International Journal of Electrical, Electronics and Computer Engineering **5**(1): 1-8(2016) **ISSN No. (Online): 2277-2626**

A Quantitative Study on Critical Success Factors in Agile Software Development Projects

Navneet Kaur and Gurpreet Singh
Department of Computer Engineering,
UCOE, Punjabi University Patiala, (PB), INDIA

(Corresponding author: Navneet Kaur)

(Received 03 December, 2015 Accepted 12 February, 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 heavy weight behind and leads to the need for different approaches. Modern approach is called agile or lightweight. Heavy weight methodologies, also considered as the traditional way to develop software, emphasis on comprehensive planning, detailed documentation, and expansive design. Unlike traditional methods, agile methodologies employ short iterative cycles, and rely on tacit knowledge within a team. What is new about agile methods is not the practices they use, but their recognition of people as the main driving force which can lead to project success. Heavy weight methodologies, also considered as the traditional way to develop software, emphasis on comprehensive planning, detailed documentation, and expansive design.

Key words: 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, Mahapatra & Mangalaraj, 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 continuously maintenance support or other software services and fine release (Chow, T.,& Cao, D.B., 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. 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'.

As a result of this meeting, the "Agile Alliance" was formed and its manifesto for agile software development emerged with features which the methodologists felt significant over some other features:

-Individuals and interactions over processes and tools.
-Working software over comprehensive documentation.
-Customer collaboration over contract negotiation.
-Responding to change over following a plan.

II. AGILE SOFTWARE DEVELOPMENT

What is the meaning of being agile? Jim Highsmith says that 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) There is not universally accepted definition of agility "Agility is dynamic, context-specific, aggressively change embracing, and growth-oriented" (Goldman *et al.,* 1995). The core concept in agile is quick response to change. Conboy and Fitzgerald (2004) carried out a review of the literature on agility across several disciplines and provide a broad definition of agility as "the continual readiness of an entity to rapidly or inherently, proactively or reactively, embrace change, through high quality, simplistic, economical components and relationships with its environment" . Despite the differences, all dentitions of ''agility'' emphasize the speed and flexibility as the primary attributes of an agile organization (Gunasekaran, 1999).

Agile methodologies are often touted as the solution to the ills created by software crisis and Methodology Movement Claim of agile proponents have been supported by The Standish report in 2012, which has earlier revealed that software crisis is still very much alive. The report claimed that only 9% of agile projects failed as compared to 29% of waterfall model project.

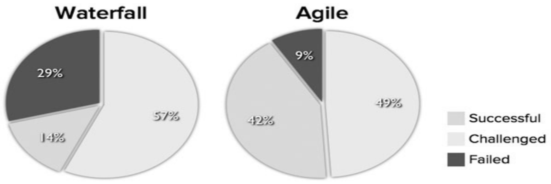

Fig. 1. Project Statistics: Agile Vs Traditional.

III. CHARACTERISTICS OF AGILE METHODOLOGIES

1. People Oriented- Agile methodologies consider people ,customers ,developers , stakeholders, and end users – as the most important factor of software methodologies. As Jim Highsmith and Alistair Cockburn state, "The most important implication to managers working in the agile manner is that it places more emphasis on people factors in the project: amicability, talent, skill, and communication" If the people on the project are good enough, they can use almost any process and accomplish their assignment. If they are not good enough, no process will repair their inadequacy (Highsmith, 2000). As Highsmith highlights, " people trump process"

2. Adaptive – The participants in an agile process are not afraid of change. Agilist welcome changes at all stages of the project. They view changes to the requirements as good things, because they mean that the team has learned more about what it will take to satisfy the market. Today the challenge is not stopping change but rather determining how to better handle changes that occur throughout a project. "External Environment changes cause critical variations. Because we cannot eliminate these changes, driving down the cost of responding to them is the only viable strategy" [24].

3. Conformance to Actual – Agile methodologies value conformance to the actual results as opposed to conformance to the detailed plan. Highsmith states, "Agile projects are not controlled by conformance to plan but by conformance to the business value" (Highsmith, 2002). Each iteration or development cycle adds business value to the ongoing product. For agilists, the decision on whether business value has been added or not is not given by the developers but instead by end users and customers.

