Improving the Planning and Requirements Definition Process

Yukako Iimura, Setsuo Yamada, and Nobuyuki Kobayashi

Abstract

The first process in software development is the planning and requirements definition. In this process, the stakeholder requirements must be organized, and the requirements for implementing the software must be clarified. However, many issues arise with this process in practice, and improvement is therefore needed. In this article, we introduce our activities related to improving the planning and requirements definition process, including studying trends in research and practice.

Keywords: software development, requirements definition, planning

1. Introduction

A survey by the Information-technology Promotion Agency (IPA) [1] found that although achievement levels for quality, cost, and delivery (QCD) targets for software development projects are gradually improving, a number of problems still remain.

To solve these problems, we have been focusing on improving the planning and requirements definition process of software development projects. This is the phase in which it is determined what is to be achieved by the software and what return on investment can be obtained.

The IPA survey also investigated the causes when QCD objectives were not met. It found that, from the developer perspective, 35% or more of the projects had problems at or before requirements definition, and from the user perspective, this figure was approximately 90%. In other words, many developers and users recognized problems in the planning and requirements definition process.

Generally, problems that occur in this process are due to differences between what the users really need and what the developers think they are supposed to build. If developers develop software based on a misunderstood view of what users need, the developed software will not meet user needs. As a result, all of the work up to the point where the difference was detected must be reworked. Several researchers have pointed out that the cost of this rework increases rapidly as the time it takes to discover the error increases. A recent survey [2] reported that the cost of fixing a requirement error discovered in the system operation phase costs 68 to 110 times more than what it would cost if discovered during the planning and requirements definition process (see **Table 1**).

Table 1. Relative cost to correct a requirement defect depending on when it is discovered.

Development phase	Relative cost to correct a defect
Requirements	Base value of 1
Design	2 to 3 times higher (than base value)
Coding	5 to 10 times higher
Testing	8 to 20 times higher
Operation	68 to 110 times higher

2. Challenges of the planning and requirements definition process

Here, we discuss the various challenges that arise in the planning and requirements definition process.

(1) Diversity of stakeholders

The term *stakeholders* refers to the group of people who are involved in a software development project or are affected by the project. Generally, the stakeholders in such a project can be divided into users and developers. Users include end-users who will be using the software, and others such as the information technology department and management, for example, the chief information officer. Similarly, developers include software engineers, vendor project managers, and management, for example, the chief technology officer. The perspectives of each of these stakeholders are different, and they often have conflicting interests. Because of this, stakeholders sometimes have contradictory requirements, and these must be adjusted in order to reach agreement on what will be achieved with the software. Thus, as the number of stakeholders increases, dealing with the contradictory requirements in this way becomes more difficult.

(2) Changing requirements

In the planning and requirements definition process, requirements from stakeholders are adjusted, and agreement is reached on the function and performance of the software. However, these functions and/or performance may need to change in the requirements process due to omissions in the requirements or later changes in the environment. Handling such changes often results in scheduling delays or degraded quality. Also, adding requirements causes wide-ranging effects if they conflict with other existing requirements, as the existing requirements then need to be re-examined, and adjustments need to be made with the relevant stakeholders.

(3) Ambiguity in requirements documentation

Because the purpose of planning and requirements definition is to reach agreement among stakeholders, documentation of these requirements is extremely important. However, if requirements definition is ambiguous, multiple readers of a requirement may have a different understanding of what it means, or they may be confused as a result of interpreting a requirements statement in different ways.

If the stakeholders have differing expectations regarding the function that the software will have due to such ambiguity in the documentation, reworking will be necessary. For example, even if the developers follow the requirements definition, they may understand them in a way that is different from the users' intentions because of the ambiguous specifications, and all of the work starting from the design phase will need to be reworked.

3. Standards and guidelines related to the planning and requirements definition process

Interest in the planning and requirements definition process has increased recently, and consequently, there has been an increase in the number of related standards and guidelines being published. This section describes the Requirements Engineering Body of Knowledge (REBOK) [3][*1]—a requirements engineering knowledge system, ISO/IEC/IEEE[*2] 29148 [4]—a standard related to requirements engineering, and the Software Life Cycle Process Japan Common Frame (SLCP-JCF) [5].

