Volume 3, Issue 2, February 2013 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

An Optimization Approach to Analysis of Requirement Pre-Processing in Software Engineering

Bechoo Lal,
Research Scholar
Dept. of Computer Science & Engineering
JJT University, Rajasthan
India

Dr. Chandrahauns R. Chavan, I/c Professor-cum-**Director** , Alkesh Dinesh Modi Institute for Finacial and Management Studies, University of Mumbai and **Associate Professor and former Director** , Jamnalal Bajaj Institute of Mnagement Studies, University of Mumbai

Abstract— In this paper, The researchers reviewed the existing requirements engineering processes to identify the actual requirements' of the customer for the for the development purpose. The requirements of the customer might be conflicts, incomplete, unidentified, undecided contradictory or ambiguous at the time of initial stage of development. The objective of this is to refine the actual requirements of the customer or organization's need through the pre- processing process which can be implemented in development environment .This research paper is based on the existing research article and analysis of requirements engineering processes.

Keywords: RE, RPP, UR, IR, AR, CR, SRS

I. INTRODUCTION

The primary goal of success of a software development process is the degree to which its meets its purpose for which it was intended. Information processing in the requirement engineering is the process of discovering that purpose, by identifying stakeholder's purpose and their needs. Their goals may vary and conflict, depending on their perspective of the environment in which they work and the task they wish to accomplish.

In this paper section I present an overview of the current research in requirement engineering, presenting in terms of main activities that constitute the field. While these activities are describe independently and in a particular order, in practice, they are actually interleaved, iterative and may span the software development process.

Section 2 background outlines the discipline that provides the foundation for Effective requirement engineering. While section 3 briefly describe the context and process model specification of process model needed in order to begin the requirement process. Section 4 describe the core activities of requirement specification process, section 5 describe the requirements pre- processing process. The section 6 gives the analysis and summary of the pre processing process. The core activities of requirements engineering are given below:

- Eliciting requirements,
- Modeling and analyzing requirements,
- Communicating requirements,
- Agreeing requirements,
- Evolving requirements.

II. BACKGROUND

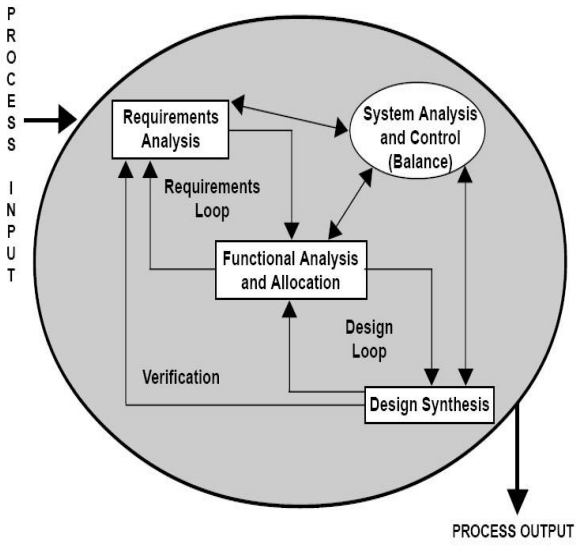
Before discussing requirement engineering activities in more detail, it is worth examining the role of requirement engineering in software and systems engineering and the many disciplines upon which it draws. Zave[5] provides one of the clearest definitions of requirement engineering: "Requirements engineering 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".

This definition is attractive for a number of reasons. First, it highlights the importance of "real-world goals" that motivate the development of a software system. These represent the 'why' as well as the 'what' of a system. Second, it refers to "precise specifications". These provide the basis for analyzing requirements, validating that they are indeed what stakeholders want, defining what designers have to build, and verifying that they have done so correctly upon delivery. Finally, the definition refers to specifications' "evolution over time and across software families", emphasizing the reality of a changing world and the need to reuse partial specifications, as engineers often do in other branches of engineering.

Requirement engineering according to Laplante (2007) is "a sub discipline of systems engineering and software engineering that is concerned with determining the goals, functions, and constraints of hardware and software systems"[6]. In

some life cycle models, the requirement engineering process begins with a feasibility study activity, which leads to a feasibility report. If the feasibility study suggests that the product should be developed, then requirement analysis can begin. If requirement analysis precedes feasibility studies, which may foster outside the box thinking, then feasibility should be determined before requirements are finalized.

III. REQUIREMENT PROCESS MODEL AND SPECIFICATION



FigIII.1: Requirement Process Model and Specification

A software requirements specification (SRS) is a complete description of the behavior of the system to be developed. It includes a set of use cases that describe all of the interactions that the users will have with the software. Use cases are also known as functional requirements. In addition to use cases, the SRS also contains nonfunctional (or supplementary) requirements. Non-functional requirements are requirements which impose constraints on the design or implementation (such as performance requirements, quality standards, or design constraints).

