



# Analysis of User Requirements Gathering Practices in Agile and Non-Agile Software Development Teams

Sunaina

Department of computer science  
Punjabi University Regional Campus, Mohali  
India

---

**Abstract**— *In software engineering, effective development can only takes place if the requirements from user side are clear and approachable. User requirements play an important role for completion of the successful projects in all of the development methodologies. If requirements are not clear it can be a reason of project failure. Now a day's some industries are not aware about the right way to collect and fulfil the requirements. Effective development can only takes place if the requirements from user side are clear and approachable. Users typically have difficulty explaining what they need, and the problems increases when developers failed to translate requirements into working programs. Ask the right questions and communicate effectively with all stakeholders is the role of project managers or business analysts. This needs training, practice and experience. Clarifying user input and communicating requirements accurately are all critical competencies for today's project managers. Majority of project failure occurred due to wrong information gathering process. So, this study will focus on the analysis of user requirements gathering in agile and non-agile software development teams. Purpose of this study is to analyse the rate of traditional and agile requirements gathering practices in different companies.*

**Keywords**— *Agile Practices, Software Development, User Requirements, Requirements Engineering, Agile Approaches to Requirements Engineering.*

---

## I. INTRODUCTION

Software Engineering is the very integrated part of the industry from mid nineties. Software industry became significant economical activity from the last few decades. According to Fayad et al. [1] 99.2% of software development companies which are small (fewer than 150 employees) develop significant products, for those firms which needs efficient software engineering practices that are suitable for their particular size and type of business. Developing software is a difficult and extremely labor-intensive activity. Developing the software is error prone because of many labor-intensive activities. Every year there are more software-based devices controlling functions that are critical to human survival. The chances of disasters and failures of these software-based devices have greatly increased.

### A. Problem definition

Effectively gathering user requirements is a critical first step of any project and perhaps one of the most challenging project management skills. It is vitally important to build the project on well-formed and verifiable user requirements to avoid cost overruns, unsatisfied users, or even project cancellation. Users typically have difficulty explaining what they need, and the problems increase when developers fail to translate requirements into working programs. Focus of study is on the analysis of user requirements gathering practices in agile and non-agile software development teams. It will be possible by conducting a survey with questionnaires, and by comparing the various responses of the companies.

### B. Background

Various previous studies highlights that it is vitally important to build the project on well-formed and verifiable user requirements to avoid cost overruns, unsatisfied users, or even project cancellation, Users typically have difficulty explaining what they need, and the problems increases when developers fail to translate requirements into working programs. In 1995, Standish Group published a survey report called chaos report. The failure rate of all type of companies is presented below [2].

- The success rate was 9% in large companies, 16.2% in medium scale companies and 28% in small scale companies.
- 61.5 % of large company projects were challenged, compared to 46.7% in medium companies and 50.4% in small companies.
- 29.5 % projects were cancelled in large companies, compares to 37.1% in medium companies and 21.6% in small companies.
- For 100 projects that start, there are 94 projects restarts.

### C. Purpose of the study

The main purpose of the study is to identify the role of user requirements gathering practices in agile and non-agile software development teams. Identify the current scenario of the companies from requirements perspectives and empirically analyzing the rate of agile and traditional requirements gathering practices.

### D. Contributions of this study

The main contributions of this study are highlighted in the following points:

- It helps the various organizations to understand the factors affecting requirements analysis practices.
- It helps people to collect knowledge about the agile features of requirements analysis.

## II. REQUIREMENTS ENGINEERING

RE basically consists of discovering, analyzing and documenting the requirements of the system. Requirement engineering is very main activity of the project because if problem occurs at this time then it becomes a cause of failure of the project. As it is also analyzed that 37% of the problems occurred in the development of challenging systems are related to the requirements phase [24].

### A. Requirements engineering processes

Requirements engineering processes are the activities to handle all types of requirements. The processes used for requirements engineering vary widely according to the people involved, organization developing the requirements and application domain. Main requirements engineering activities are shown in Figure 1.

