

Software Requirements Specification Template

6 January 2014

The following annotated template is intended to show the structure of an IEEE-830 compliant SRS document.

Template Usage:

Text contained within angle brackets ('<', '>') shall be replaced by your project-specific information and/or details. For example, <Project Name> will be replaced with either 'New Accounting System' or 'New Web Site'.

Italicized text is included to briefly annotate the purpose of each section within this template. This text should not appear in the final version of your submitted SRS.

This cover page is not a part of the final template and should be removed before your SRS is submitted.

<Project Name>

Software Requirements Specification

<Version>

<Date>

<Your Name>

Lead Software Engineer

Prepared for

<Project Name>

Revision History

Date	Description	Author	Comments
<date>	<Version 1>	<Your Name>	<First Revision>

Document Approval

The following Software Requirements Specification has been accepted and approved by the following:

Signature	Printed Name	Title	Date
	<Your Name>		

Table of Contents

REVISION HISTORY	II
DOCUMENT APPROVAL	II
1. INTRODUCTION	1
1.1 PURPOSE	1
1.2 SCOPE	1
1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS	1
1.4 REFERENCES	1
1.5 OVERVIEW	1
2. GENERAL DESCRIPTION	2
2.1 PRODUCT PERSPECTIVE	2
2.2 PRODUCT FUNCTIONS	2
2.3 USER CHARACTERISTICS	2
2.4 GENERAL CONSTRAINTS	2
2.5 ASSUMPTIONS AND DEPENDENCIES	2
3. SPECIFIC REQUIREMENTS	2
3.1 EXTERNAL INTERFACE REQUIREMENTS	3
3.1.1 <i>User Interfaces</i>	3
3.1.2 <i>Hardware Interfaces</i>	3
3.1.3 <i>Software Interfaces</i>	3
3.1.4 <i>Communications Interfaces</i>	3
3.2 FUNCTIONAL REQUIREMENTS	3
3.2.1 <Functional Requirement or Feature #1>	3
3.2.2 <Functional Requirement or Feature #2>	3
3.3 USE CASES	3
3.3.1 <i>Use Case #1</i>	3
3.3.2 <i>Use Case #2</i>	3
3.4 CLASSES / OBJECTS	3
3.4.1 <Class / Object #1>	3
3.4.2 <Class / Object #2>	3
3.5 NON-FUNCTIONAL REQUIREMENTS	4
3.5.1 <i>Performance</i>	4
3.5.2 <i>Reliability</i>	4
3.5.3 <i>Availability</i>	4
3.5.4 <i>Security</i>	4
3.5.5 <i>Maintainability</i>	4
3.5.6 <i>Portability</i>	4
3.6 INVERSE REQUIREMENTS	4
3.7 DESIGN CONSTRAINTS	4
3.8 LOGICAL DATABASE REQUIREMENTS	4
3.9 OTHER REQUIREMENTS	4
4. ANALYSIS MODELS	4
4.1 SEQUENCE DIAGRAMS	5
4.3 DATA FLOW DIAGRAMS (DFD)	5
4.2 STATE-TRANSITION DIAGRAMS (STD)	5
5. CHANGE MANAGEMENT PROCESS	5
A. APPENDICES	5

<Project Name>

A.1 APPENDIX 1.....	5
A.2 APPENDIX 2.....	5

1. Introduction

The introduction to the Software Requirement Specification (SRS) document should provide an overview of the complete SRS document. While writing this document please remember that this document should contain all of the information needed by a software engineer to adequately design and implement the software product described by the requirements listed in this document. (Note: the following subsection annotations are largely taken from the IEEE Guide to SRS).

1.1 Purpose

What is the purpose of this SRS and the (intended) audience for which it is written.

1.2 Scope

This subsection should:

- (1) Identify the software product(s) to be produced by name; for example, Host DBMS, Report Generator, etc*
- (2) Explain what the software product(s) will, and, if necessary, will not do*
- (3) Describe the application of the software being specified. As a portion of this, it should:*
 - (a) Describe all relevant benefits, objectives, and goals as precisely as possible. For example, to say that one goal is to provide effective reporting capabilities is not as good as saying parameter-driven, user-definable reports with a 2 h turnaround and on-line entry of user parameters.*
 - (b) Be consistent with similar statements in higher-level specifications (for example, the System Requirement Specification) , if they exist. What is the scope of this software product.*

1.3 Definitions, Acronyms, and Abbreviations

This subsection should provide the definitions of all terms, acronyms, and abbreviations required to properly interpret the SRS. This information may be provided by reference to one or more appendixes in the SRS or by reference to other documents.

