International Journal of Computer and Communication Technology

Volume 8 | Issue 1 Article 5

January 2017

An Insight into the Importance of Requirements Engineering

Nikhil T. More
Dept. of Computer Technology, Veermata Jijabai Technological Institute, Mumbai, India,
nikhilmore54@gmail.com

Bhushan S. Sapre
Dept. of Computer Technology, Veermata Jijabai Technological Institute, Mumbai, India,
bhushan.sapre316@gmail.com

Pramila M. Chawan
Dept. of Computer Technology, Veermata Jijabai Technological Institute, Mumbai, India,
pmchawan@vjti.org.in

Recommended Citation
More, Nikhil T.; Sapre, Bhushan S.; and Chawan, Pramila M. (2017) "An Insight into the Importance of Requirements Engineering," *International Journal of Computer and Communication Technology*: Vol. 8 : Iss. 1 , Article 5.
DOI: 10.47893/IJCCT.2017.1394
Available at: https://www.interscience.in/ijcct/vol8/iss1/5

An Insight into the Importance of Requirements Engineering

Nikhil T. More, Bhushan S. Sapre & Pramila M. Chawan

Dept. of Computer Technology, Veermata Jijabai Technological Institute, Mumbai, India
E-mail: nikhilmore54@gmail.com, bhushan.sapre316@gmail.com, pmchawan@vjti.org.in

Abstract **-** When developing a software, one of the most important aspects for success of any software project is to get the requirements right. The success of any software project depends on the quality of the requirements. As the projects change over the time, we try to study the requirement process in different type of projects focusing on classical, web-based and open source projects. The requirements dictate all other software engineering processes which in turn influence the Productivity, Quality and Risk. The requirements engineering steers the whole process of the software development to develop the right software. Requirements engineering not only helps the various software process teams but also helps the management to meet the constraints of cost, time and resources. Also requirements can help us to trace the quality of the software product.

Keywords - *Requirements; software; software engineering; management; project management.*

I. INTRODUCTION

Requirements engineering is a process in which we define the prospective system. The requirements act as a guideline for the software development team. The requirements aim at providing an unambiguous, complete and consistent set of requirement that help develop the system. The aim of the requirements engineering is to steer the development toward producing the right software [1]. If you don't get the requirements right, your project will fail, no matter how well the rest of the project was executed. The right requirements produces a number of benefits such as preventing errors, improving quality, and reducing risk throughout software development projects[2], [3]. Requirements engineering process often result in goals, requirements, and assumptions about agent behavior that are too ideal; therefor may need to be revised as the software development progresses.

Requirements, defined generally in natural language, are more often neglected during the software development process. This results in increasing the risk of the software development process. The requirements specified should be understood by everyone. Requirements are the basis for every project, defining what the stakeholders - users, customers, suppliers, developers, business – in a potential new system need from it. The main challenge therefore lies to make the requirements be made unambiguously to address the need or problem completely without using the technical jargons or conventions [4]. These requirements form the basis for various project management activities such as project planning, risk management, acceptance testing, tradeoff and change control.

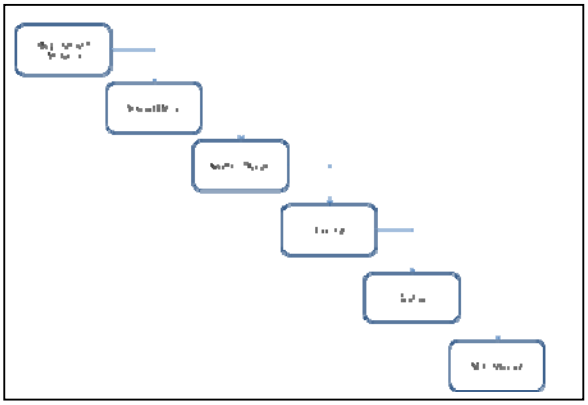

Fig. 1: A Classical Waterflow development model as depicted by R. S. Pressman [8]

The requirements engineering process deals with the real world goals for, functions of, and constraints on software systems [5]. Requirement engineering activities are bound tightly with other system engineering activities and that a complex interaction between requirements engineering practices and other processes may exist in an organization[6].

II. INFLUENCE OF REQUIREMENTS ENGINEERING IN PROJECT MANAGEMENT

As mentioned above, the requirements form the basis of various project management activities. The requirements that are acquired during the systems engineering are used to determine the constraints that rule the planning and management process of the project such as selection, acceptance, and scheduling, scoping and budgetary activities of the project. The requirements

acquired during the requirements engineering process are used to determine the various constraints that are governed by the project management processes such as time, cost, scope and quality of the software product. These requirements also help the management decide which resources to be allotted and schedule the allocation of resources in a timely manner. Also, other project management activities can be scheduled. These are mainly the project feasibility tasks that determine the Management plan for the project.

Requirements engineering forms a part of the Project initiation activities which include the recognition and the beginning of a new software project. The requirements are used to create the plan for the project which defines the breadth and depth of the project.

Requirement acquisition requires the involvement of all the stakeholders which includes, but not limited to users, owners, sponsors, developers and opponents[7]. Such an inclusive requirement process ensures that each stakeholder is satisfied and helps develop precise requirements that are detrimental to the successful completion of a project by helping in the various project management activities.

III. REQUIREMENTS IN SOFTWARE PROJECTS

Reference [8] describes the flow of the classical life cycle as depicted in fig. 1. Fig. 1 shows that the classical life cycle model begins with the requirements analysis. Although the classical software development may feel obsolete, the modern software engineering methods use the modifications of such a model. The requirements seem to steer the flow of the software process as the following processes are highly dependent on the requirements analysis. The final software product is then tested against these requirements before it is delivered to the customer. The requirements are translated into test cases. Thus requirements influence the final outcome of the product indirectly.