4. Balancing Flexibility and Planning – Plans are important, but the problem is that software projects can not be accurately predicted far into the future, because there are so many variables to take into account. A better planning strategy is to make detailed plans for the next few weeks, very rough plans for the next few months, and extremely crude plans beyond that . In this view one of the main sources of complexity is the irreversibility of decisions. If you can easily change your decisions, this means it's less important to get them right – which makes your life much simpler. The consequence for agile design is that designers need to think about how they can avoid irreversibility in their decisions. Rather than trying to get the right decision now, look for a way to either put off the decision until later or make the decision in such a way that you will be able to reverse it later on without too much difficulty (Highsmith, 2002).

5. Empirical Process – Agile methods develop software as an empirical (or nonlinear) process. In engineering, processes are either defined or empirical. In other words, defined process is one that can be started and allowed to run to completion producing the same results every time. In software development it can not be considered a defined process because too much change occurs during the time that the team is developing the product. Laurie Williams states, "It is highly unlikely that any set of predefined steps will lead to a desirable, predictable outcome because requirements change technology changes, people are added and taken off the team, and so on" (Williams and A. Cockburn, 2003).

6. Decentralized Approach – Integrating a decentralized management style can severely impact a software project because it could save a lot of time than an autocratic management process. Agile software development spreads out the decision making to the developers. This does not mean that the developers take on the role of management. Management is still needed to remove roadblocks standing in the way of progress. However management recognizes the expertise of the technical team to make technical decisions without their permission.

7. Simplicity – Agile teams always take the simplest path that is consistent with their goals. Fowler states, "They (agile teams) don't anticipate tomorrow's problems and try to defend against them today". The reason for simplicity is so that it will be easy to change the design if needed on a later date. Never produce more than what is necessary and never produce documents attempting to predict the future as documents will become outdated. "The larger the amount of documentation becomes, the more effort is needed to find the required information, and the more effort is needed to keep the information up to date" [Wendorff, 2003].

8. Collaboration – Agile methods involve customer feedback on a regular and frequent basis. The customer of the software works closely with the development team, providing frequent feedback on their efforts. As well, constant collaboration between agile team members is essential. Due to the decentralized approach of the agile methods, collaboration encourages discussion. As Martin Fowler describes, "Agile teams cannot exist with occasional communication. They need continuous access to business expertise" [Fowler, 2002].

9. Small Self-organizing teams – An agile team is a self organizing team. Responsibilities are communicated to the team as a whole, and the team determines the best way to fulfill them. Agile teams discuss and communicate together on all aspects of the project. That is why agility works well in small teams. As Alistair Cockburn and Jim Highsmith highlight, "Agile development is more difficult with larger teams. The average project has only nine people, within the reach of most basic agile processes. Nevertheless, it is interesting to occasionally find successful agile projects with 120 or even 250 people" [Highsmith, Cockburm, 2002].

IV. LITERATURE REVIEW

There are few attempts to classify agile methods within the body of systems development methodologies. In 2000, Iivari *et al.* published a four-tier framework for classifying information systems development methodologies (ISDM). The framework divides information systems methodologies into four paradigms: functionalism (objective-order), social relativism (subjective-order), radical structuralism (objective-conflict) and neo-humanism (subjective-conflict). Authors in 2004 again published a framework. This time they inserted agile methods into the framework as a separate approach belonging to the functionalist paradigm. This paradigm views methods, people, hardware, software, rules (organizational procedures) as formal, objective entities and assumes that systems development is achieved using formal concepts, planned intervention, and rationalistic tools (Hirschheim & Klein, 1989).

The frameworks lack at two fronts. The first is that agile methods are treated as a distinct approach when there is clearly a high degree of overlap between the characteristics of object-oriented approaches and the agile approaches, for example, both share the principle of "iterative and incremental development" (Iivari *et al.,* 2001). The overlap between approaches are not shown in the model provided by Ivarii *et al.* (2004). Another criticism is that the definition of agile methods is based on a single analysis of agile methods that are carried out by Abrahamsson *et al.* (2003). But in case of agile methods, there is a lot of variation in procedures and practices used by different methodologies, some are only for management of project (Scrum), some are designed for development process (XP). These all aspects are not considered by the author.

