An Insight into Requirements Engineering Processes

Mohd Sadiq and S.K. Jain

Computer Engineering Department
National Institute of Technology,
Kurukshetra-136119, Haryana, India
sadiq.jmi@gmail.com, skj_nith@yahoo.com

Abstract. Requirements Engineering (RE) determines the requirements of software according to the needs of customers. Requirements engineering process includes requirements elicitation, requirements modeling, requirements analysis, requirements verification & validation, and requirements management. Among these sub-processes, requirements elicitation plays an important role in extracting the needs of stakeholders, since it is the first requirements engineering sub-process and has cascading effect. It means errors occurring during requirements elicitation will affect the remaining RE processes. Improper understanding of requirements engineering processes may lead to failure of software projects. This article presents an insight into requirements engineering processes.

Keywords: Requirements engineering, GORE, requirements elicitation, modeling, analysis, requirements verification and validation, and requirements management.

1 Introduction

Requirements engineering (RE) is concerned with the identification of the goals to be achieved by the system. Most of the software fails because of the incomplete, inconsistent, and ambiguous requirements. Clear understanding of the RE process helps to develop a successful software system. The objective of this article is, to have an insight into requirements engineering processes, i.e., requirements elicitation, requirements modeling, requirements analysis, requirements verification & validation, and requirements management. Traditional requirements engineering treat requirements as consisting only of processes and data, thus making it difficult to understand requirements with respect to some high level concerns in the problem domain. Traditionally modeling and analysis techniques do not allow alternative system configuration, where more or less functionality is automated or different assignments of responsibility are explored. Goal Oriented Requirements Engineering (GORE) attempts to solve these problems [12]. Goals have long been used in the Artificial Intelligence (AI). Lamsweerde defines the goal as an "*objective that the system should achieve through cooperation of agents in the software-to-be and in the environment*" [11].

V.V. Das and J. Stephen (Eds.): CNC 2012, LNICST 108, pp. 313–318, 2012.

The rest part of this paper is organised as follows: Section 2 provides the brief description about the requirements engineering. Detailed description about the requirements engineering processes is given in section 3, and finally, we conclude our discussion in section 4.

2 Requirements Engineering

Zave [18] defines the "requirement engineering as the branch of software engineering concerned with the real world goal for functions of and constraints on software systems. It is also concerned with the relationship of these factors to precise specification of software behaviour, and to their evolution over time and across software families". Brooks [3] states that "the hardest single part of building a software system is deciding precisely what to build. Therefore, the most important function that the software builder performs for the client is the iterative extraction and refinement of the product requirements".

3 Requirements Engineering Processes

Nuseibeh and Easterbrook [14] states that requirements engineering processes includes requirements elicitation, modelling, requirements analysis, requirements communicating, requirements agreeing, and requirements management. Jiang et al. [8] definition of requirements engineering processes includes elicitation, analysis & negotiation, documentation, verification & validation, and requirements management. We use the tree like structure (as shown in fig. 1) to delineate the various requirements engineering processes. Fig. 1 represents the comprehensive view of RE processes namely, requirements elicitation, requirements modeling, requirements analysis, requirements verification & validation, and requirements management.

3.1 Requirements Elicitation

It is the first RE sub-process, which includes the learning, extracting, and discovering the needs of stakeholders. The objective of the requirements elicitation technique is to determine what problem needs to be solved, therefore, the elicitation techniques focuses the requirements engineer on the problem domain, rather than on possible solution of those problems. Based on the literature survey [1,2,4,5,6,7,8,9,13,14], we divide the requirements elicitation techniques into 4 types: (i) traditional techniques i.e., based on data gathering, (ii) group elicitation techniques i.e., based on groups or workshop, (iii) cognitive techniques i.e., related to the mental process, and (iv) contextual techniques i.e., based on observation of workplace of the customer.

Hickey and Davis [7] suggests that models and ethnography techniques are the recommended techniques used by most of the analyst during requirements elicitation.

Goal Oriented Requirements Engineering (GORE) determines the high level goals such as business goals and refined it into lower level goals such as technical goals. Various GORE approaches are employed to elicit the requirements, like AGORA

(**A**ttributed **G**oal **O**riented **R**equirements **A**nalysis), i*, KAOS (**K**nowledge **A**cquisition in aut**O**mated **S**pecification, and TROPOS. NFR is used for modeling the **N**on **F**unctional **R**equirements [12].

3.1.1 Traditional Techniques

To capture the initial background of the project, it is important to apply the traditional techniques. There are several types of traditional techniques like interview, survey, questionnaire, and analysis of existing documents. Among the different traditional techniques, interview [6 is the important technique to surface new information or to uncover conflict or politics. Interviews in the requirements engineering discipline generally involve asking the questions with stakeholders about the current system that they use or the system that they want to develop. Techniques such as ethnography and contextual inquiry are the important one because these techniques try to present the portrait of life as seen and understood by the analyst at the work area of the customer [9].

