system and software architecture from several viewpoints, including, but not limited to, the developer's view and the user's view.

Section 4 provides detailed design descriptions for every component defined in the architectural view(s). Sections 5 provides traceability information connecting the original specifications (referenced above) to the architectural components and design entities identified in this document.

Section 6 and beyond are appendices including original information and communications used to create this document.

1.7 DOCUMENT RESTRICTIONS

This subsection shall describe security and privacy considerations associated with the information contained in this document as well as rules and restrictions for the use and distribution of this SSDD and the information contained within.

This document is for LIMITED RELEASE ONLY to UI CS personnel working on the project and [state others who will receive the document].

2 CONSTRAINTS and stakeholder concerns

This section of the document shall identify environmental or usability constraints placed upon the development and use of the system and software, the stakeholders of the system and software, and their concerns about the system and software, if any.

2.1 CONSTRAINTS

This subsection shall identify and describe in detail the architectural and usability constraints that are imposed by the physical environment or system requirements or the user characteristics.

2.1.1 Environmental constraints.

[Insert text here.]

2.1.2 System requirement constraints.

[Insert text here.]

2.1.3 User characteristic constraints.

[Insert text here.]

2.2 STAKEHOLDER CONCERNS

This subsection shall identify all the system and software stakeholders. Some categories have already been included, add more categories as needed. Within each category add the list of stakeholders and their details. For compliance with ISO/IEC 42010:2007 (§5.2) at a minimum the following concerns shall be identified and described for the system and software object of this SSDD: appropriateness of the architected solution for achieving its desired mission, feasibility of construction, risks of system construction and operation to all stakeholders, maintainability, deployability, and evolvability. Other stakeholder concerns for the system and software might be: construction cost, expected lifetime, cost of operation, cost of maintenance, system safety, data security and privacy, operator and user safety, etc. For each concern make a reference to the corresponding stakeholder(s) (a concern might come from more than one stakeholder).

The following tabular form is preferred, but not required. You may eliminate inappropriate rows and add categories and concerns as needed.

Feasibility of constructing, testing, verifying and deploying the system and software.			
Risks of constructing, deploying, and using the system and software object of this SSDD.			
Concerns with respect to the deployability of the system and software.			
Concerns with respect to the maintainability and evolvability of the system and software.			
Concerns with respect to the security of the data the system and software will handle.			
Concerns with respect to the safety of the people interacting with the system and software.			

Stakeholder x Concern x Mitigation Table			
Stakeholder Concern	**List of Stakeholders (e.g. Acquirers, Developers, Testers, Maintainers, Users, Operators, Auditors, Others)**	**Stated Concern**	**Mitigation Mechanism or Design Criteria Reference Number**
Appropriateness of the system and software in fulfilling its mission(s).			
Cost concerns.			
[list concern]			
[list concern]			

3 SYSTEM AND SOFTWARE ARCHITECTURE

This section of the document shall describe with detail every detailed design entity or component of the system as well as the relationship and interface between them. These architectural entities, when integrated together as specified within this document, shall implement all functions performed by the system in response to an input or in support of an output as described by the System and Software Requirements Specification (SSRS). All architectural entities or components shall: be uniquely identi[FB01?]able, be well described, have clear responsibilities, have well specified interfaces, and have well described interactions with other architectural entities and any external systems. A system's architecture is usually described by using a set of different views, typically one for the developer and others for the customer, users, operators, etc. All necessary views at the architectural level (or high-level design) shall be clearly described in this section. In this section we assume that the reader is familiar with such architectural description languages. For compliance with ISO/ISEC 42010:2007 (§5.4) each view shall include at least the following details: identification, system representation using the corresponding viewpoint, configuration information, languages and modeling techniques, and references to detailed or further descriptions of the viewpoint.

3.1 DEVELOPER'S ARCHITECTURAL VIEW

This subsection contains the descriptions of a system and all of its major components, using the methods, techniques, and languages from the developer's viewpoint. Each viewpoint description includes the viewpoint identification, description, and diagrammatic representation.

3.2 USER'S ARCHITECTURAL VIEW

This subsection contains the descriptions of a system and all of its major components, using the methods, techniques, and languages from the user's viewpoint. Each viewpoint description includes the viewpoint identification, description, and diagrammatic representation.

