



Critical Success Factors in Agile Software Development Projects: A Review

Navneet Kaur and Gurpreet Singh

*Department of Computer Engineering,
UCOE, Punjabi University Patiala, (PB), INDIA*

(Corresponding author: Navneet Kaur)

(Received 09 November, 2015 Accepted 02 January, 2016)

(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: Business activities are rapidly changing now a days and there are increasingly complex requirements set on programming solutions. That puts traditional software development methods also called heavyweight behind and leads to the need for different approaches. Modern approach is called agile or lightweight. Dissertation will described the characteristics of some traditional and agile methodologies that are widely used in software development, strengths and weakness between the two opposing methodologies will be compared. We will also discuss the challenges associated with implementing agile processes in the software industry. This anecdotal evidence is rising regarding the effectiveness of agile methodologies in certain environments; but there have not been much collection and analysis of empirical evidence for agile projects in Indian environment.

Key words: Critical Success Factors, Agile Software Development Projects, comprehensive planning, detailed documentation, and expansive design

I. INTRODUCTION

Organizations are facing constantly evolving environments and changing requirements of customers (Nerur *et al.*, 2005). Many efforts which have been done to neutralize the complexities of software development, but software development process have not yet been consistently effective and faces problems yet. These problems cause rejection in final product (software), delays in delivery time and system, discontinue final products, and not pass products. Even software projects which are successfully finished and are already applied in systems may need expensive and continuously maintenance support or other software services and fine release (Chow & Cao, 2008). During the mid nineties, some software engineering practitioners introduced a new group of software development methodologies called Agile Methodologies (AMs). These new methodologies have been developed to overcome the limits of the traditional approaches (Waterfall, Unified Process, Spiral model, etc) in which work begins with the elicitation and documentation of a complete set of requirements, followed by architectural and high level design development and inspection. Agilist argued that freezing the product functionality in early phase of the project and plan everything in advance then following it might not work well in turbulent and complex

environment. Due to these heavy aspects, this methodology was known as Heavyweight or Plan driven. Besides this upfront planning, project failure rate is quite high. Reported statistics by the Standish Group(2009) showed that 24% of information system development projects fail outright, and 32% show a low success rate. This led to the development of methodologies adaptable to new internet applications or mobile devices. The name "agile" came to use around 2001, when seventeen process methodologists held a meeting to discuss future trends in software development. The methods of each of the methodologists had many common characteristics, so they decided to name these different processes 'agile'.

II. AGILE SOFTWARE DEVELOPMENT

Being agile means being able to Deliver quickly, Change quickly, and Change often (Highsmith *et al.*, 2000). In agile methods, people play a driving role in the success of the project, and lot of short time meetings are conducted for knowledge sharing and for the random change in the project if required. Methodologists argue that working software without documentation is better than non-working software with a huge amount of documentation (Koskela and Teknillinen, 2003).

Table 1: Definition of Agility according to different authors.

Author	Definition
Gunasekaran (1999)	(AM) is the ability of surviving and prospering in a competitive environment of continuous and unpredictable change by reacting quickly and effectively to changing markets, driven by customer-defined products and services
Kidd (1994)	A quick and proactive adaptation of enterprise elements to unexpected and unpredicted changes.
Iacocca Institute (1991)	“A manufacturing system with capabilities (hard and soft technologies, human resources, educated management, information) to meet the rapidly changing needs of the marketplace (speed, flexibility , customers, competitors, suppliers, infrastructure, responsiveness)”
Yusuf <i>et al.</i> (1999)	Successful application of competitive bases such as speed, flexibility, innovation, and quality by the means of the integration of re-configurable resources and best practices of knowledge-rich environment to provide customer-driven products and services in a fast changing environment
Kruchten (2001)	‘Ability to adapt and react expeditiously and appropriately to changes in its environment and to demands imposed by this environment. An agile process is one that readily embraces and supports this degree of adaptability. So, it is not simply about the size of the process or speed of delivery; it is mainly about flexibility’
Highsmith (2002)	‘Quickness, lightness, and nimbleness – the ability to act rapidly, the ability to do the minimum necessary to get the job done, and the ability to adapt to changing conditions’

III. METHODOLOGICAL TUG OF WAR: TRADITIONAL VS AGILE

Table 1: Phases of Heavyweight and Agile Methodology.

