International Journal of Engineering Innovation & Research
Volume 3, Issue 3, ISSN: 2277 – 5668

A Survey Study of Key Factors for Improving Requirements Engineering in Agile and Non-agile Software Development

Jaskirat Kaur
Email: jskrt.kr@gmail.com

Jagpuneet Bajwa
Email: jagpuneetbajwa@gmail.com

Abstract **– Requirements engineering is one of the most critical phase of software development life cycle. The primary measure of success in software development projects is largely dependent upon the effectiveness of requirements elicitation and analysis i.e. user requirements have significant impact on software quality, cost and time. Therefore, an effective and efficient system is required to improve requirements engineering process in both agile and non-agile software development. This paper highlights inadequacy in the requirements gathering practices, contrasts the agile and traditional requirement engineering and classifies the key factor that lead to efficient elicitation and analysing of user requirements.**

Keywords **– Agile Approach, Requirements Analysis, Requirements Elicitation, Traditional Approach.**

I. INTRODUCTION

Requirements engineering is the branch of software engineering that prompt the development of a software system by representing the 'why' as well as the 'what' of a system [1]. It is the very main activity of software development because according to Standish Group's CHAOS report 48% of projects failure is due to various user requirement issues: incomplete requirements, low customer involvement, Unrealistic customer expectations and change in requirements [2, 3]. So the volatile, elusive and subject to change nature of user requirements brings challenges to traditional requirement engineering. To cope with fast pace of business and innovations in technology, software projects must control the recurrently changing needs of the customers. Agile software development is presently an emerging software engineering approach that addresses the "need for an alternative to documentation driven, heavyweight software development process" and promotes close collaboration of business people and developers and welcome changing requirements [4].

II. BACKGROUND AND MOTIVATION

F. Paetsch et al (2003) find that traditional approach does not have a good way of stakeholder involvement as agile methods. D. J. Fernandez and J. D. Fernandez (2009) present fundamental information about the agile methodology to encourage its implementation by professionals. Liu Jun et al (2010) reveals how agile approach could be applied to modest-sized system development. Yu Beng Leau et al. (2012) suggest improvements for current agile development. A.Sillitti and G.Succi (2005) present an introduction to the agile methods and their approaches to requirements elicitation and management. Lucia and Qusef (2010) conclude that the secret of the success of agile approach is customer collaboration, good agile developers, and experienced project managers. Cao and Ramesh (2008) analyse the pros and the cons of various agile RE practices. S.C. Misra et al (2009) determine the factors that lead to the success of agile software development. Zornitza Racheva et al (2010) conclude that requirements prioritization is important to maximize the value for the clients and to welcome changing requirements

III. REQUIREMENT ENGINEERING

Requirements engineering primarily deals with identifying the needs of customers, understanding the context in which the developed software will be used, modeling, analysing, negotiating, and documenting the customer's requirements, validating and controlling the change in requirements [5]. It broadly includes following five activities:

Eliciting requirements: is the process of communicating with customers to discover their requirements. Customers are often unsure about their needs or are unable to articulate them clearly, requirements typically change during the course of a project, and analysts are often poorly trained in information gathering techniques. The important techniques for requirement elicitation are Interviews, Ethnography, Brainstorming, and Use Case analysis [6].

Analysing requirements: determine the clarity, completeness, contradiction, inconsistency, feasibility and other issues of the stated requirements. Conflicts in the requirements are resolved through prioritization negotiation with customers. Disputed requirements are prioritized and solutions to requirement problems are identified. The main techniques used for requirement analysis are JAD sessions, prioritization and modeling [6].

Recording requirements: communicate requirements between customers and development team. It acts as a baseline for evaluating subsequent products and processes like design, testing, verification and validation and for change control. Requirements are documented in various forms such as use cases, process specifications and natural language documents.

Validating Requirements: ensures that the requirements are an acceptable description of the system to be implemented. Techniques used for requirements

validation are requirements reviews, unit testing, evolutionary prototyping and acceptance testing [7].

Requirements Management: includes all activities concerned with change and version control, requirements tracing and requirements status tracking [8].