3.2.1 User's View Identification

Identify the view, state the purpose of the view, and identify major components or processes of the architecture.

[Insert text here.]

3.2.2 User's View Representation and Description

Provide a diagram and description of the user's view of the architecture.

[Insert diagram here.]

3.3 Developer's View Identification

Identify the view, state the purpose of the view, and identify major components or processes of the architecture.

[Insert text here.]

3.3.1 Developer's View Representation and Description

Provide a diagram and description of the developer's view of the architecture.

[Insert diagram here.]

3.3.2 Developer's Architectural Rationale

For compliance with ISO/IEC 42010:2007 (§5.6) an Architectural Description (AD) shall provide the rationale that justified the architect's decisions and selected architectures. An AD shall also provide evidence of the consideration of other alternative architectures and the rationales for discarding them.

[Insert rationale here.]

3.4 [insert name of viewpoint] ARCHITECTURAL VIEW

This subsection contains the descriptions of a system and all of its major components, using the methods, techniques, and languages from other than the developer's or user's viewpoint. Each viewpoint description includes the viewpoint identification, description, and diagrammatic representation.

Repeat this subsection for each viewpoint identified.

3.4.1 [insert name of viewpoint]'s View Identification

Identify the view, state the purpose of the view, and identify major components or processes of the architecture.

[Insert text here.]

3.4.2 [insert name of viewpoint]'s View Representation and Description

Provide a diagram of the developer's view of the architecture.

[Insert diagram and descriptions here.]

3.5 CONSISTENCY OF ARCHITECTURAL VIEWS

For compliance with ISO/IEC 42010:2007 (§5.5) an Architectural Description (AD) shall include a list of all known inconsistencies between the architectural views and an analysis of consistency across all the architectural views.

3.5.1 Detail of Inconsistencies between Architectural Views

[Insert text and graphics here.]

3.5.2 Consistency Analysis and Inconsistency Mitigations

For each inconsistency identified above, provide solutions or mitigations that resolve potential conflicts between the stakeholder viewpoints.

[Insert text or table here.]

4 SOFTWARE DETAILED DESIGN

This section of the document should describe with detail the design of the software being described in this document. The level of detail of the design entities and their relationship and interfaces shall be sufficient to enable software implementers to implement an integrate each of the described components in order to achieve full implementation of the software being described in this SSDD. This section shall specify for each design entity the following information: purpose, processing, data, interfaces, dependencies and relationships, concept of execution, needed resources, design rationale, information for reuse, types of errors, and error handling.

The detailed design must correspond to an existing architectural view, normally the developer's view, but unusual circumstances may call for other detailed design viewpoints. If so, repeat this subsection as needed for those other viewpoints.

4.1 DEVELOPER'S VIEWPOINT DETAILED SOFTWARE DESIGN

Identify the viewpoint and make reference to the diagram or model defining the view.

[Insert text, diagram or model here.]

4.2 COMPONENT/ENTITY DICTIONARY

This subsection shall list and describe all the detailed design entities and their corresponding attributes. Processing and algorithms, data and data structures,and detailed descriptions need NOT be included here, as they will be specified in subsequent sections for each component or entity listed in the table below.

Component/Entity Dictionary				
Name	**Type/Range**	**Purpose/Function**	**Dependencies**	**Subordinates**

4.3 COMPONENT/ENTITY DETAILED DESIGN

4.3.1 Detailed Design for Component/Entity: [insert Component/Entity name here]

Introduction/Purpose of this Component/Entity [insert your text here]

Input for this Component/Entity [insert your text here]

Output for this Component/Entity [insert your text here]

Component/Entity Process to Convert Input to Output [insert your text here]

Design constraints and performance requirements of this Component/Entity [insert your text here]

4.3.2 Detailed Design for Component/Entity: [insert Component/Entity name here]

Introduction/Purpose of this Component/Entity [insert your text here]

Input for this Component/Entity [insert your text here]

Output for this Component/Entity [insert your text here]

Component/Entity Process to Convert Input to Output [insert your text here]

Design constraints and performance requirements of this Component/Entity [insert your text here]

4.3.3 Detailed Design for Component/Entity: [insert Component/Entity name here]

Introduction/Purpose of this Component/Entity [insert your text here]

Input for this Component/Entity [insert your text here]

Output for this Component/Entity [insert your text here]

Component/Entity Process to Convert Input to Output [insert your text here]

Design constraints and performance requirements of this Component/Entity [insert your text here]

...