Recommended approaches for the specification of software requirements are described by IEEE 830-1998. This standard describes possible structures, desirable contents, and qualities of a software requirements specification.

TYPES OF REQUIREMENTS

Requirements are categorized in several ways. The following are common categorizations of requirements that relate to technical management [1]:

A. Customer Requirements

Statements of fact and assumptions that define the expectations of the system in terms of mission objectives, environment, constraints, and measures of effectiveness and suitability (MOE/MOS).The customers are those that perform the eight primary functions of systems engineering, with special emphasis on the operator as the key customer. Operational requirements will define the basic need and, at a minimum, answer the questions posed in the following listing [1]:

1. Operational Distribution or Deployment: Where will the system be used?
2. Mission Profile or Scenario: How will the system accomplish its mission objective?
3. Performance and Related Parameters: What are the critical system parameters to accomplish the mission?
4. Utilization Environments: How are the various system components to be used?
5. Effectiveness Requirements: How effective or efficient must the system be in performing its mission?
6. Operational Life Cycle: How long will the system be in use by the user?
7. Environment: What environments will the system are expected to operate in an effective manner?

B. Functional

It explains what has to be done by identifying the necessary task, action or activity that must be accomplished. Functional requirements analysis will be used as the top-level functions for functional analysis [1].

C. Non Functional requirements
Non-functional requirements are requirements that specify criteria that can be used to judge the operation of a system, rather than specific behaviors.

D. Performance Requirements
The extent to which a mission or function must be executed; generally measured in terms of quantity, quality, coverage, timeliness or readiness. During requirements analysis, performance (how well does it have to be done) requirements will be interactively developed across all identified functions based on system life cycle factors; and characterized in terms of the degree of certainty in their estimate, the degree of criticality to system success, and their relationship to other requirements[1].

E. Design Requirements
The "build to," "code to," and "buy to" requirements for products and "how to execute" requirements for processes expressed in technical data packages and technical manuals [1].

F. Derived Requirements
Derived requirements are implied or transformed from higher-level requirement. For example, a requirement for long range or high speed may result in a design requirement for low weight [1].

G. Allocated Requirements
This requirement is established by dividing or otherwise allocating a high-level requirement into multiple lower-level requirements. Example: A 100-pound item that consists of two subsystems might result in weight requirements of 70 pounds and 30 pounds for the two lower-level items [1].

IV. CORE ACTIVIES OF REQUIREMENTS SPECIFICATION

A. Eliciting Requirements
The elicitation of requirements is perhaps the activity most often regarded as the first step in the RE process. The term" elicitation" is preferred to "capture", to avoid the suggestion that requirements are out there to be collected simply by asking the right questions [7]. Information gathered during requirements elicitation often has to be interpreted, analyzed, modeled and validated before the requirements engineer can feel confident that a complete enough set of requirements of a system have been collected. Therefore, requirements elicitation is closely related to other RE activities – to a great extent, the elicitation technique used is driven by the choice of modeling scheme, and vice versa: many modeling schemes imply the use of particular kinds of elicitation techniques.

B. Modeling and Analyzing Requirements
Modeling is the construction of abstract descriptions that are amenable to interpretation – is a fundamental activity in RE. So much so that a number of RE textbooks (e.g., [8]), focus almost entirely on modeling methods and their associated analysis techniques. Models can be used to represent a whole range of products of the RE process. Moreover, many modeling approaches are used as elicitation tools, where the modeling notation and partial models produced are used as drivers to prompt further information gathering.
The key question to ask for any modeling approach is "what is it good for?", and the answer should always be in terms of the kind of analysis and reasoning it offers. We suggest below some general categories of requirement engineering modeling approaches, and give some example techniques under each category. We then suggest some analysis techniques that can be used to generate useful information from the models [17].

C. Communication Requirements
Requirement engineering is not only a process of discovering and specifying requirements, it is also a process of facilitating effective communication of these requirements among different stakeholders. The way in which requirements are documented plays an important role in ensuring that they can be read, analyzed, (re-)written, and validated. The focus of requirements documentation research is often on specification languages and notations, with a variety of formal, semi-formal and informal languages suggested for this purpose [8]. From logic [3] to natural language [2], different languages have been shown to have different expressive and reasoning capabilities.

