Comparison of Various Software Requirement Specification Formats based on Table of Contents

-Varshitha M K (15IT151)

Format 1(IEEE)

Table of Contents
Revision History

1. Introduction
- 1.1 Purpose
- 1.2 Document Conventions
- 1.3 Intended Audience and Reading Suggestions
- 1.4 Product Scope
- 1.5 References

2. Overall Description
- 2.1 Product Perspective
- 2.2 Product Functions
- 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. External Interface Requirements
- 3.1 User Interfaces
- 3.2 Hardware Interfaces
- 3.3 Software Interfaces
- 3.4 Communications Interfaces

4. System Features
- 4.1 System Feature 1
- 4.2 System Feature 2

5. Other Nonfunctional Requirements
- 5.1 Performance Requirements
- 5.2 Safety Requirements
- 5.3 Security Requirements
- 5.4 Software Quality Attributes
- 5.5 Business Rules

6. Other Requirements
Appendix A: Glossary
Appendix B: Analysis Models
Appendix C: To Be Determined List

Advantages

1. Very detailed and concise
2. Useful for the developers to get a good idea of what the client wants to see in the final product
3. There is a provision to add requirements later by including them in the “To Be Discussed” section
4. Evolving documentation
5. Follows a standard document convention that has to be followed by all the maintainers

Disadvantages

1. It is hard for the client to specify some requirements (like communication interfaces) in such detail at the beginning phase

2. Many sections may be marked under “To Be Discussed” remain unfilled during the initial phases
3. Takes a considerable amount of time to prepare
4. SRS will undergo many revisions due to refinement of requirements
5. Less focus on UML/class diagrams, though they help the developer visualize the final product

Reference: IEEE SRS https://web.cs.dal.ca/~hawkey/3130/srs_template-ieee.doc

Format 2 (TCS)

TABLE OF CONTENTS

1Scope

2 Referenced Documents

3 Description of Software to be Developed

4 Development Sequence and Schedule

4.1 Requirements Analysis

4.2 Software Specification Review (PDR)

4.3 Software Design

4.4 Software Critical Design Review (CDR)

4.5 Software Coding and Debug

4.6 Software Code Reviews

4.7 Module Testing

4.8 Software Testing

4.9 Integrated HW and SW Testing

4.10 Subsystem Commissioning and Integration

4.11 Software handover

5 Software Safety

5.1 Safety certificate

5.2 Communication Integrity

5.3 Initialisation

5.4 Start-up and Shut Down Procedure

6 Generic Software Requirements

6.1 Naming and Tagging Conventions

6.2 Remote Initialisation

6.3 Data

6.4 Software cyclic execution

6.5 Data time stamping

6.6 Modular Design

6.7 Measuring Units

6.8 Data resolution

6.9 I/O Validation

6.10 Synchronisation

6.11 Unused Code

6.12 Software Comments

6.13 Self-changing code

6.14 Manual Operation

6.15 Communication methods

7 Specific TCS Software Requirements

7.1 Operating Systems

7.2 Development Software

7.3 Application Software

8.4 Man-Machine Interfaces

8 Deliverables

9 Configuration Control

Advantages

1. More Descriptive format
2. Approval sheets give more description of the project designer and one who has approved the product
3. Code Debugging, code testing, software testing methods are briefly described

Disadvantages

1. Too complex to fill all the details.
2. Non-functional requirements are not clearly specified.
3. No mention about user documentation.

Reference http://www.sal.wisc.edu/PFIS/docs/rss-vis/archive/protected/salt/1700BP0009%20TCS%20SW%20Plan%20C.pdf

Format 3 (IBM)

1. **Introduction**
   - 1.1 Purpose
   - 1.2 Scope
   - 1.3 Definitions, Acronyms, and Abbreviations
   - 1.4 References
   - 1.5 Overview
2. **Overall Description**
   - 2.1 Use case model survey
   - 2.2 ER Diagram
   - 2.3 Architecture Diagram
   - 2.4 Data Dictionary
   - 2.5 Assumptions and data dependency
3. **Specific Requirements**
   - 3.1 Use-Case Reports
   - 3.2 Class Diagram
   - 3.3 Supplementary Requirements

Advantages

1. Simplest format
2. Use-case/class/ER/ Architecture diagrams aid the software developers
3. Includes Legal information and standards, not mentioned in others

Disadvantages

1. May be too simple
2. Supplementary requirements might mix up functional and non-functional requirements
3. No mention about user documentation

Reference: IBM SRS
https://www.ibm.com/developerworks/.../files/.../document/...*/SRS_Sample.doc*

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**

Advantages

1. Has extra details about memory constraints and special operations by the user
2. The “Apportioning” section is a helpful addition as it maps the requirements to the software that will carry out the given operations to satisy the requirement
3. “Site adaptation requirements” help in ensuring the end product will be compatible with more systems by specifying any data or initialization sequences that are specific to a given site
4. “Usability requirements” include measurable effectiveness, efficiency, and satisfaction criteria in specific contexts of use.
5. Provides the verification approaches and methods planned to qualify the software.
6. Supporting info contains : Sample input/output formats, descriptions of cost analysis studies, or results of user surveys; A description of the problems to be solved by the software; Special packaging instructions for the code and the media to meet security, export, initial loading.

Disadvantages

1. Much more complicated and time consuming to create. No user documentation.
2. Harder for the client to specify such requirements detail at the beginning phase (more revisions)

Reference: International standard for custom SRS- https://belitsoft.com/php-development-services/software-requirements-specificaion-document-example-international-standard

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

4. External Interface Requirements

4.1 User Interface

4.2 Hardware Interfaces

4.3 Software Interfaces

4.4 Communications Interfaces

5. Other Non-functional 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

Advantages

1. Less complicated than Format 2 to create
2. “Key Milestones” give the client a good idea of what to expect and when. This avoids unrealistic expectations
3. “Project Proposal” provides a complete summary of the given problem statement and what the end product will be capable of
4. System features are split into core and additional so that developers can assign priorities.
5. Key Resource requirements divide the huge problem into manageable logical chunks and state the necessary required expertise, internal/external resources to satisfy the requirements and the associated constraints

Disadvantages

1. Many revisions can still arise.
2. Harder for the client to specify such requirements in such detail at the beginning phase

Reference: https://www.cise.ufl.edu/class/cen3031sp13/SRS_Example_1_2011.pdf