4.3.4 Detailed Design for Component/Entity: [insert Component/Entity name here]

Introduction/Purpose of this Component/Entity [insert your text here]

Input for this Component/Entity [insert your text here]

Output for this Component/Entity [insert your text here]

Component/Entity Process to Convert Input to Output [insert your text here]

Design constraints and performance requirements of this Component/Entity [insert your text here]

4.4 DATA DICTIONARY

This subsection shall list and describe all the data and data structures defined and/or used by the components and entities specified above. For each data item or structure indicate where it is defined, referenced, and modified.

Data Dictionary				
Name	**Type/Range**	**Defined by...**	**Referenced by...**	**Modified by...**

5 REQUIREMENTS TRACEABILITY

This section shall contain traceability information from each system requirement in this specification to the system (or subsystem, if applicable) requirements it addresses. A tabular form is preferred, but not mandatory. A detailed mapping between requirements and constraints in the SSRS and architectural components and detailed entities in this SSDD is required. For compliance with ISO/IEC 15288:2008 (§6.4.3.3.c) an Architectural Description (AD) shall provide roundtrip traceability between the system and software requirements and the architectural design entities. All requirements and constraints within the SSRS shall map to a set of architectural entities. All entities in all the architectural views shall be associated with either a requirement or constraint in the SSRS or an architectural constraint within this SSDD.

	1.1							
	1.2							
	...							
	1.[n]							
	2.1							
	2.2							
	...							
	2.[n]							
	3.1							
	3.2							
	...							
	3.[n]							
...	...							
	[m].1							
	[m].2							
	...							
	[m.n]							

Priorities are: **M**andatory, **L**ow, **H**igh

SDD link is version and page number or function name.

Test cases and results are file names and **P**ass/**F**ail or % passing.

6 APPENDIX A. [insert name here]

Include copies of specifications, mockups, prototypes, etc. supplied or derived from the customer. Appendices are labeled A, B, . . . n. Reference each appendix as appropriate in the text of the document.

[insert appendix A here]

7 APPENDIX B. [insert name here]

[insert appendix B here]

Chapter 3

TEST PLAN (TP) TEMPLATE

Version 1.1, October 2013

FOREWORD

This template was created to provide system and software development projects with a model Test Plan (TP) document template. The template is based on IEEE 829 Format. It has been edited and updated by Dr. Clint Jeffery for use in UI CS 383.

The TP template begins on the next page. Just throw away this page and enter your project specifications into the following template. Don't forget to change the headers and footers as necessary. The following conventions are used to guide you in developing your TP:

[Text] **Replace** this text with your project design text.

text in italics Notes/instructions to the author. **Delete in your finished document.**

TEST PLAN (TP)

FOR

[state program/system name here]

Version [[insert version number]]
[[insert date]]

Prepared for:
[state customer name(s) here]

University of Idaho
Open Space. Open Minds.

Prepared by:

[insert your name(s)]

University of Idaho

Moscow, ID 83844-1010

CS383 TPD

RECORD OF CHANGES (Change History)

Change Number	Date completed	Location of change (e.g., page or figure #)	A M D	Brief description of change	Approved by (initials)	Date approved

A - ADDED M - MODIFIED D – DELETED

[put program /system name here]

TABLE OF CONTENTS

Section Page

1 TEST PLAN IDENTIFIER

Some type of unique company generated number to identify this test plan, its level and the level of software that it is related to. Preferably the test plan level will be the same as the related software level. The number may also identify whether the test plan is a Master plan, a Level plan, an integration plan or whichever plan level it represents. This is to assist in coordinating software and testware versions within configuration management.

[Insert text here.]

2 REFERENCES

List all documents that support this test plan. Refer to the actual version/release number of the document as stored in the configuration management system. Do not duplicate the text from other documents as this will reduce the viability of this document and increase the maintenance effort.

[Insert text here.]

3 INTRODUCTION

State the purpose of the Plan, possibly identifying the level of the plan (master etc.). This is essentially the executive summary part of the plan.