3.1 REBOK

The first edition of REBOK was released by the Japan Information Technology Services Industry Association (JISA) in 2011. It provides a set of common knowledge shared and used by stakeholders consisting of both users and developers, and for both enterprise/business software systems and embedded software systems, except for domain specific knowledge. The requirements engineering process is shown in **Fig. 1**. It configures the four key knowledge areas of (1) elicitation, (2) analysis, (3) specification, and (4) verification, validation, and evaluation in an incremental and iterative way. It aims to provide appropriate knowledge to all the stakeholders involved in the requirements engineering process at appropriate levels of expertise; the stakeholders include corporate management, end-users, project managers, and software developers.

3.2 ISO/IEC/IEEE 29148: 2011

ISO/IEC/IEEE 29148: 2011 covers processes and products related to the engineering of requirements for systems and software products and services throughout their life cycle. It provides additional guidance in the application of requirements engineering and management processes for requirements-related activities in ISO/IEC 12207 and ISO/IEC 15288. It includes perspectives from areas beyond software such as the business or project that is the

[*1] REBOK is a registered trademark of JISA (Registered trademark classification 9, registration no. 5458370).

[*2] ISO: International Organization for Standardization; IEC: International Electrotechnical Commission; IEEE: Institute of Electrical and Electronics Engineers.

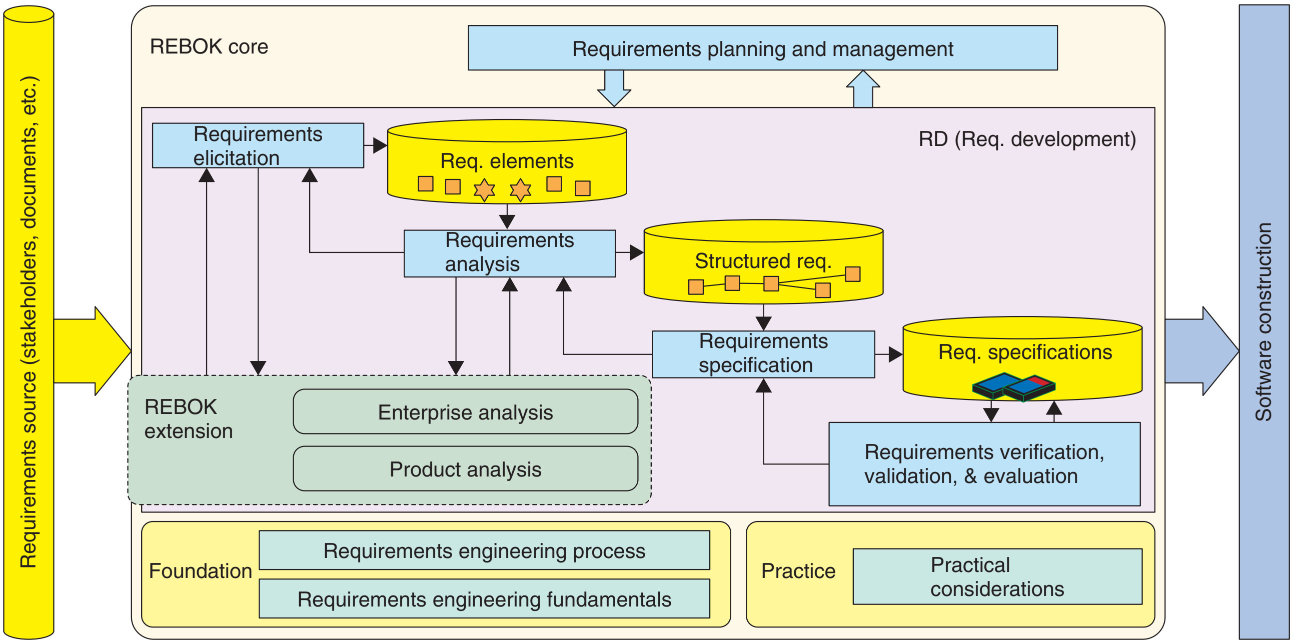

Fig. 1. REBOK process.

