Development Process of the Software Requirement Specification for the Design of COMPAS

Sanghoon Lee, Daeyou Park, Hyungbum Suh*, Sungmin Kim*, Jinil Kim**, Choongsub Yeom
Institute for Advanced Engineering, Yongin P.O.Box 25, Gyeonggi-Do 449-863, shoon@iae.re.kr
** Wolsong Nuclear Power Division, Korea Hydro & Nuclear Power*
*** SE Technologies, Yongin P.O.Box 25, Gyeonggi-Do 449-863*

1. Introduction

For automating the core management works of the Wolsong NPP, CANDU COre Management Procedures Automation System (COMPAS) is developing. The success of the development of COMPAS is mainly dependent on identifying system functionality needs from the user's prospective. This means that the user requirements must be fully discovered and implemented in COMPAS. For this purpose, we developed Software Requirement Specification (SRS) for design and implementation of COMPAS. In this paper, we introduced the process for developing SRS and the resulting skeleton of SRS used in the design of COMPAS.

2. Importance of Requirement

Many software projects have revealed the following facts.

-. It need much more work to fix the errors in the late step of the development.

Stage	Relative Repair Cost
Requirements	1-2
Design	5
Coding	10
Unit Test	20
System Test	50
Maintenance	200

Figure 1. Relative cost to fix errors..

- Many errors are discovered during testing[1].
- Many errors are due to the incorrect requirements[2].

These facts mean that good requirements can save much time and effort in the development of the software.

For all that, many software projects start modeling or even implementing without defining requirements.

3. SRS Development Process

There is no best way to develop the SRS. But there are some partial methods to elicit requirements, such as apprenticing, introspection, survey/questionnaire, brainstorming, card Sorting, focusing group, functional analysis, operational concept, prototyping, quality function deployment etc[3]. These methods should be used appropriately according to the characteristics of the project and the participants. The COMPAS project team is composed of the software engineer group and the user group. So we developed the following process for the development of the SRS of the COMPAS.

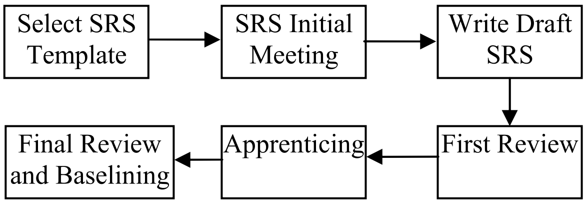

Figure 2. SRS Development Process.

2.1 Select SRS Template

It is very hard to write requirements in a white sheet. So many organizations have developed requirement specification template. Such ANSI/IEEE, European Space Agency, US DoD. We adopted IEEE 830 because of its popularity. Its skeleton is as table 1.

Table 1. IEEE 830 template skeleton.

Chap 1. Introduction	1.1 Purpose 1.2 Scope 1.3 Definitions, Acronyms and Abbreviations 1.4 References 1.5 Overview
Chap. 2 General description	2.1 Product perspective 2.2 Product function summary 2.3 User characteristics 2.4 General constraints 2.5 Assumptions and dependencies
Chap 3. Specific Requirements	- Functional requirements - External interface requirements - Performance requirements - Design constraints - Attributes eg. security, availability, maintainability, transferability/conversion - Other requirements
Appendices	

2.2 SRS Initial Meeting

The purpose of the SRS Initial Meeting is to introduce to every participant how the project will develop the SRS and the role of each participant. The SRS template and the SRS development process are introduced to every participant. It is very important to explain the importance of requirement to the team members, because there can be some one that the requirement development process is an additional job that has little benefit to the project. But as shown in figure 1, a well-defined requirement can save much effort in the later phase of the development.

2.3 Write Draft SRS

We wrote the draft SRS by two steps. The first step was the introduction of the works to be automated by COMPAS to the software engineers by the user group who are the engineers at the Wolsong NPP. Through the first step the software engineers could understand what the COMPAS should do for the users. The second step was to fill the SRS template. The software engineer group filled the SRS by analyzing the work procedure and the supporting materials provided by the user group. In writing SRS, we kept in mind that the requirement should be unambiguous, complete, verifiable, consistent, modifiable, traceable, and usable during the operation and maintenance phase. Especially for functional requirements, we used following template.

Table 2. Functional requirement template

Purpose				
Input Data	Source	Name	Type	Address
Processing				
Output	Destination	Output Name		
Coping with Error				

In this step, many questions have been found. This means that the requirement is evolving to the real requirement.

2.4 First Review

At the first review, the user group have answered to the questions issued by the software engineers during the draft SRS. Through the answers the software engineers could get closer to the real requirements of the user group. Through the first review, the unknown requirements was found and misunderstand was corrected.

2.5 Apprenticing

To understand the work of the user group, we apprenticed to the user group. By apprenticing the user group could explain their work more clearly, and the software engineers could see the real work to be automated and could refine requirements.

2.6 Final Review and Baselining

The draft SRS has been modified through the first review and apprenticing. The user group reviewed the revised SRS and the SRS is formalized served as the baseline for the next step of the development, modeling.

After the final review the SRS should be under change control. So we made SRS change procedure.

Finally we developed SRS, that is enough to proceed to the next step of development, the modeling phase. We tailored the IEEE -830 template for convenience. Table 3 shows the contents of the SRS of the COMPAS.

Table 3. COMPAS SRS skeleton

Chap 1. Introduction	1.1 Purpose 1.2 Scope 1.3 Definitions, Acronyms and Abbreviations 1.4 References 1.5 Overview
Chap. 2 General description	2.1 System Interface and Operation 2.2 System Function 2.3 System Architecture 2.4 Constraints 2.5 Assumptions and Dependencies
Chap. 3. Specific Requirements	3.1 Fuel Management 3.2 Reactor Safety Management 3.3 Instrument Management 3.4 Supporting Functions
Chap. 4 Performance and security	4.1 System Performance 4.2 Reliability 4.3 Security
Chap. 5 Glossary and Abbreviation	
Chap.6 Appendices	

4. Conclusion

Through the SRS development process, many requirements were communicated, modified and finally documented systematically. Although it is very hard to quantify the saving of the efforts by the SRS, we believe and feel that we have saved much effort already. And we can proceed to the next step of the development with little uncertainty.

There remain some open issues. But the point is that we identified the issue and may close it later. The risk that is not identified yet may be the most dangerous point in the project.

REFERENCES

[1] Boehm, B. W. et al. Some Experience with Automated Aids to the Design of Large-Scale Reliable Software, IEEE Trans. On Software Engineering 1, 1, 1975 : 125-133

[2] Tavolato, D. and K. Vincena. A Prototyping Methodology and Its Tool, In Approaches to Prototyping, Springer-Verlag 1984, pp. 434-446

[3] Harold J. Jeydt, Tools for Requirement Discovery, Creation and Elicitation, INCOSE Proceeising, 2002.

[4]IEEE-SA Standard Board, IEEE Std 830-1998 – IEEE Recommended Practice for Software Requirement Specification, 1998