4 TEST ITEMS

These are things you intend to test within the scope of this test plan. Essentially, something you will test, a list of what is to be tested. This can be developed from the software application inventories as well as other sources of documentation and information.

[Insert text here.]

5 SOFTWARE RISK ISSUES

Identify what software is to be tested and what the critical areas are, such as:

- *1. Delivery of a third party product.*
- *2. New version of interfacing software*

- *3. Ability to use and understand a new package/tool, etc.*
- *4. Extremely complex functions*
- *5. Modifications to components with a past history of failure*
- *6. Poorly documented modules or change requests*

[Insert text here.]

6 FEATURES TO BE TESTED

This is a listing of what is to be tested from the USERS viewpoint of what the system does. This is not a technical description of the software, but a USERS view of the functions.

[Insert text here.]

7 FEATURES NOT TO BE TESTED

This is a listing of what is NOT to be tested from both the Users viewpoint of what the system does and a configuration management/version control view. This is not a technical description of the software, but a USERS view of the functions.

[Insert text here.]

8 APPROACH

This is your overall test strategy for this test plan; it should be appropriate to the level of the plan (master, acceptance, etc.) and should be in agreement with all higher and lower levels of plans. Overall rules and processes should be identified.

- *Are any special tools to be used? What are they?*
- *What metrics will be collected for this test?*
- *How many configurations/platforms are to be tested?*
- *How will elements in the design deemed "untestable" be processed?*

[Insert text here.]

9 ITEM PASS/FAIL CRITERIA

What are the Completion criteria for this plan? This is a critical aspect of any test plan and should be appropriate to the level of the plan. [Insert text here.]

10 SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS

If the number or type of defects reaches a point where the follow on testing has no value, it makes no sense to continue the test; you are just wasting resources.

Specify what constitutes stoppage for a test or series of tests and what is the acceptable level of defects that will allow the testing to proceed past the defects. [Insert text here.]

11 TEST DELIVERABLES

What is to be delivered as part of this plan?

- *Test plan document*
- *Test cases*
- *Relevant error logs or problem reports*

One thing that is not a test deliverable is the software itself that is listed under test items and is delivered by development. [Insert text here.]

12 REMAINING TEST TASKS

If this is a multi-phase process or if the application is to be released in increments there may be parts of the application that this plan does not address. These areas need to be identified to avoid any confusion should defects be reported back on those future functions. This will also allow the users and testers to avoid incomplete functions and prevent waste of resources chasing non-defects. [Insert text here.]

13 ENVIRONMENTAL NEEDS

Are there any special requirements for this test plan, such as:

- *Special hardware such as simulators, static generators etc.*
- *How will test data be provided. Are there special collection requirements or specific ranges of data that must be provided?*

[Insert text here.]

14 STAFFING AND TRAINING NEEDS

Training on the application/system.

Training for any test tools to be used. [Insert text here.]

15 RESPONSIBILITIES

Who is in charge?

This issue includes all areas of the plan. Here are some examples:

- *Selecting features to be tested and not tested.*
- *Ensuring all required elements are in place for testing.*

[Insert text here.]

16 SCHEDULE

Should be based on realistic and validated estimates. If the estimates for the development of the application are inaccurate, the entire project plan will slip and the testing is part of the overall project plan.

[Insert text here.]

17 PLANNING RISKS AND CONTINGENCIES

What are the overall risks to the project with an emphasis on the testing process? Specify what will be done for various risk events.

[Insert text here.]

18 APPROVALS

Who can approve the process as complete and allow the project to proceed to the next level (depending on the level of the plan)?

[Insert text here.]

19 GLOSSARY

Used to define terms and acronyms used in the document, and testing in general, to eliminate confusion and promote consistent communications.

[Insert text here.]

20 APPENDIX A. [insert name here]

Include copies of test examples, etc. supplied or derived from the customer. Appendices are labeled A, B, . . . n. Reference each appendix as appropriate in the text of the document.

[insert appendix A here]

21 APPENDIX B. [insert name here]

[insert appendix B here]

Chapter 4

IMPLEMENTATION ORGANIZATION (IO) TEMPLATE

Version 1.0, October 2013

FOREWORD

This template was created to provide system and software development projects with a model implementation organization document template. The template is based on Jeffery making things up as he goes along. It has been edited and updated by Dr. Clint Jeffery for use in UI CS 383.