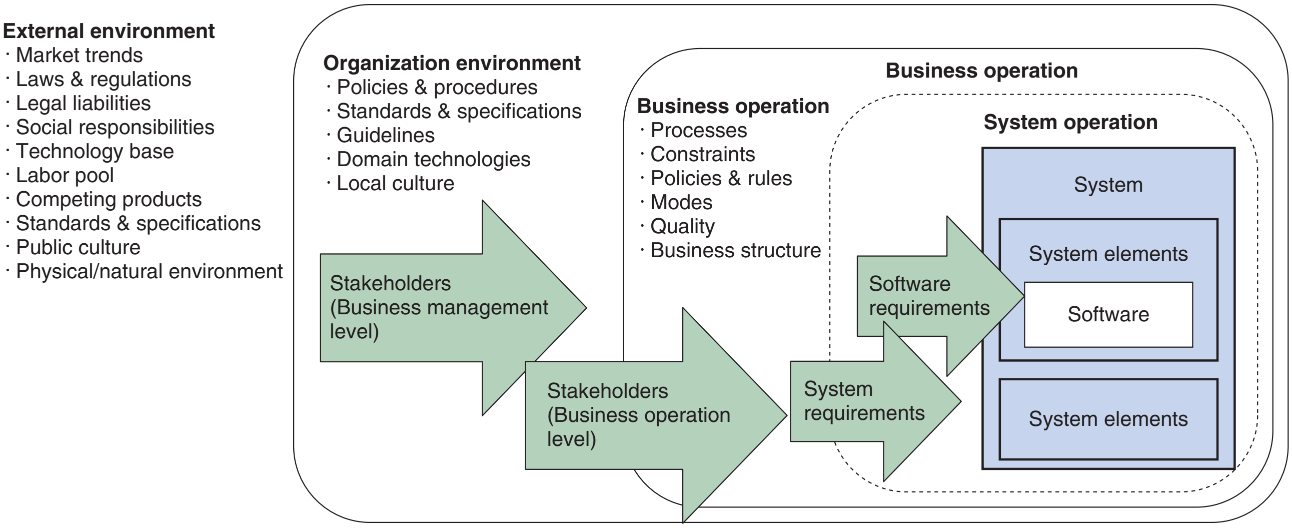

Fig. 2. Areas covered by ISO 29148.

basis for the system and the operation of the system, as shown in **Fig. 2**.

3.3 Software Life Cycle Process Japan Common Frame (SLCP-JCF)

The Software Life Cycle Process Japan Common Frame (SLCP-JCF) was created in 1994 in Japan with the goal of eliminating misunderstandings among stakeholders involved in software development. It provides comprehensive regulation of tasks and roles needed in the areas of business analysis, business design and software-related planning, requirements definition, development, operation, and maintenance as well as various other related activities. SLCP-JCF

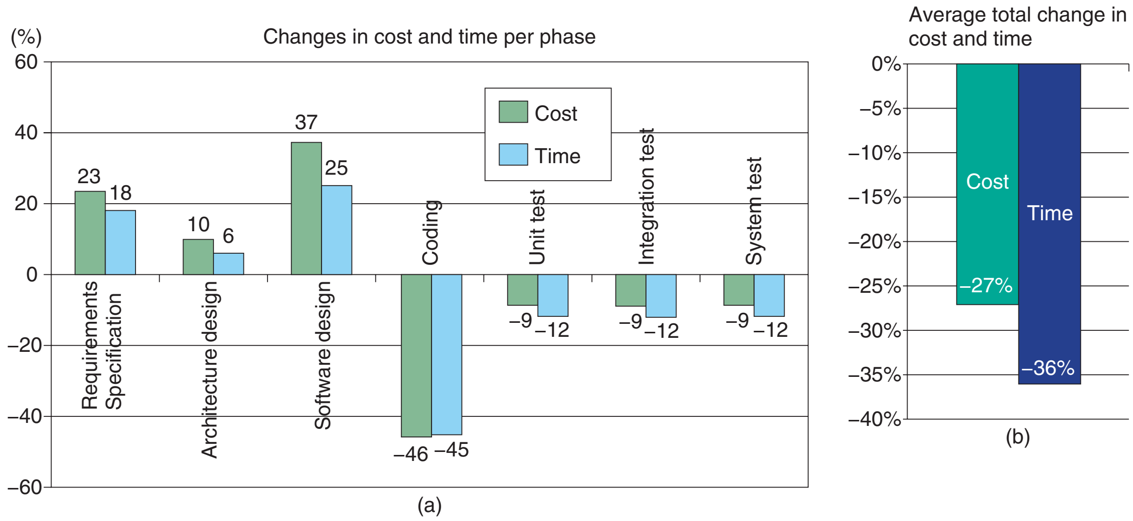

Fig. 3. Cost and time changes of software development in (a) individual development phases and (b) overall.

2007 was published in 2007 and incorporates the idea that there are fundamental issues to be addressed in the process from business and project planning through to requirements definition and enhancing the operation and user processes. SLCP-JCF 2013 was published in 2013, with improvements made to the requirements definition, operations, and service processes.

4. Initiatives related to planning and requirements definition

The NTT Software Innovation Center is studying the following areas related to planning and requirements definition.