Conceptually, requirements engineering includes three types of activity:

- **Eliciting requirements:** involves the process of communicating with users and customers to determine their requirements. We can call it requirements gathering also.
- **Analysing requirements:** is also part of requirement engineering to determine the clarity, incompleteness, contradiction and other issues of the stated requirements and these issues are also resolved.
- **Recording requirements:** involves documenting the requirements in various forms such as use cases, process specifications and natural language documents.

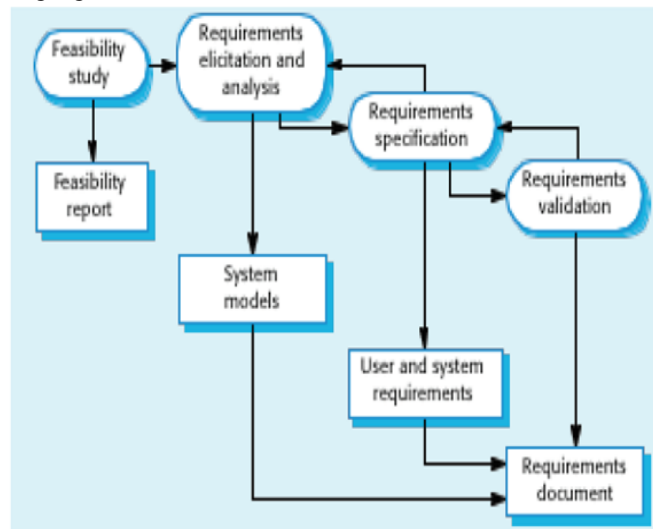


Figure 1 Requirements engineering processes [19].

In 2006, Ray Pharris presented a paper “The importance of requirements definition in IT systems Development” [21] and explore the challenges associated with requirements definition as a major cause of failure in IT systems development. These challenges are [21]:

- IT Project Team Performance
- Size and Complexity of the Project
- Inadequate Understanding of Business Processes
- Selecting the Right Development Methodology
- Organizational Culture

And provide a conclusion that requirements determination is a critical stage in any software development process due to its fundamental importance to making sure the project ultimately solves the right problem in the right way

#### *B. Agile approaches to requirements engineering*

AMs include practices focused on the key factors like incomplete requirements, low customer involvement to reduce the risk of failure. In particular, the aim of incremental development, frequent releases, requirements prioritization before every iteration, and customer involvement is to address the main risk factors.

##### *1) The customer*

In AMs, the customer assumes a principal role. Usually, the term “customer” identifies a set of stakeholders that belongs to the organization that is paying for the development of a software product and belongs to the organization. In this case, due to the different perceptions of the problem that the stakeholders have, the interaction between the development team and the stakeholders is complex [25].

In some AMs, the most common practice is customer on site practice. This means that the customer is a member of the development team, and is always available to discuss issues related to the project with any team member [26].

Some specific requirements for the customer defined by the customer-on-site practice are:

- **Availability:** The questions coming from the development team has to be answered by the customers. Any delay in the answer delays the development of the product.
- **Complete knowledge:** The representative for all the stakeholders is the customer. Since he is the domain expert, he is able to answer all questions and knows how the input/output data required and how the application should work. Again, this is possible if the size of the project is limited.
- **Decision power:** The customer is able to make commitments and final decisions. Acceptance of the features implemented, change in requirements etc. can be decided directly by the customer. There should be a fast decision making process.

##### *2) Waste in requirements*

AMs focus on reduction and the identification of waste in the development process to avoid the creation of waste later in the process. In lean practices, waste always generates further waste so the reduction of waste is too much important [7] [27]. For example, if a factory produces more goods than required by the customers the system produces the following further waste:

- A warehouse
- People and process to manage the warehouse
- People and process to manage the interaction between the warehouse and the factory

##### *3) Requirements evolution*

Elicit all the requirements from the user at the beginning of the development project is very difficult. That’s why they also assume that as the customer may change its mind requirements can evolve. So to manage those upcoming requirements is very necessary. Agile methods approach in two ways to manage variability in requirements:

- **Decoupling requirements:** Requirements have to be as independent as possible in order to clearly identify what to implement and make the order of their implementation irrelevant.
- **Requirement elicitation and prioritization:** There is a requirements collection and prioritization activity at the beginning of each iteration. New requirements are identified and prioritized during this process. In requirements prioritization process, if the requirement is very important it is scheduled for the implementation in the upcoming iteration, otherwise it is kept on hold. Requirement kept on hold indefinitely, if it is not important enough.