D. Agreeing Requirements
As requirements are elicited and modeled, maintaining agreement with all stakeholders can be a problem, especially where stakeholders have divergent goals. Recall that validation is the process of establishing that the requirements and models

elicited provide an accurate account of stakeholder requirements. Explicitly describing the requirements is a necessary precondition not only for validating requirements, but also for resolving conflicts between stakeholders. Techniques such as inspection and formal analysis tend to concentrate on the coherence of the requirements descriptions: are they consistent, and are they structurally complete? The formal method SCR [9] illustrates this approach. The SCR tool provides automated checking that the formal model is syntactically consistent and complete. In contrast, techniques such as prototyping, specification animation, and the use of scenarios are geared towards testing a correspondence with the real world problem. For example, have all the aspects of the problem that the stakeholders regard as important been covered?

Requirements validation is difficult for two reasons. The first reason is philosophical in nature, and concerns the question of truth and what is knowable. The second reason is social, and concerns the difficulty of reaching agreement among different stakeholders with conflicting goals.

E. Evolving Requirements

Successful software systems always evolve as the environment in which these systems operate changes and stakeholder requirements change. Therefore managing change is a fundamental activity in requirements engineering [10]. Changes to requirements documentation need to be managed. Minimally, this involves providing techniques and tools for configuration management and version control [11], and exploiting traceability links to monitor and control the impact of changes in different parts of the documentation. Typical changes to requirements specifications include adding or deleting requirements, and fixing errors. Requirements are added in response to changing stakeholder needs, or because they were missed in the initial analysis. Requirements are deleted usually only during development, to forestall cost and schedule overruns, a practice known as requirements scrubbing [12].

F. Requirements Pre – Processing Process

This paper has set out an information pre processing system to identify and detect the major problem during requirement analysis. The problem can stated as [2]:

1. Unclear Requirement (UR)
2. Incomplete Requirement (IR)
3. Ambiguous or Contradictory Requirement (AR/CR)
4. Undecided
5. Unable to identify
6. Unable to express

I feel that no process is fully completed if it does not produce the correct result, and damage the entire development system.

V. FRAMEFOWK OF REQUIREMENT PRE- PROCESSING PROCESS

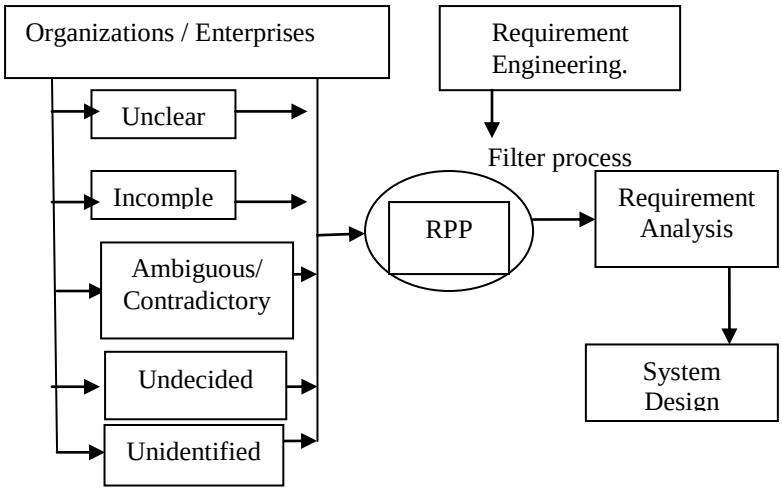


Fig V: Requirement Pre Processing Process

The above frame work show the internal mechanism of the information pre processing system where the actual data is coming and where it is going. It works like as filter process to detect unclear, incomplete and ambiguous or contradictory

data where it is going to implement for the system designing. Before moving to the filtering process I will focus on the following activities:

A. Unclear Requirements (UR)

In several cases organization and stakeholders are unable to identify the problem and their needs and requirement. They do not know what they are exact demanding from the development process. In this the development process would be suffer and next process will blocked. To detect this process we should follow the following point.

1- Try to find out needs and requirement,
2- What they are expecting from the development team,
3- Identify the focus point ,
4- Short out unnecessary data
5- Optimize the requirement
6- Validate the data

B. Incomplete Requirements (IR)

Incomplete requirement can be stated as the organization and stakeholders' people do not know the exact what they are going to solve. It if very difficult process to find the things which is unknown related to development system. In that case the Information pre processing system's responsibility to know the exact needs and requirement of the development. The following procedure can be implemented:

1- Reads the organization and stake holders problem,
2- Find out the problems which are going to automate,
3- List out ,
4- Try to identify, is it the exact what they want to do?
5- Optimize as per needs and requirement,
6- Validate the data

C. Ambiguous and Contradictory Requirements (AR/CR)

No system can be development on behalf of the ambiguous idea or contradictory method. The developers must be avoided ambiguity during development process. To detect this problem the developers must to analyze the functional and non functional requirements whether it is related to system and development process. To avoid ambiguity in the development process the following activity can be stated:

1- The objective of the task must be clear,
2- Describe the goal point,
3- Set the focus point
4- Input and result of the task must be cleared.
5- Describe the framework of the process model.