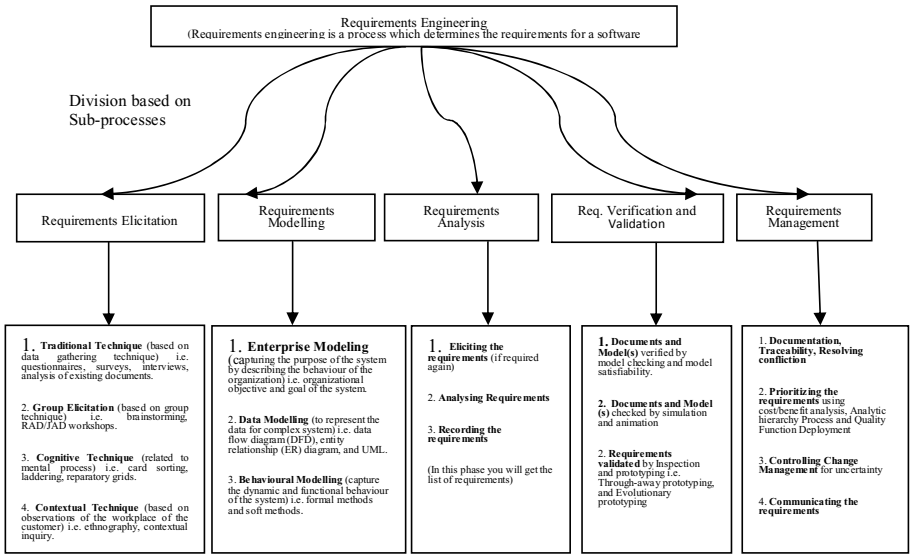

Fig. 1. A Tree-like structure representing the summary of the various Requirements Engineering processes

3.1.2 Group Techniques

Group elicitation technique provides an environment where the people of different domain generate the ideas for the solution of a given problem. Brainstorming is an example of group elicitation technique. Brainstorming attempts to generate new ideas and find a solution for the specific problem. Brainstorming session generally includes the group of 4-10 people. It overcomes the communication barriers among the stakeholders. Joint application development (JAD) is a variation of brainstorming in which stakeholders and users are allowed to participate and discuss the design of the proposed system [7].

3.1.3 Cognitive Techniques

Cognitive techniques [14] are related to the mental processes. It is employed only when complex decision is required. Cognitive technique includes card sorting, laddering, protocol analysis, and reparatory grids.

3.1.4 Contextual Techniques

Contextual techniques, elicits the requirements at the workplace of the customers. In these techniques the requirements for the software system are collected in the context of the end user and the work area of the customer. This technique uncovers hidden details of certain work area that would not be uncovered by different methods like, interview [9].

Ethnography is an observational method and it tries to present the portrait of life as seen and understood by those who live and work within the domain concerned [9]. Contextual inquiry is a field interviewing method that can be used for eliciting requirements for the software system.

3.2 Requirements Modeling

Visual presentation [15] shows semantic information in a form that can be seen, browsed, and edited by humans. Use cases are suitable to elicit the functional requirements, but it neglects the elicitation of the non-functional requirements, such as security requirements. Sindre and Opdahl [16], proposed the misuse case notations and templates to elicit the security requirements. Templates are important because they encourage developers to write clear and simple action sequence. In this section we explain three different areas of modeling i.e. enterprise modelling, data modelling, and behavioural modelling. AGORA, i*, NFR, GBRAM, KAOS, and TROPOS also supports the modeling to represent the FR and NFR of a system.

Enterprise modeling is employed to capture the purpose of a system, by describing the behaviour of the organization in which that system will operate. High level business goals can be used to find out the requirements. *Data modeling* is used to manage and organize the large volume of data. *Behavioural Modeling* describes the overall behaviour of a system. Data processing models and state machine models are the typical examples of the behavioural models. Both of these models are used to describe the behaviour of system. Data processing models show how data is processed as it moves through the system and data flow diagram is used to model the system's data processing. State machine models show the system response to events [15].

3.3 Requirements Analysis

Requirements analysis examines the requirements and selects the best and useful requirements from the set of elicited requirements. After the requirements analysis, if requirements analyst thinks that certain requirements are missing or the requirements are not according to the need of the product, requirements elicitation process will again take place to find out the requirements. Requirements analysis can be define as an activity, which involves the elicitation (if required again), analysing, and recording the requirements. During this process analyst can invite the testers to improve the requirement list [17]. Both requirements engineering and good testing practice are critical to success [10].

3.4 Requirements Verification and Validation

Requirements verification and validation (RV&V) model makes it sure that the product that is developed is according to the needs of users. The actual definition of the verification is that "*are we building the product right*". The ready part of the software and its documents are examined and reviewed by the experts to find out any defects in it. Requirements validation ensures that models and documents accurately express the stakeholder's needs. Research in this area focuses on improving the information provided to the stakeholder for feedback including simulation and animation [4]. There are several methodologies and strategies that are used with requirement verification and validation like inspection, and state based exploration.

