

# Requirements Engineering Process Models

Pin Ng

Hong Kong Community College  
Hong Kong Polytechnic University, Hong Kong.

## Abstract

Requirements engineering (RE) consists of several interrelated activities which span throughout the whole process of software development. The activities include requirements elicitation, requirements specification, requirements validation, and requirements management. As a preliminary work of our research, this paper provides a comprehensive review of some of the frequently cited process models for requirements engineering.

**Keywords:** *Software Requirements, Requirements Engineering.*

## 1. Introduction

The success of a software system is measured by the degree to which it can meet the purpose for which it was intended [6]. Findings from an industrial survey by the Standish Group indicate that only 28% of the software development projects are successful [13]. A primary cause for this situation is due to the problems related to software requirements.

Requirements engineering has been advocated as an important phase in the software development life cycle [7][11] for tackling the problems related to software requirements. It is concerned with defining the purpose of a software system and the contexts in which it will be used [6]. The deliverables from requirements engineering guide and influence all the steps in the development process.

In Zave [15], Requirements Engineering (RE) is defined as:

*“RE is the branch of software engineering concerned with the real-world goals for functions of and constraints on software systems. It is also concerned with the relationship of these factors to precise specifications of software behavior, and to their evolution over time and across software families”.*

The definition implies that, in RE, there should be a mechanism for validating the conformance of the target software system with those “real-world goals” specified in the software requirements specification.

## 2. Software Requirements

A software requirement is a statement of a system service or constraint about the target computer-based system [4]. More formally, according to IEEE [1], the definitions of software requirements are:

- (1) A condition or capability needed by a user to solve a problem or achieve an objective.
- (2) A condition or capability that must be met or possessed by a system or system component to satisfy a contract, standard, specification, or other formally imposed documents.
- (3) A documented representation of a condition or capability as in (1) or (2).

In practice, there exist wide variety of computer-based systems and each of them requires some specific type of software requirements to be satisfied.

Sawyer and Kotonya [8] classified computer-based systems into two broad categories: user-configured systems and custom systems. User-configured systems refer to integrating together a system from existing software products by the end-users. Most of the personal computer systems are of this category. The software requirements for user-configured systems are created by the software developers who develop various software products or packages according to their perception about what is marketable. In the situation of developing custom system, a client produces a set of requirements for a software system and the software developer implements and delivers that system. The software requirements for a custom system describe the services or functionalities to have to be provided by the target system as a whole so as to satisfy the client’s needs.

Custom systems can further be classified into three major types [8]:

*Information systems:* An information system is concerned with processing information using some standard computer hardware systems, such as mainframe computers and workstations, which are running with standardized commercial operating systems. Typical examples are, accounting information system, order booking system, and student enrollment system.

*Embedded systems:* In embedded systems, software is used as a controller [14] in some broader hardware system. They range from simple systems, like a MP3 player, to complex control systems, such as cruise controller.

*Command and control systems:* These are combinations of information systems and embedded systems where specially built machines are used to collect and store information for further processing. These systems usually involve the coordination of various types of computer hardware, which are networked in functioning, for example, automated teller machine (ATM) system.

In general, a software requirements specification for a computer-based system is a complete description of the required behaviors of the target system which are expressed in terms of functional requirements and non-functional requirements [4].

Functional requirements define the internal workings of the software, such as data manipulation and processing, and other specific functionality that shows how the user's requirements are to be satisfied. Nonfunctional requirements are about constraints on the design or implementation, such as performance, security, and availability requirements. Some examples of software requirements are listed in Table 1.

Table1: Examples of software requirements

| Requirement              | Purpose  | Sample software requirements for an online banking system  |
|--------------------------|--|--|
| Functional requirement   | Outline of what the system will do for the users.        | The system permits users to access their bank accounts and withdraw money from the accounts.                   |
| Performance requirement  | Speed or duration of using the system.                   | Each transaction will be completed within 3 minutes.   |
| Security requirement     | Measures taken to prevent improper or unauthorized use.  | Users are required to enter password to authorize each transaction.  |
| Availability requirement | Ability of the system to be used in its intended manner. | The system will be available for transactions 24 hours a day, 7 days per week, except for planned maintenance. |

### 3. Requirements Engineering Process Models

RE consists of several interrelated activities which span throughout the whole process of software development for managing software requirements. A number of RE process models have been proposed. In this section, some of the frequently cited process models for requirements engineering are described and compared.

#### 3.1 Requirements Engineering: A Roadmap

Nuseibeh and Easterbrook [6] emphasize that requirements engineering is the process of discovering requirements, by identifying stakeholders and their needs, and documenting them for analysis, communication, and subsequent implementation. The core requirements engineering activities include:

- *Eliciting requirements:* To find out what problems need to be solved and identify the system boundaries.
- *Modeling and analyzing requirements:* Models are constructed for analyzing the captured requirements.
- *Communicating requirements:* Requirements are documented to facilitate effective communication of requirements among different stakeholders.
- *Agreeing requirements:* Requirements are agreed among the stakeholders and formed the basis for requirements validation.
- *Evolving requirements:* Changes to requirements documentation are managed in response to changing stakeholders' needs.