The IO template begins on the next page. Just throw away this page and enter your implementation notes into the following template. Don't forget to change the headers and footers as necessary. The following conventions are used to guide you in developing your IO:

[Text] **Replace** this text with your implementation and organization notes.

text in italics Notes/instructions to the author. **Delete in your finished document.**

IMPLEMENTATION

FOR

[state program/system name here]

Version [[insert version number]]
[[insert date]]

Prepared for:

[state customer name(s) here]

Prepared by:

[insert your name(s)]

University of Idaho

Moscow, ID 83844-1010

CS383 TPD

RECORD OF CHANGES (Change History)

Change Number	Date completed	Location of change (e.g., page or figure #)	A M D	Brief description of change	Approved by (initials)	Date approved

A - ADDED M - MODIFIED D – DELETED

[put program /system name here]

TABLE OF CONTENTS

Section Page

1 IMPLEMENTATION MANIFEST

The manifest provides an overview of the implementation, emphasizing the directory structure and the contents of the various source files. A subsection of this includes the organization of static and dynamic data files such as XML or SQL database(s) and their organization. The goal is to aid developers in navigating around the system, so that they can quickly find the section of code they are interested in.

[Insert text here.]

2 REFERENCES

List all documents that support this implementation description. Refer to the actual version/release number of the document as stored in the configuration management system. Do not duplicate the text from other documents as this will reduce the viability of this document and increase the maintenance effort.

[Insert text here.]

3 INTRODUCTION

Unlike the MANIFEST, which emphasizes files and directories and is similar to a table of contents, this section gives a general introduction to the implementation: languages and tools used, primary architectural components and subsystems.

4 COMPONENTS

The classes or modules of the system are described. This may be organized around topics or subsystems, or it may be alphabetical for reference purposes.

[Insert text here.]

5 APPENDIX A. [insert name here]

Implementation appendices may include full or partial source code, or other supporting material. Appendices are labeled A, B, . . . n. Reference each appendix as appropriate in the text of the document.

[insert appendix A here]

6 APPENDIX B. [insert name here]

[insert appendix B here]

Chapter 5

SOFTWARE METRICS (SM) TEMPLATE

Version 1.0, October 2013

FOREWORD

This template was created to provide system and software development projects with a model software metrics document template. The template is based on Jeffery making things up as he goes along. It has been edited and updated by Dr. Clint Jeffery for use in UI CS 383.

The SM template begins on the next page. Just throw away this page and enter your implementation notes into the following template. Don't forget to change the headers and footers as necessary. The following conventions are used to guide you in developing your SM:

[Text] **Replace** this text with your implementation and organization notes.

text in italics Notes/instructions to the author. **Delete in your finished document.**

METRICS

FOR

[state program/system name here]

Version [[insert version number]]
[[insert date]]

University of Idaho
Open Space. Open Minds.

Prepared for:

[state customer name(s) here]

Prepared by:

[insert your name(s)]

University of Idaho

Moscow, ID 83844-1010

CS383 SMD

RECORD OF CHANGES (Change History)

Change Number	Date completed	Location of change (e.g., page or figure #)	A M D	Brief description of change	Approved by (initials)	Date approved

A - ADDED M - MODIFIED D – DELETED

[put program /system name here]

TABLE OF CONTENTS

Section Page

1 INTRODUCTION

This is the executive summary part of the software metrics document. It should explain in general terms what has been measured for the software system, and summarize what conclusions can be drawn from the resulting measurements.

2 MEASUREMENT ITEMS

These are things you intend to measure within the scope of this measurement plan. Essentially, something you will measure, a list of what is to be measured. This can be developed from the software application inventories as well as other sources of documentation and information.

[Insert text here.]

3 APPROACH

This is your overall strategy for this measurement plan; it should be appropriate to the software being constructed.

- *Are any special tools to be used? What are they?*
- *What metrics will be collected?*

[Insert text here.]

4 ITEM PASS/FAIL CRITERIA

What are the Completion criteria for this plan? This is a critical aspect of any test plan and should be appropriate to the level of the plan. [Insert text here.]

