

Research article

Role and Responsibility Process Model

Saima Tasneem^{a*}, Muhammad Ali Farooq^b, Nadeem Iqbal^c, Najeeb Haider^d, Naveed Ahmad^{e*}

^a Dept. of Computer Science, Indus International Institute, Dera Ghazi Khan, Pakistan

^b Dept. of Computer Science, University of agriculture, faisalabad, Pakistan

^c Department of Management Sciences, Ghazi University, Dera Ghazi Khan, Pakistan

^d Department of Statistics, Ghazi University, Dera Ghazi Khan, Pakistan

^e Department of management Sciences, Indus International Institute, Dera Ghazi Khan, Pakistan

* Corresponding authors. E-mail addresses: saima_tasneem@yahoo.com (Saima Tasneem), naveedahmad@indusdgk.edu.pk (Naveed Ahmad)

Article history:

Received 3 September 2014; Accepted 1 October 2014; Available online 3 October 2014.

Abstract

Agile methodologies, that are most suitable in dealing with volatile business requirements, are likely to face the same challenge as they require developers to drastically change their work habits and acquire new skills. The Role and responsibility Process is one of a number of agile methods for developing software, and a part of the Agile Alliance. It is a disciplined approach to assigning and managing tasks and responsibilities in a development organization. The goal of this process is to produce, within a predictable schedule and budget, high-quality software that will be capable of meeting the needs of its end users. In the local cottage industry of Pakistan, because of the low budget, small team and short time limit, it is almost impossible to follow all standard procedures of model for a successful software application development. The aim of this study is to modify the model according to the needs of local industry of Pakistan.

Keywords: Process model, Role and responsibility, Software engineering, Motion control, Mobile robot

© 2015 Knowledge Journals. All rights reserved.

1. Introduction

There is increasing pressure on software development teams to deliver working systems to business in very shorter time periods. Agile software development practices have increasingly been adopted to respond to the challenges of volatile business requirements, where the markets and technologies evolve rapidly and present the unexpected (Pikkariainen et al, 2008). Representatives from the agile development movement claim that agile ways of developing software are more fitting to what is actually needed in industrial software development. If this is so, successful industrial software development should already exhibit agile characteristics (Hansson et al, 2006).

It is widely believed that Systems Development Methods (SDM) can help improve the software development process. Nevertheless, their deployment often encounters resistance from systems developers. Agile methodologies, the latest batch of SDM that are most suitable in dealing with volatile business requirements, are likely to face the same challenge as they require developers to drastically change their work habits and acquire new skills (Chan and Thong, 2009).

The Role and Responsibility Process is a comprehensive process model that is tailor able, provides templates for the software engineering products, and integrates the use of the Unified Modeling Language (UML). It is rapidly becoming a de facto standard for developing software. The process supports the definition of requirements at multiple levels (Cooper et al, 2006). The Role and Responsibility Process captures many of the best practices in modern software development and presents them in a tailor able form that is suitable for a wide range of projects and organizations. The Role and Responsibility Process delivers these best practices to the project team online in a detailed, practical form (Kruchten, 2003).

2. Material and Methods

A survey of existing literature is used to analyze that agile methods are often seen as providing ways to avoid overheads typically perceived as being imposed by traditional software development environments. However, few organizations are psychologically or technically able to take on an agile approach rapidly and effectively (Qumer and Sellers, 2008).

It is widely believed that Systems Development Methods (SDM) can help improve the software development process. Nevertheless, their deployment often encounters resistance from systems developers.

Agile methodologies, the latest batch of SDM that are most suitable in dealing with volatile business requirements, are likely to face the same challenge as they require developers to drastically change their work habits and acquire new skills (Chan and Thong, 2009).

The Role and Responsibility Process is a comprehensive process covering almost all aspects of software development projects. However, due to the great level of detail provided by Role and responsibility Process, many professionals do not consider RUP practical for small, fast paced projects. Role and responsibility Process proved to be adaptable to the needs of small projects and was very effective in both projects. One key to the successful application of Role and Responsibility Process in small projects is the careful selection of a proper subset of artifacts and keeping these artifacts very concise and free from unnecessary formalism.