3.5 Requirements Management

This is the last sub-process of the requirements engineering. Requirements management is the process of documentation, traceability, prioritization, agreement, communication, and controlling change in the requirements. Requirements engineering is not only the process to find out the requirements, but it is also a process of providing the effective communication of the requirements among different stakeholders [4]. Requirements traceability is employed to read, navigate, query, and change requirements documentation. After eliciting and modeling the requirements, it is difficult to maintain an agreement between/ with all stakeholders, especially when stakeholders have divergent goals. Requirements prioritization plays an important role in the final cost of the software, because addressing the high priority requirements before low priority requirements significantly reduce the cost of the software. There are several techniques to prioritize the requirements like, AHP, QFD, and Cost/ Benefit analysis.

4 Conclusion

This paper presents a comprehensive view of the requirements engineering processes. Requirements elicitation techniques are employed to understand the goal, and objectives for building a software system. Requirements' modeling is used to represent the requirements. Most of the GORE approaches supports the modeling notations for the requirements representations like AGORA, i*, NFR (for modeling the non functional requirements), KAOS, GBRAM, and TROPOS. It is difficult to elicit the high level objective of an enterprise using traditional requirements engineering processes; therefore, to elicit the high level objectives, Goal Oriented Requirements Engineering (GORE) is used. Requirements analysis is the process of determining the user expectations for the software system. Requirements verification and validation (RV&V) model makes it sure that the product that is developed is according to the needs of users. Requirements management is employed for the requirements documentation, communications, prioritizations, and controlling change in the requirements. Fractional knowledge of RE processes may lead to failure of software projects.

References

1. Albayrak, O., Hulya, Bicakcr, M.: Incomplete Software Requirements and Assumptions made by Software Engineers. In: 16th Asia Pacific Software Engineering Conference, pp. 333–339 (2009)
2. Bell, T.E., Thayer, T.A.: Software Requirements: Are They Really a Problem? In: ICSE-2: 2nd International Conference on Software Engineering, San Francisco, pp. 61–68 (1976)
3. Brooks, F.P.: No Silver Bullet: Essence and Accidents of Software Engineering. IEEE Computer 20(4), 10–19 (1987)
4. Cheng, B.H.C., Atlee, J.M.: Current and Future Research Directions in Requirements Engineering. In: Lyytinen, K., Loucopoulos, P., Mylopoulos, J., Robinson, B. (eds.) Design Requirements Workshop. LNBIP, vol. 14, pp. 11–43. Springer, Heidelberg (2009)
5. Davis, C.J., et al.: Communication Challenges in Requirements Elicitation and the use of the Repertory Grid Technique. Journal of Computer Information System, Special Issue, 78–86 (2006)
6. Goguen, J.A., Linde, C.: Techniques for Requirements Elicitation. In: Proceedings, Requirement Engineering, pp. 152–164 (1993)
7. Hickey, A.M., Davis, A.M.: Elicitation Technique Selection: How Do Experts Do It? In: 11th IEEE International Conference on Requirement Engineering (2003)
8. Jiang, L., Eberlein, A., Far, B.H.: Combining Requirements Engineering Techniques-Theory and Case Study. In: Proceedings of the 12th IEEE International Conference and Workshops on the Engineering of Computer –Based systems (ECBS 2005), pp. 105–112 (2005)
9. Keller, T.: Contextual Requirements Elicitation: An Overview. Seminar in Requirement Engineering, Department of Informatics. University of Zurich
10. Kukkanen, J., Vakevainen, K., Kauppinen, M., Uusitalo, E.: Applying a systematic Approach to Link Requirements and Testing: A Case Study. In: 16th IEEE Asia pacific Software Engineering Conference, pp. 482–488 (2009)
11. Lamsweerde, A.V.: Requirements Engineering in the Year 00: A Research perspective. In: 22nd International Conference on Software Engineering (2000)
12. Lapouchnian, A.: Goal Oriented Requirements Engineering: An Overview of the Current Research. Technical Report, Department of Computer Science, University of Toronto (June 28, 2005)
13. Nurmuliani, N., Zowghi, D., Williams, S.P.: Using card Sorting technique to Classify Requirements Change. In: Proceedings of the 12th IEEE International Requirement Engineering Conference (RE 2004) (2004)
14. Nuseibeh, B., Easterbrook, S.: Requirements Engineering: A Roadmap. In: Proceedings of the Conference on the Future of Software Engineering, New York (2000)
15. Rumbaugh, J., Jacobson, I., Booch, G.: The Unified Modelling Language Reference Manual, 2nd edn. Pearson Education
16. Sindre, G., Opdahl, A.L.: Eliciting Security Requirements with Misuse Cases. Requirement Engineering Journal, 34–44 (2005)
17. Uusitalo, E., Komssi, M., Kauppinen, M., Davis, A.M.: Linking Requirements and Testing in Practice. In: 16th IEEE International Requirements Engineering Conference, pp. 265–270 (2008)
18. Zave, P.: Classification of Research Efforts in Requirements Engineering. ACM Computing Surveys, 315–321 (1997)