A. Classical Software Projects

This model highlights the need of sound requirements engineering practices for the development of successful projects. Requirements engineering is a step by step process by which a vague statement of needs is converted to system component requirements and even subsystem requirements and the corresponding models that are formed in the process form the basis for a detailed model further down the process stream in the requirements engineering process. Also, the requirements can help in modularization of the processes for the development of the software project, so that non overlapping processes can be developed in parallel without crossing over each other.

B. Requirements in Web Projects

Web projects are in a lesser way different from the classical software developed for the standalone system. The web projects need to concentrate on the security aspect along with the classical functional and nonfunctional requirements. Also the web projects need to be compatible as they work on a large number of non-homogenous computers. Thus the web projects requirements analyses are more or less similar to the traditional software projects.

The Web Information Systems are growing more and more complex, and are present in numerous domains. With the increased dependency, they have become critical systems for the business strategy of many organizations [9]. The organizations, therefore, have adapted their software processes to deal with the overall structure of the web applications [10].

C. Requirements in Open Source Projects

Open source projects are a novel way of software process where in a number of development activities take place in parallel over the same piece of the software. This means that each development team develops the same piece of software according to their own interpretations.

In such type of developments several alternatives are developed of which many codes overcome different problems in their own way. Thus, a number of solutions are available in the open source projects. Also the previous software codes are available for reference and for elimination of the bugs in such codes. Such a process has become detrimental for the success of open source software. Reference [11] tries to develop a Web Engineering meta-model that is highly dependent on the Requirements.

IV. REQUIREMENTS AND AGILITY

The business environments in which the current businesses operate are challenging the traditional Requirements Engineering approaches. In the current scenario, the software organizations have to deal with rapidly evolving requirements and manage the changes in requirements so that they are addressed before they become obsolete [12]. Various agile methods, which are aimed to address the above mentioned challenges, are been currently used to tackle such situations and are finding some interest among the practitioners and researchers. Seven agile methods for the Requirements Engineering have been mentioned in [13] as follows

- Face-to-face communication
- Iterative requirements engineering
- Requirement prioritization goes extreme

- Managing requirements change through constant planning
- Prototyping
- Test-driven development

The agile requirements engineering differs from traditional requirements engineering in that it takes an iterative discovery approach [13]. Although agile requirement engineering practices provide benefits such as improved understanding of customer need and the ability to adapt to the evolving needs of today's dynamic environment, they pose distinct challenges.

V. CONCLUSION

As we have seen, the requirements that have been defined at the starting the project influence the rate of success of the project. The requirements engineering is a process that steers the success of a project. Requirements engineering is not only seen as a software development process but it is also a tool for the project manager to decide upon the feasibility of the software project. A project manager may use the requirements acquired in the requirements engineering process to decide the cost, time, scope, and quality constraints tradeoff. Thus requirements engineering can be both a software engineering and project management process.

We have seen the various obstacles that the current organizations are facing as a result of the current environment in which the requirements are evolving rapidly. These could be addressed using agile processes. It has been suggested by [12] that the development organizations carefully compare the costs and benefits of agile requirement engineering practices in their project environment.

REFERENCES

[1] B. Lawrence, K. Weigers and C. Ebert, "The Top Risks of Requirements Engineering," IEEE Software, Nov/Dec 2001, pp. 62-63.

[2] F. Brooks, "No Silver Bullet: Essence and Accidents of Software Engineering," Computer, vol.20, no. 4, pp. 10-19, Apr 1987.

[3] J. Procaccino, J. Verner, S. Wvermyer, and M. Darter, "Case Study: Factors for Early Prediction of Software Development Success," Information and Software Technology, vol. 44, pp. 53-62, 2002.

[4] E. Hull, K. Jackson, and J, Dick, "Reauirements Engineering," 2nd ed., Springer, 2005.

[5] A. van Lamsweerde and E. Letier, "Handling Obstacles in Goal-Oriented Requirements Engineering," IEEE Trans. Software Engg, vol 26, no. 10, pp. 978-1005, Oct. 2000.

[6] D. Damian and J. Chisan, "An Empirical Study of the Complex Relationships between Requirements Engineering Processes and Other Processes that Lead to Payoffs in Productivity , Quality, and Risk Management", IEEE Trans. Software Engg., vol. 32, no. 7, pp. 433-543, July 2006.

[7] K. Schwalbe, " Project Management in IT," Indian Edition, Cengage Learning, 2007.

[8] R .S. Pressman, "Software Engineering: A Practitioner's Approach," McGrawHill, 1997.

[9] D. Lowe, "Web System requirements: an overview," Requirements Engineering Journal, vol 8, no. 2, pp. 102-113, 2003.

[10] A. Ginige, "Web engineering: Managing the complexity of web systems development," 14th Int. conf. on Software Engineering and Knowledge Engineeing, July 2002.

[11] F. Molina, A. Toval, J. Pardillo, C. Cachero, "Towards a Requirements-aware common web engineering metamodel," Latin America Web Conference, IEEE Press, pp. 75-82, 2008, doi: 10.1109/LA-WEB.2008.10.

[11] B. Boehm, "Requirements that handle IKIWSI, COTS, and rapid change," Computer, pp. 99-102, July 2000.

[12] L. Cao and B. Ramesh, "Agile Requirements engineering practices: An Empirical study," IEEE Software, pp 60-67, Jan/Feb 2008.

❑❑❑