2.1 Results and Discussions

In this research focus was to study the exiting requirement management techniques, and try to create a new one for the local industry of Pakistan.

The need of the new requirement management was felt because there are a lot of organizations that are running their own in-house software development, but are facing problem with a lot of requirement management activities.

To build successful complex software systems, developers must collaborate with each other to solve

issues. To facilitate this collaboration, specialized tools, such as chat and screen sharing, are being integrated into development environments. Currently, these tools require a developer to maintain a list of other developers with whom they may wish to communicate and to determine who within this list has expertise for a specific situation. For large, dynamic projects, like several successful open-source projects, these requirements place an unreasonable burden on the developer (Minto and Murphy, 2007).

The reasons for requirement management problem with in-house teams was due working without following any requirement management techniques. The major reason of not using any management technique was extra budget and skilled person required. We already have mentioned this issue. So the need was felt that there must be a new management technique which should be constructed keeping in view the environment and resources of the in-house software development. As the end users of the system are the employees of the company itself and every employee is very well aware of his responsibilities and he can explain his requirements regarding system very well. The problem is just to keep it documented in such a way that system have fully traced back and forward it.

2.2 Request Initiator

End user will initiate the request to accomplish it against one of his responsibilities. User will describe the type of requirement request then he will describe the requirement in detail. Figure below depicts the detail.

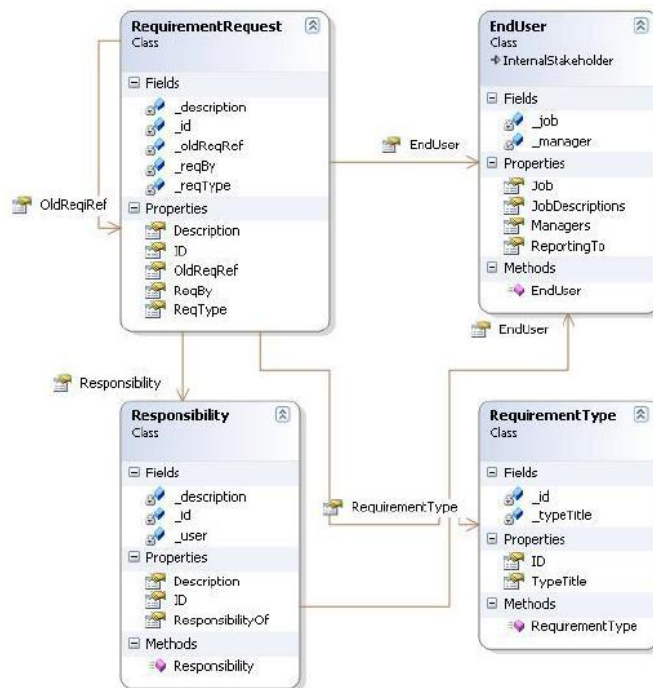


Fig. 1: Request Initiator class diagram

2.3 Request Authorization

When requirement request posted by the end user, his manager or authorized person will check it against the responsibility list of the employee. If requirement

request fulfill the responsibility check list and requirement is verified by the authorized person then he can forward it to development team lead or manager.

