

The Software Requirement Development of Information System

Xianyu Li Hua Wang Xiaokun Li Huijun Zuo and Shan Wang
 Beijing Shi QingHe DaLou Zi Jiu, 100085
 Beijing, China
 919wh_cn@sina.com

Abstract—Based upon the synopsis of software requirement development, this paper describes the essentials of user requirements exploitation, product requirements exploitation, requirements analysis and validation, and then puts forward engineering process relation and requirement development practice. Meanwhile, further works are put forward. This will benefit the software requirement development of information system.

Keywords-information system; software process; requirement development.

I. INTRODUCTION

Along with the broadening of software scale and extending of application field, the prophase tasks of software development are more and more important, and software requirement development is the first and crucial step.

Software requirement development is to analyze and ascertain the requirements of users, software product and components, which contain the needs that relate with life cycle phases and product attributes, and constrains due to the choice of some design solution project, etc.

For the complexity of information system, uncertainty of software requirements and difficulty of software integration, software requirement development should be actively, patiently and continually carried through during the process of the information system development [1, 2, 3].

II. REQUIREMENT DEVELOPMENT RESUME

In the light of the CMMI theory, requirements development process area has three special targets, i.e. user requirement development, product requirement development, and requirement analysis and validation. The practices related to the third target are helpful to the practices related to the anterior targets. Requirements development processes can recursively act with technical solution project processes one another. Software requirement development process is shown in Fig. 1.

A. User Requirements Exploitation

User requirements exploitation is to collect the needs, expectations, constrains and interfaces, and switches to use requirements. It has the following practices.

1) *Eliciting needs.* It is to elicit stakeholder needs, expectations, constraints, and interfaces for all phases of the product lifecycle. It contains four essentials.

a) Investigate and study users' needs, expectations, constraints, and interfaces completely.

b) Channel off requirements by means of various methods, such as prototypes and cases.

c) Let persons of manifold roles to participate in requirement analysis, such as technical master, manager and operator.

d) Catch requirements from the viewpoint of total life cycle.

2) *Developing customer requirements.* It is to transform stakeholder needs, expectations, constraints, and interfaces into customer requirements, and it contains two essentials.

a) Document the users' requirements.

b) System acceptance criteria are included in the users' requirements.

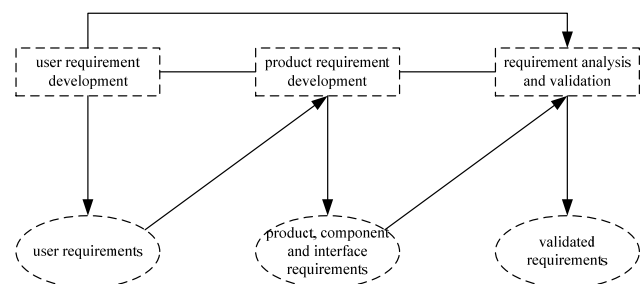


Figure 1. Requirement Development Process

B. Product Requirements Exploitation

Product requirements exploitation is to elaborate on customer requirements so as to develop product and component requirements. It has the following practices.

1) *Establishing product and product component requirements.* It is to establish and maintain product and product component requirements, which are based on the customer requirements, and it contains three essentials.

a) Catch and depict requirements from the viewpoint of total life cycle.

b) Product requirements are rooted in requirements of user and development organization, and derived from technical constraints.

c) Establish horizontal track relation among requirements.

2) *Allocating product component requirements.* It is to allocate the requirements for each product component, and it contains three essentials.

a) Product is divided into components.

b) Functional and non-functional requirements of each component are defined.

c) Establish horizontal track relation among requirements.

3) *Identifying interface requirements.* Some new interfaces between product component and exterior component are set up as software design progresses and product architecture changes. It contains three essentials.

a) Interface requirements contain interior and exterior interface requirements.

b) Interface requirements of hardware are concerned with electron and machine characters.

c) The necessary interfaces, interface functions, transferred messages and special restrictions are described with respect to interface requirements.

C. Requirement Analysis and Validation

Requirements analysis and validation is to analyze and validate requirements so as to develop required functional definitions. It has the following practices.

1) *Establishing operational concepts and scenarios.* It contains two essentials.

a) Operation concepts are to depict how system is made use of, installed, delivered, maintained, technically sustained from user and total life cycle viewpoint.

b) User roles are identified by describing corresponding relations between use roles and scenarios, use cases, scenarios and operation environment.