4.1 Planning and requirements definition process

As stated previously, there are many stakeholders in the planning and requirements definition process. To achieve smooth communication among them, they must have a common understanding of what sorts of procedures need to be implemented, what steps need to be taken, what inputs and outputs are necessary, and what they each must accomplish with respect to the software development project.

Accordingly, we have been studying planning and requirements definition processes that clarify and standardize the procedures that must be implemented for planning and requirements definition, for defining the styles of deliverables, and for clarifying the division of roles.

Also, as discussed earlier, requirements change during the planning and requirements definition process due to changes in the environment or adjustments made between stakeholders. These changes in requirements also require changes in the schedule or functions to be realized, which have already been agreed upon by stakeholders. Consequently, we are also studying requirements management processes that enable effective management of any changes made to requirements and analysis of the effects of adopting the changes.

We believe that application of these processes we have been studying will help to advance software development projects by achieving mutual understanding among multiple stakeholders.

4.2 Study of methods and techniques used in the planning and requirements definition process

Although procedures and outcomes are decided during this process, the work could proceed more smoothly if certain know-how or skills were applied such as knowledge of writing requirements documents in ways that reduce ambiguity. We are therefore also creating and organizing examples of such know-how and skills.

4.3 Participation in REBOK Working Group (WG)

REBOK has been presented at international conferences related to requirements engineering [6]. The English edition of REBOK is under construction in order to meet the expectations of many people in research and industry. Initiatives to support the use of REBOK are also being undertaken. We are participating in planning for the REBOK WG and are contributing to activities that promote REBOK.

5. Study of the state of development

The improvements achieved at a German company as a result of initiatives they introduced are shown in **Fig. 3**. By performing requirements definition and other upstream processes soundly, they were successful in reducing the cost and time required for later processes, and were able to reduce the overall development cost by 27% and the development time by 36% [7].

We hope to achieve the same kinds of improvements by studying the planning and requirements definition processes and the techniques used in those processes.

In this study, it will be necessary to understand the issues involved in the early process through surveys and other means, and to incorporate input from those involved in actual development.

References

[1] The Information-technology Promotion Agency, Japan, "2012 Survey of the State of the Software Industry," Apr. 2013 (in Japanese).

[2] K. Wiegers, "More About Software Requirements: Thorny Issues and Practical Advice," Microsoft Press, 2006.

[3] JISA REBOK WG (ed.), Requirements Engineering Body of Knowledge, Version 1.0, Kindaikagakusha, 2011 (in Japanese).

[4] ISO/IEC JTC1/SC7, "IEEE Std. 29148: 2011, IEEE Systems and software engineering—Life cycle processes—Requirements engineering," 2011.

[5] The Information-technology Promotion Agency, Japan, "Common Framework 2013," (in Japanese).

[6] M. Aoyama, T. Nakatani, and S. Saito, "REBOK Manifest: Towards a Requirements Engineering Body of Knowledge," Proc. of the 18th IEEE International Requirements Engineering Conference (RE 2010), IEEE Computer Society, pp. 383–384, Sydney, Australia.

[7] M. Broy, "Opportunities and Challenges in Software Engineering for the Next Generation Automotive," Proc of the IEEE 35th Annual Computer Software and Applications Conference (COMPSAC 2011), Munich, Germany.

Yukako Iimura

Researcher, Software Engineering Project, NTT Software Innovation Center.

She received the M.S. degree in engineering from Kumamoto University in 2001 and joined NTT the same year. Her research interests include requirements engineering and knowledge engineering. She is a member of REBOK WG and the Information Processing Society of Japan (IPSJ).

Nobuyuki Kobayashi

Senior Research Engineer, Supervisor, Software Engineering Project, NTT Software Innovation Center.

He received the B.E. and M.E. degrees in applied mathematics and physics from Kyoto University in 1987 and 1989, respectively. He joined NTT in 1989. His research interests include database engineering and software engineering. He is a member of IPSJ, the Institute of Electronics, Information and Communication Engineers, and the Database Society of Japan.

Setsuo Yamada

Research Engineer, Software Engineering Project, NTT Software Innovation Center.

He received the M.S. degree in information science from Tokyo Denki University in 1992 and the Ph.D. degree in engineering from Tottori University in 2006. He joined NTT in 1992. From 1997 to 2000, he was with ATR Spoken Language Translation Research Laboratories. He moved to NTT in 2000. His research interests include natural language processing and software engineering, especially requirements engineering. He is a member of IPSJ and the Association for Natural Language Processing.