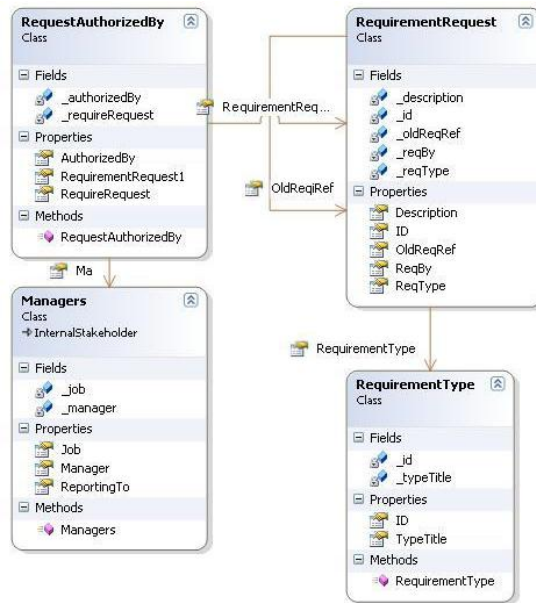


Fig. 2: Request Authorization class diagram

2.4 Requirement development

Once the requirement request will receive by the development team lead he will cross check it with the responsibility check list of the employee role; He will also check the cross reference of the authorized person of the requirement request. Once everything will check

and balance then the development team will start analysis on the requirement request. As before said the customer is actually the employees of the company so there will be no confusion in detail of the requirement request. If some ambiguity find in the requirement then it will be discussed with the end user and the authorized person both.

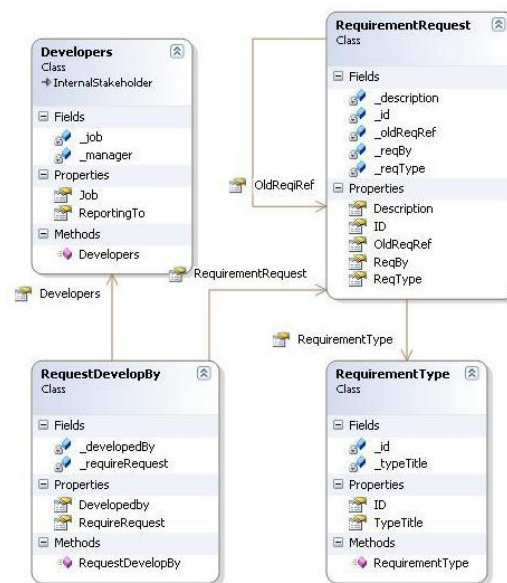


Fig. 3: Requirement development class diagram

2.5 Request testing

Once requirement will developed, it will forward to the end user from where request initiates for the

testing the requirement. Again as the employees are on board there will be no confusion if they find any problem with the required request.

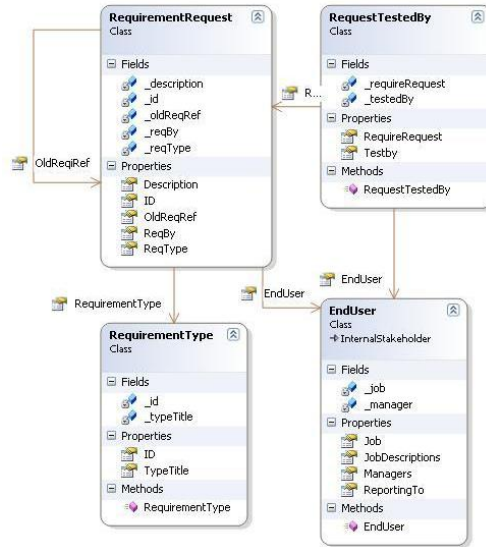


Fig. 4: Request Testing class diagram

2.6 Requirement traceability orbit

Requirement traceability object will use to trace the requirement at any stage of the software development life cycle. Traceability object will contain the actual requirement request and the end user who

initiate the request. Requirement request will contain the role and the responsibility matrix of the end user. Traceability will also contain the authority object and testing object to describe the detail of authorized person and the tester.