2) *Establishing a definition of required functionality.* Functionality definition is also called function analysis, and includes action, sequence, input and output, etc. It contains three essentials.

a) Function framework comprises three parts, i.e. function analysis, function classification and requirement allocation.

b) Requirements are classified, and assigned to proper persons and processes.

c) Requirements are defined for each function.

3) *Analyzing requirements.* Requirements are analyzed to ensure that they are necessary and sufficient. It contains three essentials.

a) Requirements are analyzed to ensure derived requirements fulfill original requirements.

b) Requirements are available and testable.

c) Key requirements and requirement metrics are identified.

4) *Analyzing requirements to achieve balance.* Requirements are analyzed to balance stakeholder needs and constraints. It contains two essentials.

a) Tradeoff is put in practice of requirements, schedule, cost and quality.

b) Risks of requirements are identified.

5) *Validating requirements.* Requirements are validated to ensure the resulting product will perform as intended in the user's environment. It contains three essentials.

a) Requirements validation is presided and approved by users.

b) A series of means are adopted in requirements validation of prototype, demonstration, review, primary operation and phase delivery.

c) There is practical evidence of requirements validation.

III. REQUIREMENT DEVELOPMENT ANALYSIS

Software requirement development belongs to engineering processes.

A. Engineering Process Relation

Engineering processes can be integrated with single product development so as to support product oriented process improvement policy. And engineering processes comprise the following special processes, and the relation among engineering processes is shown in Fig. 2.

- Product Integration;
- Requirement Development;
- Requirement Management;
- Technical Solution;
- Validation;
- Verification.

Requirement development process area provide requirements to technical solution area that changes requirements into product architecture, component design and product components. The requirements are also applied into product integration area that integrates product components and carries through verification of component integration interfaces to ensure that the interfaces follow the interface requirements derived from requirement development. Engineering process has two modes of process implementation, i.e. recursion and iteration [4, 5, 6].

B. Requirement Development Practice

Software process model depicts the activities that carried out sequentially in software life cycle. We adopt the life cycle model shown in Fig. 3, which includes three processes of requirement development during the development stages.

1) *Software demonstration.* Software demonstration is the critical step for the establishment of software use requirements. The integrated, reasonable and distinct requirements are the basis of implementation of software engineering practice. The opinions from operation, supportability, test and development units are taken into consideration to carry through the tradeoff of software functions, performances, schedule and cost, and then the demands can be brought forward of functions, performances, operation flow, technical specifications of information exchange, acceptance criterion and supportive condition. Meanwhile, collectivity technical project is compiled based upon the above investigation and analysis.

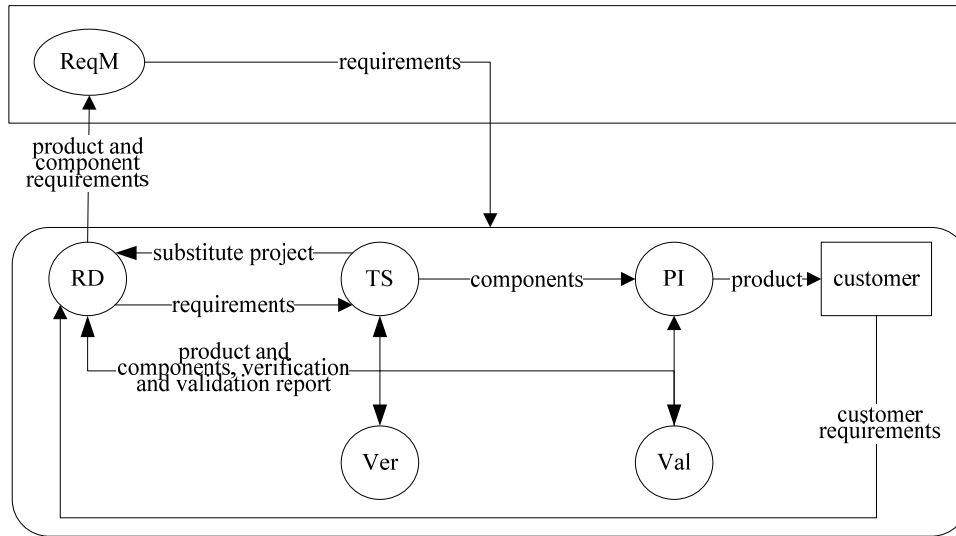


Figure 2. The Relation among Engineering Processes

2) *Requirement analysis.* According to software development contract, we carry out development scheme and requirement analysis. The development scheme includes the contents of development methods, phase partition, review and testing activities, personnel responsibilities, adopted standards and tools and so on; and requirement analysis includes the contents of operation environment, functions, performances, datum, interfaces, safety and confidentiality. Meanwhile, we form software requirements specification. In addition, the requirement analysis of second stage emphasizes on the new incremental parts of requirements.