A different way of classifying the agile methods was proposed by Conboy and Fitzgerald (2004). They classified agile methods by first defining the agility because they argue that there is no formal definition of 'agility' in information systems and "no consensus as to what constitutes an agile method, either in academia or in industry" (Conboy & Fitzgerald, 2004). They state that the agile manifesto (Agile Alliance, 2001) is not an appropriate set of criteria for defining agility, because each of the commonly known agile methods partially fulfils the principles of the manifesto (Visconti & Cook, 2004). They described a conceptual framework for agility. The authors recognized change as the main reason why organizations are adopting agile and they isolated the sources of change to customer, competition, technology, and social factors. Other categories in their framework were the agility strategy, agility capabilities and agility providers. The framework is used to assess the ability of a team to be agile and if the team is 'agile' then only team can successfully implement agile practices. There is no actual assessment of the framework against real organizations, projects or teams and the framework assesses projects rather than methods.

Iacovelli and Souveye (2008) try to classify methods through four views. According to the authors these four views are the decomposition of an agile method: Usage view, Process and Product view, Applicability view, Capability to Agility view. Usage view captures why using the agile method, this view tries to evaluate all the benefits that the development team and the customer can gain by applying the method. The capability to agility view represents what is the part of agility in the method, how agile is the method. This view represents all the aspects of the agility concept. Applicability view explains the impact of environmental aspects on the method. It represents when the environment is favorable to apply the agile method. The process and product view represents the activities performed in the process.

V. 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 and Rockhart, 1981; Rockhart and Crescenzi,1984). Critical Success Factor is explained by Bullen. C.V 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).

VI. RESEARCH METHODOLOGY

According to the research objective the whole process of gathering essential data and preparing them for analysis is called data collection.

The research in this thesis is a pivot research conducted on Indian software industry working in agile methodologies. This multi-disciplinary research is based on experimental and empirical software engineering. To achieve the ultimate goal to find out critical success of agile methodologies we undertook is literature survey and empirical investigation through survey. A thorough literature survey and exploratory case study have been used to identify the success factors of software engineering organizations which are using agile practices for software development. Survey conducted for research purposes possesses three distinct characteristics (Fowler, 2000). First, the survey produces quantitative descriptions of some aspects of studied population. Second, the main method of collecting information is by asking people structured and predefined questions. Third, information is generally collected about a fraction of the population under study but it is collected in such a way so as to be able to generalize the findings to the population

The demographic characteristics of the study are illustrated in this section.

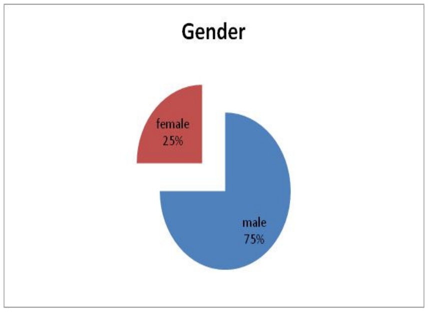

Fig. 2. Gender Demography .

shows that male employees comprise the majority of staff: 75.00 per cent of this company which use agile process. It can be clearly seen that mainly men prefer to work in the software development company. However, there were also 25.00 per cent women, making up a comparatively low proportion of employees.

Figure 3.2 also shows that approximately 40 per cent of total employees in company have around 5 years work experience and know agile process for around 4 years (c.f.). Indeed the largest groups of employees are in the age groups 35-44 and 25-34 with approximate 30.00 per cent of employees in each group. Smallest group belongs to the age groups 18-24 and 50+ groups.