1.4 References

This subsection should:

- (1) Provide a complete list of all documents referenced elsewhere in the SRS, or in a separate, specified document.*
 - (2) Identify each document by title, report number - if applicable - date, and publishing organization.*
 - (3) Specify the sources from which the references can be obtained.*
- This information may be provided by reference to an appendix or to another document.*

1.5 Overview

This subsection should:

- (1) Describe what the rest of the SRS contains*

(2) Explain how the SRS is organized.

2. General Description

This section of the SRS should describe the general factors that affect the product and its requirements. It should be made clear that this section does not state specific requirements; it only makes those requirements easier to understand.

2.1 Product Perspective

This subsection of the SRS puts the product into perspective with other related products or projects. (See the IEEE Guide to SRS for more details).

2.2 Product Functions

This subsection of the SRS should provide a summary of the functions that the software will perform.

2.3 User Characteristics

This subsection of the SRS should describe those general characteristics of the eventual users of the product that will affect the specific requirements. (See the IEEE Guide to SRS for more details).

2.4 General Constraints

This subsection of the SRS should provide a general description of any other items that will limit the developer's options for designing the system. (See the IEEE Guide to SRS for a partial list of possible general constraints).

2.5 Assumptions and Dependencies

This subsection of the SRS should list each of the factors that affect the requirements stated in the SRS. These factors are not design constraints on the software but are, rather, any changes to them that can affect the requirements in the SRS. For example, an assumption might be that a specific operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not available, the SRS would then have to change accordingly.

3. Specific Requirements

This will be the largest and most important section of the SRS. The customer requirements will be embodied within Section 2, but this section will give the D-requirements that are used to guide the project's software design, implementation, and testing.

Each requirement in this section should be:

- *Correct*
- *Traceable (both forward and backward to prior/future artifacts)*
- *Unambiguous*
- *Verifiable (i.e., testable)*
- *Prioritized (with respect to importance and/or stability)*

<Project Name>

- *Complete*
- *Consistent*
- *Uniquely identifiable (usually via numbering like 3.4.5.6)*

Attention should be paid to the carefully organize the requirements presented in this section so that they may easily accessed and understood. Furthermore, this SRS is not the software design document, therefore one should avoid the tendency to over-constrain (and therefore design) the software project within this SRS.

3.1 External Interface Requirements

3.1.1 User Interfaces

3.1.2 Hardware Interfaces

3.1.3 Software Interfaces

3.1.4 Communications Interfaces

3.2 Functional Requirements

This section describes specific features of the software project. If desired, some requirements may be specified in the use-case format and listed in the Use Cases Section.

3.2.1 <Functional Requirement or Feature #1>

3.2.1.1 Introduction

3.2.1.2 Inputs

3.2.1.3 Processing

3.2.1.4 Outputs

3.2.1.5 Error Handling

3.2.2 <Functional Requirement or Feature #2>

...

3.3 Use Cases

3.3.1 Use Case #1

3.3.2 Use Case #2

...

3.4 Classes / Objects

3.4.1 <Class / Object #1>

3.4.1.1 Attributes

3.4.1.2 Functions

<Reference to functional requirements and/or use cases>

3.4.2 <Class / Object #2>

...

3.5 Non-Functional Requirements

Non-functional requirements may exist for the following attributes. Often these requirements must be achieved at a system-wide level rather than at a unit level. State the requirements in the following sections in measurable terms (e.g., 95% of transaction shall be processed in less than a second, system downtime may not exceed 1 minute per day, > 30 day MTBF value, etc).

3.5.1 Performance

3.5.2 Reliability

3.5.3 Availability

3.5.4 Security

3.5.5 Maintainability

3.5.6 Portability

3.6 Inverse Requirements

*State any *useful* inverse requirements.*

3.7 Design Constraints

Specify design constraints imposed by other standards, company policies, hardware limitation, etc. that will impact this software project.

3.8 Logical Database Requirements

Will a database be used? If so, what logical requirements exist for data formats, storage capabilities, data retention, data integrity, etc.

3.9 Other Requirements

Catchall section for any additional requirements.

4. Analysis Models

List all analysis models used in developing specific requirements previously given in this SRS. Each model should include an introduction and a narrative description. Furthermore, each model should be traceable the SRS's requirements.

4.1 Sequence Diagrams

4.3 Data Flow Diagrams (DFD)

4.2 State-Transition Diagrams (STD)

5. Change Management Process

Identify and describe the process that will be used to update the SRS, as needed, when project scope or requirements change. Who can submit changes and by what means, and how will these changes be approved.

A. Appendices

Appendices may be used to provide additional (and hopefully helpful) information. If present, the SRS should explicitly state whether the information contained within an appendix is to be considered as a part of the SRS's overall set of requirements.

Example Appendices could include (initial) conceptual documents for the software project, marketing materials, minutes of meetings with the customer(s), etc.

A.1 Appendix 1

A.2 Appendix 2