3) *Milestone review.* Milestone reviews are carried through to confirm that the related tests have been completely and effectively conducted, the software system has satisfied the requirements of related phases, and the technical states of software have been distinctly founded. New requirements are mentioned in the first milestone review, the incremental and revised requirements are absorbed into overall development demands in the second milestone review, and the software system can perfectly be delivered to the users after the last milestone review.

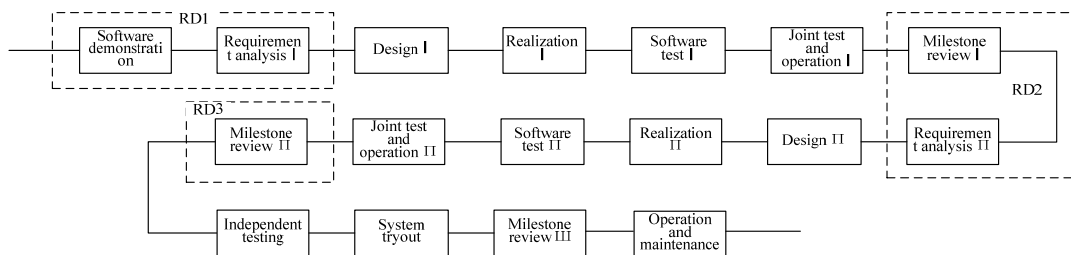


Figure 3. Software Life Cycle Model with Requirement Development Process

IV. FURTHER WORK

While we have achieved much valuable experience in requirement development and software development of information system, there are some actual problems with software requirement development, for instance, the validation of user requirements is some difficult, the ascertainment of interface requirements need constant effort, and the implementation of quality requirements need to be strengthened. Accordingly, we should take pertinent measures [7, 8].

A. Strengthening the Validation of User Requirements

In the light of the multiplicity of user requirement sources, requirement development process is complex and zigzag, and the related requirements should be analyzed to strike a balance between diversification and consistency. Moreover, the validation of user requirement description should be defined and regulated in management and technology levels.

B. Improving the Cooperation of Interface Requirements

For the reason of some uncertainty of technical configuration of exterior system, interface requirements are

difficult to be tracked and ascertained in time. It requires that discussion and cooperation should be actively actualized, and beforehand investigation, technical interface and peer review would be helpful to the timely definitude of exterior and interior interface requirements.

C. *Advancing the Implementation of Quality Requirements*

Quality attribute requirements are just mentioned alike in the requirement documents of various systems, and they should be concretely and distinctly described. It's still complex and meaningful to implement related design of quality attributes. Quality attributes include functionality, reliability, usability, maintainability, etc.

V. CONCLUSION

Software requirement development is much important to the software development of information system, and it's wise to process requirement development along with the evolvement of life cycle model, and we should strengthen the validation of user requirement description, improve the cooperation of interface requirements and advance the implementation of quality attribute requirements, so that requirement development process could be continually improved and helpful to the betterment of development capability of software and system.

ACKNOWLEDGMENT

We are grateful to the specialists and secretaries for their guidance, understanding and patience, and they are a constant source of support and encouragement.

REFERENCES

- [1] L. N. Lu, *Software Requirement*. Beijing: Machinery Industry Press, 2000.
- [2] Z. Y. Zhou, *Modern Software Engineering*. Beijing: Science Press, 2002.
- [3] Y. P. Yang, *The Syncretization of Modern Software Engineering Technology and CMM*. Beijing: People Post Press, 2002.
- [4] R. J. Zheng, *Practical Software Engineering*. Beijing: Qinghua University Press, 2002.
- [5] L. Ruan, M. Y. Lu, et al., *Quality and Reliability Management of Equipment Software*. Beijing: National Defence Industry Press, 2006.
- [6] D. Galin, *Software Quality Assurance*. Pearson Education Limited, 2004.
- [7] C. Tasso, *Software Quality Engineering*. Computational Mechanics Publications, 1997.
- [8] L. P. Shari, *Software Engineering: Theory and Practice*. Prentice-Hall, Inc., 1998.