Elicitation and analysis of user requirements gets affected due to various issues i.e. these issues may lead to the situation where user requirements keep changing even when the system development has been started. The major issues among them are:

- Technically skilled and experienced personnel may not be available to perform requirements engineering activities.
- The initial ideas about what is needed are often incomplete and doubtful in the minds of the people leading the acquisition process.
- Project managers often commit to customers without having enough information about project objectives and without discussing details with project team [9].
- Customers often demand new requirements even after the cost and schedule have been fixed and won't commit to a set of written requirements.
- Communication between customers and developers is slow.
- Customers are technically not sound so they are incapable of participating in review meetings, they don't understand the development process and unrealistic expectations of customers often lengthen development schedules [10].
- Engineers and developers often try to make the requirements fit an existing system or model, rather than develop a system specific to the needs of the customers.
- Analysis is often carried out by programmers, rather than engaging a client representative that has people skills and the domain knowledge to understand a customer's needs properly [11].
- Developers start coding instantaneously before they actually understand the whole requirement from analyst, which causes lots of defect fixing or reworking in test/verification phase.
- Usually there is a friction between developers and customers i.e. customers may feel that developers are not cooperative when they refuse to sign up for the development schedule that the customers want, or when they fail to deliver on their promises. Developers may feel that customers unreasonably insisting on unrealistic schedules or requirements changes after requirements have been base lined. There might simply be personality conflicts between the two groups [9].

IV. Comparison between Traditional and Agile Requirement Engineering

By considering the activities of Requirement engineering process the comparison between agile requirements engineering and traditional requirements engineering [3] has been shown in table I:

Table I: Traditional requirement engineering Vs agile Requirement engineering

Stage	Traditional Requirements Engineering	Agile Requirements Engineering
Requirement Capture	Communication gap between customers and developers. The interest of customers is inconsistent and difficult to reconcile.	Customers and developers communicate more frequently and closely.
Requirement Modeling	Requirement priority is only taken when project first started and requirement modeling would be saved as part of requirement document.	Requirement priority is checked at each stage of development cycle and agile modeling is "throw-to-use" that only helps to understand the software.
Requirement Specifications	Massive requirement documentation is developed with complete user requirements when project first started and is not updated frequently. It is suitable for large teams, for explaining the same thing to pretty different people is really unworthy. There is low level of clients' participation.	Vague modest-sized requirement documentation is developed that confirms specific requirements before the next development cycle and is updated very frequently. It is suitable for small teams, as communication could run more smoothly in them. There is high level of clients' participation.
Requirement Verification	No meetings are conducted to verify requirements.	Requirements are verified by conducting inspective meetings and reception testing. Inspective meetings confirm the direction of projects, gain trust between clients and teams.

V. Analysis and Results

The overall goal of the paper is to improve the requirement engineering process using a survey-based statistical study. Specifically, we have attempted to answer the following question:

From the perspective of analysts, developers, managers, what are the factors to be taken care of while dealing with user requirements that will influence the success of projects?

We hypothesize various success factors on the basis of literature review and agile manifesto i.e. most of them are optimized set of agile principles. Time, cost, and quality are three key elements of any project's success. So Our definition of success include five criteria- Reduced delivery schedules, increased return on investment,

increased ability to meet with the current customer requirements, increased flexibility to meet with the changing customer requirements, improved business processes. We have recognized twelve potential candidates for success factors. In our study we relate each of these 12 independent factors with dependent variable "Success". Survey questions were also formulated from these factors. Table II lists all factors with their corresponding labels.

Table II: Variables

Variable	Label
Organizational culture	Ind1
Team capability	Ind2
Personal characteristics	Ind3
Team distribution	Ind4
Communication and negotiation	Ind5
Customer satisfaction	Ind6
Customer commitment	Ind7
Customer collaboration	Ind8
Prioritization	Ind9
Team Size	Ind10
Reviews and Tests	Ind11
Planning	Ind12
Success (Dependent)	Ind13

Research Hypotheses

H1. The stronger the organizational culture, the more likely would be the success of projects.
H2. The better the personal characteristics of the team members in a project, the more likely would be the success of projects.
H3. The greater the satisfaction of the customers in projects, the more likely would be the success of projects.
H4. The greater the commitment of the customers in projects, the more likely would be the success of projects.
H5. The greater the collaboration of the customers in projects, the more likely would be the success of projects.
H6. The more managed prioritization of requirements done, the more likely would be the success of projects.
H7. The more reviews and tests done, the more likely would be the success of projects.
H8. The more specific requirements confirmed before the next development cycle, the more likely would be the success of projects.
H9. The more technically competent the team members are in a project, the more likely would be the success of projects.
H10. The more closely located the project teams are, the more likely would be the success of projects.
H11. The more the communication and negotiation is between project people, the more likely would be the success of projects.
H12. The smaller the size of the teams in a project, the more likely would be the success of projects.