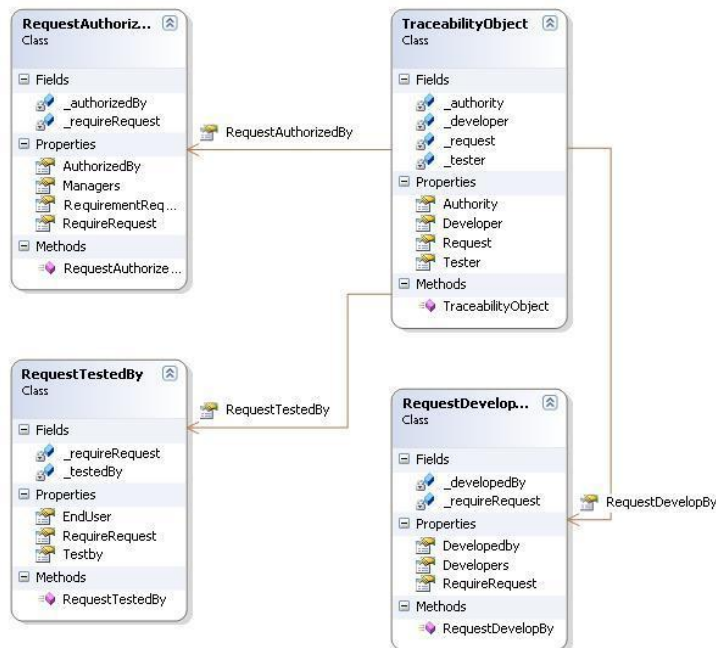


Fig. 5: Requirement Tracability Orbit class diagram

3. Conclusion

A requirement is an agreed standard of need of what a particular product or service should be or perform. Formally it is used in system or software engineering. Each requirement is referred to a necessary attribute, capability, characteristic, or quality of a system. A set of functionalities performed by a software system are called its functional requirements.

Non Functional Requirement in Software engineering presents a systematic and pragmatic approach to building quality into software system. Systems must exhibit software quality attributes such as accuracy, performance, security, safety, availability, maintainability, safety. 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. A framework which is used to plan and control the process of development is known as software development methodology.

References

- Belani, H., Z. Car and A. Caric (2009). RUP-based process model for security requirements engineering in value-added service development, IWSESSICSE Workshop on Software Engineering for Secure Systems, pp:54-60.
- Chan, F. and Y. K. Thong (2009). Acceptance of agile methodologies: A critical review and conceptual framework, *Decision Support Systems*, Elsevier Science Publishers B. V. 46(4): 803-814.
- Cooper, K., S. P. Abraham and R. S. Unnithan. (2006). Integrating visual goal models into the Rational Unified Process, *Academic Press, Inc*, 17(6): 551-583.
- Gotterbarn, D. (2004). UML and agile methods: in support of irresponsible development, *ACM SIGCSE Bulletin*, 36(2): 11 – 13.
- Hansson, C., Y. Dittrich, B. Gustafsson and S. Zarnak. (2006). How agile are industrial software development practices? *Journal of Systems and Software*, 79 (9): 1295-1311.
- Kruchten, P. (2003). *The Rational Unified Process*, Addison-Wesley Longman Publishing Co., Inc, 3, pp:280-282.
- Louvieris P., J. P. Perry and P. B. Davies. (2004). A/B Dashboard: The Case for a Virtual Information Systems Development Environment to Support a RAD Project, *IEEE Computer Society*, Volume 8, pp: 8026-8032.
- Manzoni, L.V. and R.T. Price. (2003). Identifying Extensions Required by RUP (Rational Unified Process) to Comply with CMM (Capability Maturity Model) Levels 2 and 3, *IEEE Transactions on Software Engineering*. 29(2): 181 – 192.
- Misra, S. C., V. Kumar and U. Kumar. (2009). Identifying some important success factors in adopting agile software development practices, *Journal of Systems and Software*, Elsevier Science Inc. 82(11): 1869-1890.
- Pikkarainen, M., J. Haikara, O. Salo, P. Abrahamsson and J. Still. (2008). The impact of agile practices on communication in software development, *The Empirical Software Engineering*, Kluwer. 13(3): 303-337.
- Qumer, A. and B. H. Sellers. (2008). A framework to support the evaluation, adoption and improvement of agile methods in practice, *Journal of Systems and Software*, Elsevier Science Inc. 81(11): 1899-1919.