5 SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS

If the number or type of defects reaches a point where the follow on testing has no value, it makes no sense to continue the test; you are just wasting resources.

Specify what constitutes stoppage for a test or series of tests and what is the acceptable level of defects that will allow the testing to proceed past the defects. [Insert text here.]

6 TEST DELIVERABLES

What is to be delivered as part of this plan?

- *Test plan document*
- *Test cases*
- *Relevant error logs or problem reports*

One thing that is not a test deliverable is the software itself that is listed under test items and is delivered by development. [Insert text here.]

7 REMAINING TEST TASKS

If this is a multi-phase process or if the application is to be released in increments there may be parts of the application that this plan does not address. These areas need to be identified to avoid any confusion should defects be reported back on those future functions. This will also allow the users and testers to avoid incomplete functions and prevent waste of resources chasing non-defects. [Insert text here.]

8 ENVIRONMENTAL NEEDS

Are there any special requirements for this test plan, such as:

- *Special hardware such as simulators, static generators etc.*
- *How will test data be provided. Are there special collection requirements or specific ranges of data that must be provided?*

[Insert text here.]

9 STAFFING AND TRAINING NEEDS

Training on the application/system.

Training for any test tools to be used. [Insert text here.]

10 RESPONSIBILITIES

Who is in charge?

This issue includes all areas of the plan. Here are some examples:

- *Selecting features to be tested and not tested.*
- *Ensuring all required elements are in place for testing.*

[Insert text here.]

11 SCHEDULE

Should be based on realistic and validated estimates. If the estimates for the development of the application are inaccurate, the entire project plan will slip and the testing is part of the overall project plan.

[Insert text here.]

12 PLANNING RISKS AND CONTINGENCIES

What are the overall risks to the project with an emphasis on the testing process? Specify what will be done for various risk events.

[Insert text here.]

13 APPROVALS

Who can approve the process as complete and allow the project to proceed to the next level (depending on the level of the plan)?

[Insert text here.]

14 GLOSSARY

Used to define terms and acronyms used in the document, and testing in general, to eliminate confusion and promote consistent communications.

[Insert text here.]

15 APPENDIX A. [insert name here]

Include copies of test examples, etc. supplied or derived from the customer. Appendices are labeled A, B, . . . n. Reference each appendix as appropriate in the text of the document.

[insert appendix A here]

16 APPENDIX B. [insert name here]

[insert appendix B here]

Chapter 6

DELIVERY/INSTALLATION (DI) TEMPLATE

Version 1.0, October 2013

FOREWORD

This template was created to provide system and software development projects with a model delivery and installation document template. The template is based on Jeffery making things up as he goes along. It has been edited and updated by Dr. Clint Jeffery for use in UI CS 383.

The DI template begins on the next page. Just throw away this page and enter your implementation notes into the following template. Don't forget to change the headers and footers as necessary. The following conventions are used to guide you in developing your DI:

[Text] **Replace** this text with your implementation and organization notes.

text in italics Notes/instructions to the author. **Delete in your finished document.**

PROJECT DELIVERY/INSTALLATION (DI)
FOR

[state program/system name here]

Version [[insert version number]]
[[insert date]]

Prepared for:

[state customer name(s) here]

Prepared by:

[insert your name(s)]

University of Idaho

Moscow, ID 83844-1010

CS383 TPD

RECORD OF CHANGES (Change History)

Change Number	Date completed	Location of change (e.g., page or figure #)	A M D	Brief description of change	Approved by (initials)	Date approved

A - ADDED M - MODIFIED D – DELETED

[put program /system name here]

TABLE OF CONTENTS

Section Page

1 TEST PLAN IDENTIFIER

Some type of unique company generated number to identify this test plan, its level and the level of software that it is related to. Preferably the test plan level will be the same as the related software level. The number may also identify whether the test plan is a Master plan, a Level plan, an integration plan or whichever plan level it represents. This is to assist in coordinating software and testware versions within configuration management.

[Insert text here.]

2 REFERENCES

List all documents that support this test plan. Refer to the actual version/release number of the document as stored in the configuration management system. Do not duplicate the text from other documents as this will reduce the viability of this document and increase the maintenance effort.

[Insert text here.]