To avoid the confusion about the task and development process, the developers must be know the nature of the stake holders and organizations. I am also focusing on some little bit idea about to identify the needs and requirements of the stake holders and organizations.

VI. ANALYSIS PROCESS

Analysis process is the important components which analyzed the organization's needs and requirements (legal entities such as companies, standards bodies), which have a valid interest in the system. They may be affected by it either directly or indirectly. A major new emphasis in the 1990s was a focus on the identification of stakeholders. It is increasingly recognized that stakeholders are not limited to the organization employing the analyst. Other stakeholders will include:

1. Anyone who operates the system (normal and maintenance operators).
2. Anyone who benefits from the system (functional, political, financial and social beneficiaries).
3. Anyone involved in purchasing or procuring the system. In a mass-market product organization, product management, marketing and sometimes sales act as surrogate consumers (mass-market customers) to guide development of the product.
4. Organizations which regulate aspects of the system (financial, safety, and other regulators).
5. People or organizations opposed to the system (negative stakeholders; see also Misuse case)
6. Organizations responsible for systems which interface with the system under design.
7. Those organizations who integrate horizontally with the organization for which the analyst is designing the system.

Stakeholder's interviews are a common technique used in requirement analysis. These interviews may reveal requirements not previously envisioned as being within the scope of the project, and requirements may be contradictory. However, each stakeholder will have an idea of their expectation or will have visualized their requirements.

VII. CONCLUSION

In this paper, researcher identified various factors of requirements pre processing processes which is intended to refine the actual needs and requirements of the customers in enterprises or organizations. There are various parameters which are essential in requirements pre processing process, in case if the customer or stake holders are unable to decide , un identified, or uncompleted their requirement. This research paper would be benefited for organizations or enterprises.

ACKNOWLEDGMENT

I am Bechoo Lal an MCA from Banaras Hindu University (BHU), M.Tech(CSE), pursuing PhD(CSE) and currently working as Assistant Professor in Western College, Dept. of IT, University of Mumbai. Dr. Chandrahauns R. Chavan, I/c Professor-cum-Director , Alkesh Dinesh Modi Institute for Finacial and Management Studies, University of Mumbai and Associate Professor and former Director , Jamnalal Bajaj Institute of Mnagement Studies, University of Mumbai , India.

REFERENCES

[1] Systems Engineering Fundamentals Defense Acquisition University Press, 2001
[2] Executive editors: Alain Abran, James W. Moore; editors Pierre Bourque, Robert Dupuis, ed (March 2005). "Chapter 2: Software Requirements ". Guide to the software engineering body of knowledge (2004 ed.). Los Alamitos, CA: IEEE, Computer Society Press. ISBN: 0-7695-2330-7.
[3] http://www.swebok.org/ch2.html. Retrieved 2007-02-08. "It is widely acknowledged within the software industry that software engineering projects are critically vulnerable when these activities are performed poorly."
[4] Wiegers, Karl E. (2003). Software Requirements (2nd ed.). Redmond, WA: Microsoft Press. ISBN 0-7356-1879-8. http://www.processimpact.com.
[5] .Zave, P. (1997). Classification of Research Efforts in Requirements Engineering.ACM Computing Surveys, 29(4): 315-321.
[6] Phillip A. Laplante (2007) What Every Engineer Should Know about Software Engineering. Page 44.
[7] Goguen, J. & Jirotka, M. (Ed.). (1994). Requirements Engineering: Social and Technical Issues. London: Academic Press
[8] Davis, A. (1993). Software Requirements: Objects, Functions andStates. Prentice Hall.
[9] Heitmeyer, C. L., Jeffords, R. D. & Labaw, B. G. (1996). AutomatedConsistency Checking of Requirements Specifications. IEEETransactions on Software Engineering and Methodology, 5(3): 231- 261.
[10] Bohner, S. A. & Arnold, R. S. (Ed.). (1996). Software ChangeImpact Analysis. IEEE Computer Society Press
[11] Estublier, J. (2000). Software Configuration Management: ARoadmap. In this volume.
[12] Boehm, B. (1991). Software Risk Management: Principles andPractices. IEEE Software, 8(1): 32-41.
[13] Ghezzi, C. & Nuseibeh, B. (1998). Guest Editorial - ManagingInconsistency in Software Development. Transactions on Software Engineering, 24(11): 906-907.
[14] Hunter, A. & Nuseibeh, B. (1998). Managing InconsistentSpecifications: Reasoning, Analysis and Action. ACM Transactions on Software Engineering and Methodology, 7(4): 335-367.
[15] Produce Davis, A. (1993). Software Requirements: Objects, Functions and States. Prentice Hall. ced.