##### *4) Non-functional requirements*

During the requirement collection activity, non functional requirements are collected implicitly. After every iteration, the product is released and the customer is able to test the product. If the problems related to non functional qualities are identified, the team can adapt the system to meet such requirements in iterations without affecting too much schedule.

### **III. RESEARCH METHODOLOGY**

The research objective is analysis of user requirements gathering practices in software development teams. The aim is to carry out the research for the companies involved in the software development.

To achieve the set objectives, the questionnaire will be framed by taking care of the agile principles so that we can provide a low level knowledge to the respondents about user requirements gathering practices. So the questionnaire not only takes part in research but also makes publicity of user requirements gathering practices to those who are not aware about agile requirements gathering practices. Respondents were contacted by direct mailing and they were offered personal appointments to fill out

questionnaires. They could choose to have the company data processes anonymously. However, reliability of the questionnaire will also be judged. To quantify the responses, depending upon the need of study likert scale technique will be used and analysis of data will be done by using statistical techniques.

#### IV. CONCLUSIONS

Requirements engineering is the main activity in the software development. If the requirements are not clear, it will be act as one of the main reasons of project failure. Various requirements engineering activities in agile and non agile software development methodologies are discussed. Practitioners even today are not confident of the applicability or suitability of agile in varying real life situations. When organizations are shifting from traditional to agile, it does not able to adopt agile properly and also not traditional. Sometimes blending of traditional and agile creates failures. There is lack of such empirical studies in literature in which analysis of user requirements gathering practices has to be done. And it will also represent the role of customers, developers and managers in requirement engineering process. Moreover the feedback from software developers and requirement analysts through a closed ended survey questionnaire is collected. Although these provide valuable information about the current practices of requirements, descriptive statistics will be used to analyze the rate of agile and traditional requirements gathering practices. It focuses on agility in the software development firms on the basis of requirements practices.

#### ACKNOWLEDGMENTS

I would like to place on record my deep sense of gratitude to people who helped and supported me in my work. I express my sincere gratitude to Ms. Jagpuneet Bajwa , Assistant professor ,Department of computer science, Punjabi University Regional Center for Information Technology and Management, Mohali, for guiding me with attention and care. She has taken pain to go through the work and make necessary corrections when needed. Most importantly, I would like to thank my parents and the Almighty for showing me the right direction, to help me stay calm and keep moving even at times when there was no hope