#### 3.2 SWEBOK

SWEBOK [2] gives a detailed specification of the knowledge areas of software engineering. In relation to software requirements, the specific knowledge area includes acquisition, analysis, specification, validation and management of software requirements. The steps involved in the requirements engineering process [8] are:

- *Requirements elicitation:* To develop an understanding of the problem that the software system is to solve.
- *Requirements analysis:* This step comprises the activities of requirements classification, conceptual

modeling, architectural design and requirements allocation, and requirements negotiation.

- *Requirements specification:* This step is about production of a documentation of requirements that can be systematically reviewed, evaluated, and approved.
- *Requirements validation:* This step is concerned with the process of examining the requirements documents to ensure that they are defining the system that the user expects, through the activities of requirements reviews, prototyping, and model validation and acceptance tests.
- *Requirements management:* Given the iterative nature of the requirements process, requirements continue to evolve as design and development proceed. Change has to be managed by ensuring that proposed changes go through a defined review and approval process, and by applying requirements tracing, impact analysis, and software configuration management.

### 3.3 SEI Curriculum Module

In the SEI Curriculum Module SEI-CM-19-1.2 [9], requirements engineering process is defined with the following steps:

- *Requirements identification:* Software requirements are elicited from users or derived from system requirements.
- *Identification of software development constraints:* Typical constraints, such as cost, capabilities of hardware, fault tolerance objectives and portability, are identified.
- *Requirements analysis:* The analysis includes assessment of potential problems, classification of requirements, feasibility and risks analysis.
- *Requirements representation:* The identified software requirements are specified in textual form, graphical models, or prototypes.
- *Requirements communication:* The documented software requirements are presented to the stakeholders for review and approval.

- *Preparation for validation of software requirements:* Acceptance criteria are defined for requirements validation.

### 3.4 Summary of RE Processes

By summarizing the activities of these requirements engineering process models in Table 2, it reveals a common flow of activities: from requirements elicitation and analysis, requirements specification and review, to requirements validation and acceptance tests.

Table 2: Summary of requirements engineering processes

| RE process models                       | Requirements elicitation and analysis  | Requirements specification and review   | Requirements validation   | Requirements management   |
|---|--|---|---|---|
| Requirements engineering: a roadmap [6] | <ul style="list-style-type: none"> <li>▫ Eliciting requirements</li> <li>▫ Modeling and analyzing requirements</li> </ul>  | <ul style="list-style-type: none"> <li>▫ Communicating requirements</li> </ul>  | <ul style="list-style-type: none"> <li>▫ Agreeing requirements</li> </ul>                               | <ul style="list-style-type: none"> <li>▫ Evolving requirements</li> </ul>   |
| SWEBOK [8]                              | <ul style="list-style-type: none"> <li>▫ Requirements elicitation</li> <li>▫ Requirements analysis</li> </ul>  | <ul style="list-style-type: none"> <li>▫ Requirements specification</li> </ul>  | <ul style="list-style-type: none"> <li>▫ Requirements validation</li> </ul>                             | <ul style="list-style-type: none"> <li>▫ Requirements management</li> </ul> |
| SEI Curriculum [9]                      | <ul style="list-style-type: none"> <li>▫ Requirements identification</li> <li>▫ Identification of software development constraints</li> <li>▫ Requirements analysis</li> </ul> | <ul style="list-style-type: none"> <li>▫ Requirements representation</li> <li>▫ Requirements communication</li> </ul> | <ul style="list-style-type: none"> <li>▫ Preparation for validation of software requirements</li> </ul> |   |

- *Requirements elicitation and analysis:* The detailed requirements are collected and consolidated through the investigation based on the existing system documentations and through the interaction with the users [9]. The problems to be solved and the system boundaries are identified [6]. As there may be many sources, the requirements have to be classified, prioritized according to their importance, and check if the resulting requirements reflect the real needs of the users [8].
- *Requirements specification:* This is about the documentation of the set of requirements after consolidation and analysis. There are several ways for specifying the requirements [9] such as textual form graphical models, or prototypes. The documented requirements can facilitate communication among the stakeholders [6], who will systematically reviewed, evaluated and approved the requirements specification [8].

- *Requirements validation:* The agreed requirements specification for the basis of requirements validation [6]. A series of validation activities is performed, such as checking if all functions required by the users are included through prototyping and model validation [8]; and defining the acceptance criteria [9] for the validation process.

Because of the iterative nature of the requirements engineering process and the changing requirements, Nuseibeh and Easterbrook [6], and Sawyer and Kotonya[8] further defined the activities for managing requirements.

- *Requirements management:* Specified requirements are modified in response to changing stakeholders' requirements [6] and the iterative nature of the requirements process [8]. A series of change management, software configuration management and requirements traceability [3] are involved.

#### 4. Requirements Engineering Process Maturity Levels

In real practice, within an organization, there are a number of major stakeholders [11] concerning software development:

- Project manager: who is in charge of scheduling and allocation of resources in software development project.
- System analysts: who are responsible for the specification of software requirements and test plan.
- Software engineers: who are responsible for software development.
- Clients or system end-users: who will use the system after it has been delivered.
- Testing team: who will perform the software testing based on the test plan.
- Maintenance team: who will perform the maintenance of the software system in case some modifications are required.