The truths of the above hypotheses are validated using data collected from surveys. The collected data is analyzed by Pearson's coefficient of correlation. But before this, reliability test and descriptive statistics of collected data was performed. The reliability test reveals how consistent the questionnaire is and how consistently the survey respondents responded to the items. It was calculated with the help of statistical tool SPSS. A Cronbach's alpha with value greater than 0.6 is considered standard in survey research. Table III depicts reliability statistics.

Table IIII: Reliability statistics

Coefficient of Reliability	Value of Cronbach's Alpha
Cronbach's Alpha	.742 (Acceptable)

Descriptive Statistics

To recapitulate the information latent in the collected data we calculated minimum, maximum, mean, and standard deviation of the data. Table IV shows the descriptive statistical results.

Table IIIV: Descriptive statistics

Variables	N	Minimum	Maximum	Mean	Std. Deviation
Ind1	50	1.00	2.40	1.5760	.38469
Ind2	50	1.00	3.00	1.6800	.62073
Ind3	50	1.00	3.60	1.9320	.55345
Ind4	50	1.00	3.50	2.2500	.63286
Ind5	50	1.00	4.00	1.9400	.76692
Ind6	50	1.00	3.00	1.4000	.53452
Ind7	50	1.00	2.00	1.2800	.45356
Ind8	50	1.00	3.00	1.3800	.56749
Ind9	50	1.00	2.50	1.6420	.41458
Ind10	50	1.33	5.33	2.4674	.86527
Ind11	50	1.00	3.00	2.0440	.40515
Ind12	50	1.00	3.00	1.3200	.51270
Ind13	50	1.20	2.80	1.9720	.43894

By considering the mean values in table II, amongst all the 12 independent variables, Ind7 (customer commitment) has the lowest mean of 1.2800 with a standard deviation of .45356 that means most of the respondents strongly agree and give highest priority to customer commitment more than anything else. Ind12 (planning), Ind8 (customer collaboration), Ind6 (customer satisfaction), Ind1 (organization culture), Ind2 (Team

capability), Ind9 (prioritization), Ind3 (personal characteristics) and Ind5 (communication and negotiation) are in the range of "agree, strongly agree". Ind10 and Ind 4, has the highest mean value, this shows that most of the respondents are "somewhat agree" with parameters Team distribution and Team size. It is observed that none of the independent variables has a mean score above 3 (in the 7-point Likert scale). This shows that on an average, the respondents were not in disagreement with the factors in their projects.

Correlation Analysis

Correlation analysis helps us to understand the degree of relationship between all our 12 independent variables and the dependent variable "Success" (Ind13). Table V summarize correlation results.

Table V: Correlation Analysis

Variables	Correlation Coefficient	Significance (2-tailed)	Statement
Ind1	.301	.034	Correlation is significant at the 0.05 level
Ind2	.176	.221	Correlation is not significant at the 0.05 level
Ind3	.318	.024	Correlation is significant at the 0.05 level
Ind4	-.253	.076	Correlation is not significant at the 0.05 level
Ind5	.237	.097	Correlation is not significant at the 0.05 level
Ind6	.431	.002	Correlation is significant at the 0.01 level
Ind7	.409	.003	Correlation is significant at the 0.01 level
Ind8	.437	.002	Correlation is significant at the 0.01 level
Ind9	.330	.019	Correlation is significant at the 0.05 level
Ind10	-.004	.976	Correlation is not significant at the 0.05 level
Ind11	.296	.037	Correlation is significant at the 0.05 level
Ind12	.422	.002	Correlation is significant at the 0.01 level

On the basis of correlation analysis, the variables significantly related to Success are: Organizational culture, Personal characteristics, Customer satisfaction, Customer commitment, Customer collaboration, Prioritization, Reviews and Tests, Planning. Variables not significantly related with Success are: Team capability, Team Size, Team distribution, Communication and negotiation. And on the basis of the significance values and the values of correlation coefficients, we either accept or reject our 12 hypotheses. So hypotheses H9, H10, H11, and H12 are rejected and rest are accepted.