#### REFERENCES

- [1] M.E. Fayad, M Laitinen, R.P. Ward, "Software engineering in the small," *Communications of the ACM*, vol. 43, no. 3, pp. 115-118, 2000.
- [2] The Standish Group, The Chaos report, [Online]. Available: <http://www.projectsmart.co.uk/docs/chaos-report.pdf> (dated 8/Jan/2010).
- [3] Version one, 3rd Annual Survey: "The State of Agile Development," [Online]. Available: [http://www.versionone.com/pdf/3rdAnnualStateOfAgile\\_FullDataReport.pdf](http://www.versionone.com/pdf/3rdAnnualStateOfAgile_FullDataReport.pdf), 2008.
- [4] R.S. Pressman, "Software engineering: A practitioner's approach ," Boston: McGraw-Hill, 2005.
- [5] M. Huo, J. Verner, L. Zhu & M.A. Babar, "Software quality and agile methods" presented at the *COMPSAC '04: Proceedings of the 28<sup>th</sup> Annual International Computer Software and Applications Conference*, pp. 520-525, 2004.
- [6] W. W. Royce, "Managing the Development of Large Software Systems", in the *Proceedings of 9<sup>th</sup> International Conference Software Engineering, IEEE Computer Society*, pp. 328-338, 1987.
- [7] F. Paetsch, A. Eberlein, F. Maurer, "Requirements engineering and agile software development", *proceedings 12<sup>th</sup> IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, pp. 308-313, June 2003.
- [8] T. Ohno, "Toyota Production System: Beyond Large-Scale Production," Productivity Press, 1988.
- [9] Andrea De Lucia, Abdallah Qusef, "Requirement engineering in agile software development", *journal of emerging technologies in web intelligence*, vol. 2, no.3, pp. 212-220, August 2010.
- [10] S. Powell, F. Keenan, K. McDaid, "Enhancing agile requirements elicitation with personas ", *IADIS International journal on computer science and information systems*, vol. 2, no.1, pp. 82-95, 2007.
- [11] S. Bhalerao, D. Puntambekar & M.Ingle, "Generalizing agile software development life cycle", *International journal on computer science and engineering*, vol. 1, no. 3, pp. 222-226, 2009.
- [12] J. Kalermo & J. Risanen, "Agile software development in theory and practice," M.S. thesis, Department of computer science and information system at University of Jyväskylä, 2002.
- [13] Z. Rachdeva, M. Daneva , A. Herrmann & R.J. Wieringa, "A conceptual model and process for client-driven agile requirements prioritization", in the *proceedings of the 4<sup>th</sup> international conference on research challenges in computer science*, Nice, France, pp. 287-297, ISBN: 978-1-4244-4840-1, 2010.
- [14] M. Maguire & N. Bevan, "User requirement analysis : a review of supporting methods", *proceedings of IFIP 17th world computer congress*, Montreal, Canada, pp. 133-148, 25-30 august 2002.
- [15] A. Chaudhary, A. Punia & M. Pujar, "Requirements engineering role in agile development", presented at Infosys labs. [Online]. Available: <http://www.infosys.com/Infosyslabs/publications/documents/requirement-engineering-agile-development.pdf>, 2008.

- [16] T Aschauer, "Could agile requirements be automated", Innovations for requirement analysis. From stakeholders need to formal designs lecture notes in computer science, Springer, vol. 5320, pp. 25-42, 2008.
- [17] Tobin J. Lehman, Akhilesh Sharma, "Software development as a service: agile experiences," presented at Annual SRII Global conference, pp. 749-758, ISBN: 978-0-7695-4371-0, 2011.
- [18] A. Buchalcevova, "Research of the Use of Agile Methodologies in the Czech Republic", C. Barry, M. Lang, W. Wojtkowski, G. Wojtkowski, S. Wrycza, & Zupancic, *The Inter-Networked World: ISD Theory, Practice, and Education*. Springer-Verlag: New York, ISBN 978-0387304038, 2008.
- [19] Ian Sommerville, "Software Engineering," 7<sup>th</sup> edition, 2004.
- [20] M.A. Awad, "A comparison between agile and traditional software development methodologies", M.S. Thesis, University of Western Australia, 2005.
- [21] Yunyun Zhu, "Requirement engineering in an agile environment", M.S. thesis, Uppsala university, Department of information technology, 2009.
- [22] Ray Phariss, "The importance of Requirement Definition in IT System Development" IS 6840, Systems Analysis, November 29, 2006.
- [23] Hass, Kathleen B. (2007) "The Blending of Traditional and Agile Project Management", PM World Today, Vol. 10 (5), May 2007.
- [24] A. Polini, "Software Requirements," <http://www1.isti.cnr.it/~polini/lucidiSE/Requirements1.pdf>, February 2010.
- [25] K. Beck, "Extreme Programming Explained: Embrace Change," Addison-Wesley, 1999.
- [26] J. Rasmussen, "Introducing XP into Greenfield Projects: Lessons Learned," *IEEE Software*, vol. 20, no. 3, pp. 21-28, May/June 2003.
- [27] J.P. Womack, D.T. Jones, "Lean Thinking: Banish Waste and Create Wealth in Your Corporation," Simon & Schuster, 1998.