Phase	Heavyweight	Agile
Assumption	Problem is understood in the beginning of the project and output is defined from the starting of the project.	The preferred output is not fully known until solution is delivered.
Planning	Thorough scheduling of time frame with clearly defined products and documents to be delivered at the end.	Overall high level plan for product development life cycle with planning of only current iteration
Requirement engineering	Detailed and completely defining specifications upfront. Requirement change is a formal work	Welcoming change at any time of the project development. Relaxed change request process
Architecture	Comprehensive and detailed architecture specifications are defined	Minimal outline of architecture and revolution of architecture during the course of project
Coding	Programming is concentrated in one phase and are isolated, Specifications derive the programming process	Programming work is performed throughout the project and onsite customer, collective code ownership, pair programming is some of the features used in this phase.
Testing	Testing is done at the end of the project. Testing is the responsibility of Testing team only.	Testing activities are performed throughout the project, test driven development and pair programming is helpful in reduction of errors early in the project.

IV. CHARACTERISTICS OF AGILE METHODOLOGIES

1. People Oriented
2. Adaptive
3. Conformance to Actual
4. Balancing Flexibility and Planning
5. Empirical Process.
6. Decentralized
7. Simplicity
8. Collaboration
9. Small Self-organizing teams

V. SURVEY OF LITERATURE

Review of literature helps to lay a strong foundation for research projects. It is very essential and important part of research, it helps in figuring out what has been done in relation to the problem being investigated. It ensures that no duplication occurs in the present study. Furthermore, it brings about important understandings and insights necessary for the development of a logical framework (Gay, 1976). Referring to the research that has already been done by different researchers in the related area helps to find any gaps, if exists. Review of literature helps in identifying critical knowledge gaps and motivates researchers to close this breach. This chapter serves the same purpose.

Relevant literature has been exhaustively surveyed and analyzed so as to identify the current state of affairs in the fields of agile software development, knowledge management and distributed agile software development to dig out the available gaps in the area. This thesis study contains 2 sessions – literature review and empirical analysis. As it has been describe in many chapters of this study this thesis work is a quantitative study which evaluates some factors of agile methodology in practical software development process in a company. As a result both literature review and empirical study is essential to get the result. Review of literature is to know about the state of the art and to gain more insight of the high points of agile methodologies (Objective 1and 2). Empirical approach to know about the end user perspective of success factor of agile in Indian scenario (Objective 3

VI. RESEARCH PROBLEM

It is argued that most of these projects do not fail due to technology, but due to social and organizational deficits, and a lack of effective communication especially in Indian scenario, where cultural ,social, religious aspects are the most infusing factors. Furthermore, larger projects are more likely to fail than small projects. Since agile methods strongly focus on people and interactions, it is likely that communication and leadership style are important candidate success factors in agile project success, but there is hardly enough proof whether the global factors apply in Indian scenario also. Social and cultural diversity of India add up to this complexity. These assumptions may especially hold in larger projects, since these factors become even more important when scaling up agile methodologies. Therefore, it is important to gain more understanding about the relative importance of these critical success factors.

VII. CRITICAL SUCCESS FACTOR

Critical Success Factor is introduced as an approach which detects names and evaluates an organization's performance. This approach was first explained by Rockhart (1984) and after that year was developed and became established in better way (Bullen, Rockhart, 1981; Rockhart and Crescenzi, 1984). Critical Success Factor is explained by Bullen as limited number of domains in which real satisfaction will result and ensure accomplishment specially in competitive performance for all individuals , departments and organization. Critical success factors are key areas where every thing is supposed to be done in right method through business process in order to flourish the accomplishment and in order to achieve manager's goals. In software development project area, the Critical Success Factors method has also been considered in recent studies. Critical success factors in

development projects are usually found to be relevant to project management techniques basis or to relevant to the combination of software development and business strategy (Bytheway, 1999). Another research works explains that Critical success factors in software development projects contains variety of dimensions, start from the development life cycle, estimation and validation and end to executive management and project management, or resource management and strategic planning (Bosghossian, 2002).

VIII. SUCCESS FACTORS IN AGILE SOFTWARE DEVELOPMENT PROJECTS

So far, hardly any formal study on Critical Success Factors (CSF) in the agile software development project has not been found from Indian context, based on recent researches in previously reviewed literature or practitioner literature which are relevant to agile development process topic. Although, some case studies and theoretical researches about successes or pitfall problems in agile implementation in agile development projects existed and some of them has been referenced in this study. Over viewing both failures and successes factors in literature review will help author to identify the possible success factors in agile development projects, on the other side failures factors which can help professionals to understand what and how to avoid certain serious problems and critical issues these facts are important for success of a project.

IX. CONCLUSION

The critical requirement of staying successful is to find out and meet the challenges and success factors and concentrate on success factors. If the organization be able to meet this requirement and predicting it properly, the organization can become more productive for stakeholders and as a result, it will become more accomplished. This goal is possible by means of adapting agile development methodology and concentrating on its success factors.