3 INTRODUCTION

State the purpose of the Plan, possibly identifying the level of the plan (master etc.). This is essentially the executive summary part of the plan.

4 TEST ITEMS

These are things you intend to test within the scope of this test plan. Essentially, something you will test, a list of what is to be tested. This can be developed from the software application inventories as well as other sources of documentation and information.

[Insert text here.]

5 SOFTWARE RISK ISSUES

Identify what software is to be tested and what the critical areas are, such as:

- *1. Delivery of a third party product.*

- *2. New version of interfacing software*

- *3. Ability to use and understand a new package/tool, etc.*
- *4. Extremely complex functions*
- *5. Modifications to components with a past history of failure*
- *6. Poorly documented modules or change requests*

[Insert text here.]

6 FEATURES TO BE TESTED

This is a listing of what is to be tested from the USERS viewpoint of what the system does. This is not a technical description of the software, but a USERS view of the functions.

[Insert text here.]

7 FEATURES NOT TO BE TESTED

This is a listing of what is NOT to be tested from both the Users viewpoint of what the system does and a configuration management/version control view. This is not a technical description of the software, but a USERS view of the functions.

[Insert text here.]

8 APPROACH

This is your overall test strategy for this test plan; it should be appropriate to the level of the plan (master, acceptance, etc.) and should be in agreement with all higher and lower levels of plans. Overall rules and processes should be identified.

- *Are any special tools to be used? What are they?*
- *What metrics will be collected for this test?*
- *How many configurations/platforms are to be tested?*
- *How will elements in the design deemed "untestable" be processed?*

[Insert text here.]

9 ITEM PASS/FAIL CRITERIA

What are the Completion criteria for this plan? This is a critical aspect of any test plan and should be appropriate to the level of the plan. [Insert text here.]

10 SUSPENSION CRITERIA AND RESUMPTION REQUIREMENTS

If the number or type of defects reaches a point where the follow on testing has no value, it makes no sense to continue the test; you are just wasting resources.

Specify what constitutes stoppage for a test or series of tests and what is the acceptable level of defects that will allow the testing to proceed past the defects. [Insert text here.]

11 TEST DELIVERABLES

What is to be delivered as part of this plan?

- *Test plan document*
- *Test cases*
- *Relevant error logs or problem reports*

One thing that is not a test deliverable is the software itself that is listed under test items and is delivered by development. [Insert text here.]

12 REMAINING TEST TASKS

If this is a multi-phase process or if the application is to be released in increments there may be parts of the application that this plan does not address. These areas need to be identified to avoid any confusion should defects be reported back on those future functions. This will also allow the users and testers to avoid incomplete functions and prevent waste of resources chasing non-defects. [Insert text here.]

13 ENVIRONMENTAL NEEDS

Are there any special requirements for this test plan, such as:

- *Special hardware such as simulators, static generators etc.*
- *How will test data be provided. Are there special collection requirements or specific ranges of data that must be provided?*

[Insert text here.]

14 STAFFING AND TRAINING NEEDS

Training on the application/system.

Training for any test tools to be used. [Insert text here.]

15 RESPONSIBILITIES

Who is in charge?

This issue includes all areas of the plan. Here are some examples:

- *Selecting features to be tested and not tested.*
- *Ensuring all required elements are in place for testing.*

[Insert text here.]

16 SCHEDULE

Should be based on realistic and validated estimates. If the estimates for the development of the application are inaccurate, the entire project plan will slip and the testing is part of the overall project plan.

[Insert text here.]

17 PLANNING RISKS AND CONTINGENCIES

What are the overall risks to the project with an emphasis on the testing process? Specify what will be done for various risk events.

[Insert text here.]

18 APPROVALS

Who can approve the process as complete and allow the project to proceed to the next level (depending on the level of the plan)?

[Insert text here.]

19 GLOSSARY

Used to define terms and acronyms used in the document, and testing in general, to eliminate confusion and promote consistent communications.

[Insert text here.]

20 APPENDIX A. [insert name here]

Include copies of test examples, etc. supplied or derived from the customer. Appendices are labeled A, B, . . . n. Reference each appendix as appropriate in the text of the document.

[insert appendix A here]

21 APPENDIX B. [insert name here]

[insert appendix B here]