Requirements engineering process maturity [12] refers to the extent to which an organization has a defined

requirements engineering process based on good requirements engineering practices [3].

Organizations with a mature requirements engineering process will have the process explicitly defined. They will use appropriate methods and techniques for requirements engineering. With reference to the Software Engineering Institute's capability maturity model [10], the requirements process maturity model [12] is defined using a three-level model (see Figure 1).

*Level 1 - Initial:* Organizations at the initial level do not have a defined requirements engineering process and they often fail to produce quality requirement documents on time and within budget. The organizations are dependent on the skills and experience of individual developers for requirements elicitation, analysis and validation.

*Level 2 - Repeatable:* In those organizations at this level, standards for requirements documents and requirements descriptions are defined, together with policies and procedures for requirements management. Some advanced tools and techniques may be involved in the requirements engineering processes. The organizations can produce requirements documents on schedule with consistently high quality.

*Level 3 - Defined:* Organizations at this level have a defined requirements engineering process model based on good practices and techniques, with an active process improvement program in place. The organizations can make objective assessments of the value of new methods and techniques.

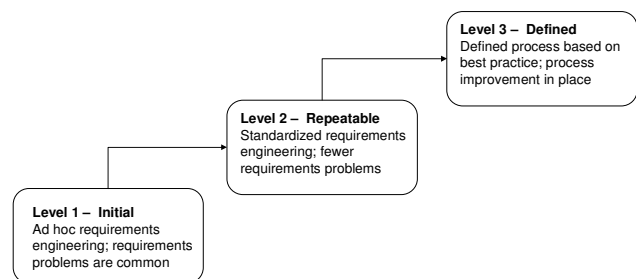


Fig. 1 Requirements engineering process maturity.

The actual practices of requirements engineering depend on the organization's maturity in implementing software requirements engineering processes. In particular, some major human issues are:



*Staff training:* the technical personnel, including the development team, the testing team and the maintenance team, should have sufficient training [3] of applying systematic approaches in requirements specification, test cases design, test cases execution, and test review.

*User involvement in RE process:* the development team should be able to communicate well with the users (clients) in order to capture their requirements correctly. The users will also be involved in conducting the acceptance testing [5] for requirements validation.

#### 4. Conclusions

RE consists of several interrelated activities which span throughout the whole process of software development for managing software requirements. The activities include requirements elicitation, requirements specification, requirements validation, and requirements management. A number of RE process models have been proposed. The paper summarizes the preliminary work of our research, which provides a basic understanding of the RE process and key activities involved in relation to the process maturity level.

We shall further conduct questionnaire survey for collecting the views from the practitioners regarding the industrial practice of the RE process. This could help us to attain a set of “best practice” strategies for implementing the RE process.

#### References

- [1] IEEE, IEEE Standard Glossary of Software Engineering Technology, IEEE Std 610.12-1990, Institute of Electrical and Electronics Engineers, New York, 1990.
- [2] IEEE, Guide to the software engineering body of knowledge (SWEBOK), Professional Practices Committee, IEEE Computer Society, 2004.
- [3] C. Jones, Software engineering best practices: lessons from successful projects in the top companies, New York : McGraw-Hill, 2010.
- [4] G. Kotonya, and I. Sommerville, Requirements engineering : processes and techniques, Chichester ; New York : J. Wiley, 1998.
- [5] K. Naik, and P. Tripathy, Software testing and quality assurance: theory and practice. John Wiley & Sons, 2008.
- [6] B. Nuseibeh, and S. Easterbrook, “Requirements engineering: a roadmap”. Proceedings of the International Conference on Software Engineering (ICSE-2000), pp.35–46, 2000.

- [7] R.S. Pressman, Software Engineering: A Practitioner's Approach, 8th ed., McGraw-Hill, 2015.
- [8] P. Sawyer, and G. Kotonya, “Software Requirements”, Guide to the software engineering body of knowledge (SWEBOK), Professional Practices Committee, IEEE, 2004.
- [9] SEI, Software Requirements SEI Curriculum Module SEI-CM-19-1.2, Software Engineering Institute, Carnegie Mellon University, 1990.
- [10] SEI, The Capability Maturity Model: Guidelines for Improving the Software Process, Software Engineering Institute, Carnegie Mellon University, Addison Wesley, 1995.
- [11] I. Sommerville, Software engineering, 9th ed., Boston : Pearson, 2011.
- [12] I. Sommerville, and P. Sawyer, Requirements Engineering: A Good Practice Guide. John Wiley & Sons, 2004.
- [13] Standish Group, Extreme CHAOS, available at [www.standishgroup.com](http://www.standishgroup.com), 2001.
- [14] C. Walls, Embedded software the works, 2nd, ed., Oxford : Newnes/Elsevier, 2012.
- [15] P. Zave, “Classification of research efforts in requirements engineering”, ACM Computing Surveys, vol.29, no.4, pp.315–321, 1997.