VI. CONCLUSION

Requirement elicitation and analysis are critical activities in the software development life cycle and they need extra attention to be paid because if requirements are inadequate, can become root cause of project failure. Organizational culture, Personal characteristics, Customer satisfaction, Customer commitment, Customer collaboration, Prioritization, Reviews and Tests, Planning are critical success factors and should be adopted during requirement engineering to reduce time, reduce cost and to increase quality of software.

REFERENCES

[1] B. Nuseibeh, and S. Easterbrook, *Requirements engineering: a roadmap*, Proceedings of the Conference on the Future of Software Engineering. ACM, 2000.

[2] Lee, Christopher, Luigi Guadagno, and Xiaoping Jia, *An agile approach to capturing requirements and traceability*, Proceedings of the 2nd International Workshop on Traceability in Emerging Forms of Software Engineering (TEFSE 2003). Citeseer. 2003.

[3] L. Jun, W. Qiuzhen and G. Lin, *Application of Agile Requirement Engineering in Modest-sized Information Systems Development*, Second WRI World Congress on Software Engineering, pp. 207-210, 2010.

[4] A. Cockburn, and J. Highsmith, *Agile software development, the people factor*, *Computer* 34.11, pp. 131-133, 2001.

[5] H. Betty, C. Cheng, and M.J. Atlee, *Research Directions in Requirement Engineering*, In Future of Software Engineering, pp. 285-303, IEEE Computer Society, 2007.

[6] I. Sommerville, and P. Sawyer, *Requirements engineering: a good practice guide*, John Wiley & Sons, Inc., 2000.

[7] S.U. Arif, Q. Khan and S.A.K Gahyyur, *Requirements Engineering Processes, Tools/Technologies, & Methodologies*, International Journal of Reviews in Computing, pp. 41-56, IJRIC 2010.

[8] F. Paetsch, A. Eberlein and F. Maurer, *Requirements Engineering and Agile Software Development*, Proceedings of the Twelfth International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, pp. 308, 2003.

[9] H. Naz, M.N. Khokhar, *Critical Requirements Engineering Issues & their Solutions*, In Computer Modeling and Simulation, ICCMS'09, International Conference on, pp. 218-222, IEEE, 2009.

[10] McConnell, Steven, *Rapid Development: Classic Mistakes Enumerated*, 1996, 1st ed., Redmond, WA: Microsoft Press, ISBN 1-55615-900-5.

[11] Wikipedia, Requirements analysis, 2014.

[12] Daniel J. Fernandez and John D. Fernandez, *Agile Project Management- Agilism versus Traditional Approaches*, The Journal of computer Information Systems, ABI/INFORM Global, pp. 10, Winter 2008/2009.

[13] Y. B. Leau, W.K. Loo, W.Y. Tham and S.F Tan, *Software Development Life Cycle Agile vs Traditional Approaches*, International Conference on Information and Network Technology, vol. 37, pp. 162-167, 2012.

[14] A. Sillitti and G. Succi, *Requirements Engineering for Agile Methods*, In Engineering and Managing Software Requirements, pp. 309-326, Springer Berlin Heidelberg, 2005.

[15] A.D. Lucia and A. Qusef, *Requirements Engineering in Agile Software Development*, journal of emerging technologies in web intelligence, vol. 2, issue. 3, August 2010.

[16] S.C. Misra, V. Kumar and U. Kumar, *Identifying some important success factors in adopting agile software development practices*, The Journal of Systems and Software vol. 82, issue 11, pp. 1869-1890, 2009.

[17] Z. Racheva, M. Daneva, A. Herrmann, *A conceptual model and process for client-driven agile requirements prioritization*, In Research Challenges in Information Science (RCIS), 2010 Fourth International Conference on, pp. 287-298, IEEE, 2010.

AUTHOR'S PROFILE

Jaskirat Kaur
has completed Master of Philosophy (M.Phil) in Computer Science from Department of Computer Science, Punjabi University Patiala. She holds Master of Computer Application (M.C.A) degree from Punjab Agricultural University, Ludhiana Punjab, India (2012). She is an Assistant Professor in College of Management and Technology, Patiala, India. Her area of interest is Software Engineering.

Jagpuneet Bajwa
pursuing Ph.D. from Department of Computer Science, Punjabi University Patiala. She is an Assistant Professor in Department of Computer Science, Punjabi University Patiala from 8 years. Her area of interest is Software Engineering.