In order compare various ideas about success factors of agile methodology with potential reason for problem and success of agile software development. The second purpose of this research is to develop some contribution about this previous research study methodology in system development process, through figuring out the practical agile success factors roles during implementation of agile methodology are (i) To identify various high points of agile methodology over traditional software development methodologies, (ii) To identify various success factors involved in agile software development, (iii) To identify applicability of success factors in Indian software development organization working in agile methodologies.

REFERENCES

- [1]. Boehm, B. and Turner, R. (2004). *Balancing Agility and Discipline: A guide for the perplexed*, Addison-Wesley, USA, first edition, pp. 165–19.
- [2]. Bosghossian, Z.J., (2002). An investigation into the critical success factors of software development process, time, and quality, Ph.D. Thesis, Pepperdine University, Malibu, California.
- [3]. Bullen, C.V., Rockhart, J.F., (1981). A primer on critical success factors (Working Paper No. 69), Massachusetts Institute of Technology, Sloan School of Management, Center for Information Systems Research, Cambridge, Massachusetts.
- [4]. Bytheway, A.J., (1999). Successful software projects and how to achieve them. *IEEE Software*, **16**(3), 15–17
- [5]. Chaten Kumar, P., & Muthu, R. (2009). Agile maturity model (AMM): A software process improvement framework for agile software development practices. *International Journal of Software Engineering*, **2**(1), 3-28.
- [6]. Chow, T., & Cao, D. B. (2008). A survey study of critical success factors in agile software projects. *Journal of Systems and Software*, **81**(6), 961-971.
- [7]. Conboy, K., Fitzgerald, B. (2004). Toward a conceptual framework of agile methods: A study of agility in different disciplines. In: *Proceedings of the 2004 ACM Workshop on interdisciplinary Software Engineering Research*, pp. 37-44.
- [8]. Fowler, M.: The Agile Manifesto: where it came from and where it may go, <http://martinfowler.com/articles/agileStory.html> (2002).
- [9]. Gunasekaran, A.: Agile manufacturing: a framework for research and development. *International Journal of Production Economics*. Vol. **62**, pp. 87–105 (1999).
- [10]. Highsmith J.: (2002). *Agile Software Development Ecosystems*, Addison-Wesley, Boston, MA..
- [11]. Highsmith, J. (2003). *Agile project management: Principles and Tools*. Agile Project Management Advisory Service Executive Report.
- [12]. Highsmith, J., (2002). *Agile Software Development Ecosystems*. Addison- Wesley, Boston, Massachusetts.
- [13]. Highsmith, J., Cockburn, A., 2001. Agile software development. 1. The business of innovation. *IEEE Computer* **34**, 120–127.
- [14]. Iacovelli A., Souveyer C.: Framework for Agile Method Classification. In: *Proceedings of Model Driven Information System Driven Engineering- Enterprise, User and System Model (MoDISE –EUS)*, pp. 91-102 (2008).
- [15]. Iivari, J., Hirschheim, R., Klein, H. K.: A Dynamic Framework for Classifying Information Systems Development Methodologies and Approaches. *J. of Management Information Systems*. vol. **17**(3), pp 179-218 (2001).
- [16]. J. Highsmith and A. Cockburn, “Agile Software Development: The People Factor”, *IEEE Computer*, <http://www.jimhighsmith.com/articles/IEEEArticle2Final.pdf> Accessed on 10/10/2014.
- [17]. M. Fowler, “The New Methodology,” <http://www.martinfowler.com/articles/newMethodology.html> Accessed on 12/12/201.
- [18]. Medinilla, Á. (2012). *Agile Management: Leadership in an Agile Environment*. Springer.
- [19]. Misra, S. C., Kumar, V., & Kumar, U. (2006). Success Factors of Agile Software Development. In *Software Engineering Research and Practice*, 233-239.
- [20]. Misra, S. C., Kumar, V., & Kumar, U. (2009). Identifying some important success factors in adopting agile software development practices. *Journal of Systems and Software*, **82**(11), 1869-1890.
- [21]. Nerur, S., Mahapatra, R., Mangalaraj, G.: Challenges of migrating to agile methodologies.
- [22]. Rockhart, J.F., Crescenzi, A.D., (1984). Engaging top management in information technology. *Sloan Management Review*, **25**(4), 3–16.
- [23]. The Standish Group International, “The CHAOS Report ” http://www.standishgroup.com/sample_research/chaos_1994_1.php Accessed on 2/2/2005.