

# Comparison Of Various Software Requirements Specification Formats



## IT350: Software Engineering Assignment 2

**Course Mentor : Biju R Mohan**

**Course Instructor : Ms. Raksha R Nadigar**

**Submitted by:**

**Abhishek S**

**15IT202 3<sup>rd</sup> year IT**

**16<sup>th</sup> January, 2018**

**National Institute Of Technology Karnataka**

# IEEE Format

<b>Table of Contents.....</b>	<b>1</b>
<b>1.Introduction.....</b>	<b>3</b>
1.1 Purpose.....	3
1.2 Document Conventions.....	3
1.3 Intended Audience and Reading Suggestion.....	3
1.4 Product Scope.....	3
<b>2. Overall Description.....</b>	<b>4</b>
2.1 Product Perspective.....	4
2.2 Product Functions.....	4
2.3 User Classes and Characteristics.....	4
2.4 Operating Environment.....	4
2.5 Design and Implementation Constraints.....	5
2.6 Assumption and Dependencies.....	5
<b>3. External Interface Requirement.....</b>	<b>5</b>
3.1 User Interface.....	5
3.2 Hardware Interface.....	5
3.3 Software Interface.....	6
3.4 Communication Interfaces.....	6
<b>4. System Features.....</b>	<b>6</b>
<b>5. Other Non functional Requirements.....</b>	<b>8</b>
5.1 Performance Requirements.....	8
5.2 Safety Requirements.....	8
5.3 Security Requirements.....	8
5.4 Software Quality Attributes.....	8
<b>6. Other Requirements.....</b>	<b>9</b>
Appendix A: Glossary.....	9
Appendix B: Analysis Model.....	9
Appendix C: To be determined list.....	9

## Pros:

1. A very detailed explanation of each and every concept.
2. To the point detailing, no extra bits
3. Useful for developers to get a good idea about the proceedings of the project.

4. Any changes made can be easily recorded via the "Revision Table".
5. The document evolves as the project starts to be implemented.
6. A standard document convention.
7. Client gets an idea of how the user interface may look.

## Cons:

1. Many requirements cannot be specified in the beginning by the client.
2. It is a rigorous task to complete the SRS.
3. As it is an evolving document, there will be many changes made.
4. "To be determined list" sometimes remains unfulfilled and might be bothersome.

# Format 2

<b>1. Introduction</b>	<b>4</b>
1.1 Purpose	4
1.2 Scope	4
1.3 Definitions, Acronyms and Abbreviations	4
1.4 References	5
1.5 Technologies to be used	5
1.6 Overview	5
<b>2. Overall Description</b>	<b>6</b>
2.1 Use-Case Model Survey	6
2.2 WEB Architecture diagram	8
2.3 ER Diagram	9
2.4 Architecture diagram	10
2.5 Data Dictionary	11
2.6 Assumptions and Dependencies	15
<b>3. Specific Requirements</b>	<b>16</b>
3.1 Use-Case Reports	19
3.2 Class Diagram	29
3.2 Supplementary Requirements	31

## Pros:

1. A very simple format
2. Use case, class diagrams help in a better understanding of the project

## Cons:

1. Maybe a too simple SRS.
2. No mention about user documentation.
3. No revision history maintained.
4. Too little or subtle explanation of things.

# Format 3

<b>1. Introduction</b>	<b>5</b>
1.1 Purpose	5
1.2 Scope	5
1.3 Definitions, Acronyms, and Abbreviations	6
1.4 References	6
1.5 Overview	6
<b>2. Overall Description</b>	<b>6</b>
<b>3. Specific Requirements</b>	<b>7</b>
3.1 Functionality	7
3.2 Usability	11
3.3 Reliability & Availability	11
3.4 Performance	12
3.5 Security	12
3.6 Supportability	13
3.7 Design Constraints	13
3.8 On-line User Documentation and Help System Requirements	13
3.9 Purchased Components	13
3.10 Interfaces	14
3.10.1 User Interfaces	14
3.10.2 Hardware Interfaces	14
3.10.3 Software Interfaces	14
3.10.4 Communications Interfaces	15
3.11 Licensing Requirements	15
3.12 Legal, Copyright, and Other Notices	15
3.13 Applicable Standards	15
<b>4. Supporting Information</b>	<b>15</b>

## Pros:

1. Simple and easy to make.
2. The copyright, license and all legal formalities get a mentioning in the SRS.

## Cons:

1. No UML, USE CASE diagrams which are helpful for the developers.

# Format 4

- 1. Purpose**
- 2. Scope**
- 3. Product Perspective**
  - 3.1. System Interfaces
  - 3.2. User Interfaces
  - 3.3. Hardware Interfaces
  - 3.4. Software Interfaces
  - 3.5. Communications Interfaces
  - 3.6. Memory Constraints
  - 3.7. Operations
  - 3.8. Site Adaptation Requirements
- 4. Product Functions**
- 5. User Characteristics**
- 6. Limitations**
- 7. Assumptions And Dependencies**
- 8. Apportioning Of Requirements**
- 9. Specific Requirements**
- 10. External Interfaces**
- 11. Functions**
- 12. Usability Requirements**
- 13. Performance Requirements**
- 14. Verification**
- 15. Supporting Information**

## Pros:

1. Standards for qualification of the software is mentioned.
2. "Site adaptation Requirements" ensures that the final stable software is compatible across all devices.
3. Has information on memory constraints.

## Cons:

1. Hard to write all the details and is complicated too.
2. Client cannot specify all details in the initial stages of the software development.

# Format 5

## **1. Introduction**

- 1.1 Purpose
- 1.2 Document Conventions
- 1.3 Intended Audience and Reading Suggestions
- 1.4 Project Scope

## **2. Overall Description**

- 2.1 Product Perspective
- 2.2 Product Features
- 2.3 User Classes and Characteristics
- 2.4 Operating Environment
- 2.5 Design and Implementation Constraints
- 2.6 User Documentation
- 2.7 Assumptions and Dependencies

## **3. System Features**

Core Features

Additional Features

## **4. External Interface Requirements**

- 4.1 User Interface
- 4.2 Hardware Interfaces
- 4.3 Software Interfaces
- 4.4 Communications Interfaces

## **5. Other Nonfunctional Requirements**

- 5.1 Performance Requirements
- 5.2 Safety Requirements
- 5.3 Security Requirements
- 5.4 Software Quality Attributes

## **6. Key Milestones**

## **7. Key Resource Requirements**

## **8. Other Requirements**

## **9. Appendix A Glossary**

## **10. Appendix B Project Proposal**

## Pros:

- 1. System feature contains subjects as core features and additional features which help in better understanding of the software.

2. "Key milestones" can be compared to time periods where the client will be shown the progress of the software.
3. Less complicated.
4. "Project proposal" gives an overall view of how the end product will be.
5. "Software quality attributes" ensures the quality of end product to the clients.

## Cons:

1. The developers may not keep up to the "Key milestones" promise and may cause issues.
2. Client can't specify all the requirements in the beginning.