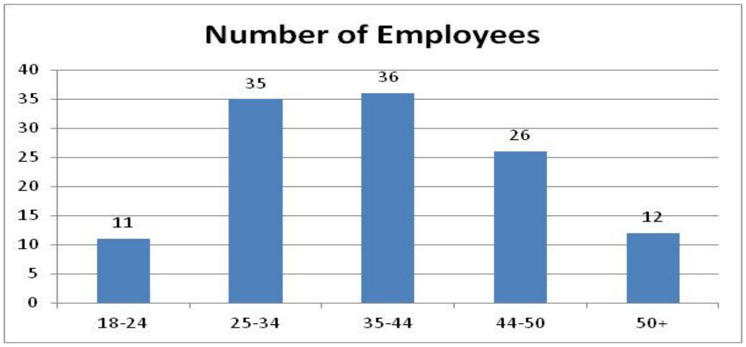

Fig. 3. Shows the work experiences distribution of employees in this company.

VII. SURVEY ANALYSIS

A through analysis is performed on the data collected from the sample respondents. The following section describes statistical analysis performed and testing of hypothesis for the study. Summary of the data gather for the sample questions have been presented here. The goal of this chapter is to show the results of the survey and to provide a simple summary of the responses.

Question 1: How do you estimate your understanding of Agile Methodologies?
First question deals with the understanding of the agile methods means companies are actually following the right way of writing the code. The fig 4 shown upper average range is more than 50 percent of employees.

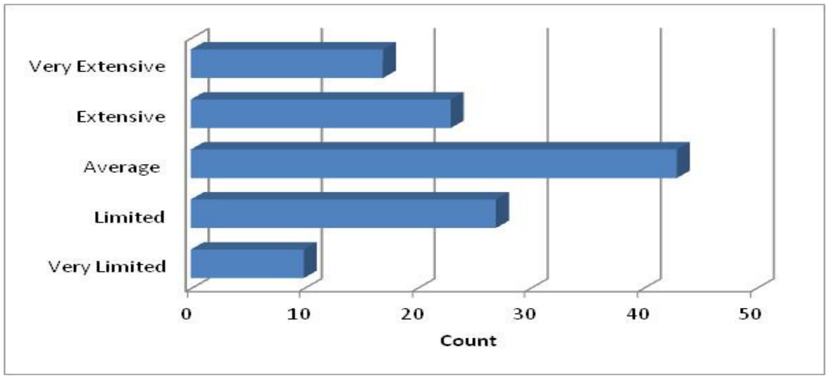

Fig. 4. Way of writing code.

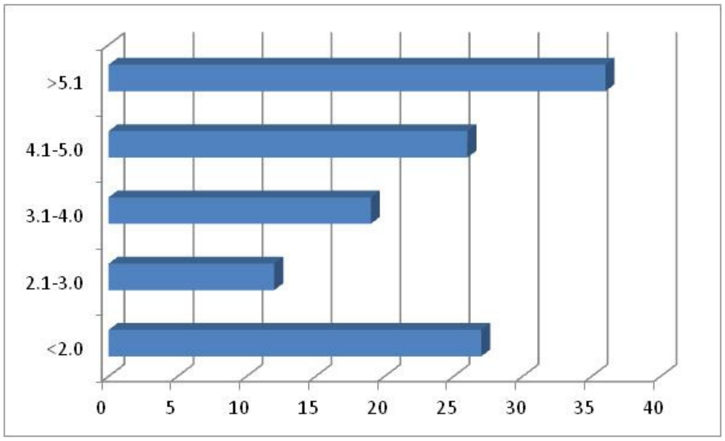

Fig. 5. More than 3 years experience.

Question 2: How many years have practical experiences in agile methodology? Second question is compliment to the first question. Results of this questions solid the claim of the first question by their work experiences in agile process in software development. This fig 5 show that more than 50 percent of employees have over 3 years' experience in this filed.

Question 3: Has adoption of management commitment in organization dimension of agile process effect delivering a good working product? Although direct measurement of this question is very difficult as a result majority of people claim that hardly any effect of management commitment. Rather it is better to say that majority of people have a positive feeling for the effect of management commitment on good working project.

Question 4: Has agile logistical arrangements in organization dimension of agile process effect delivering a good working product?

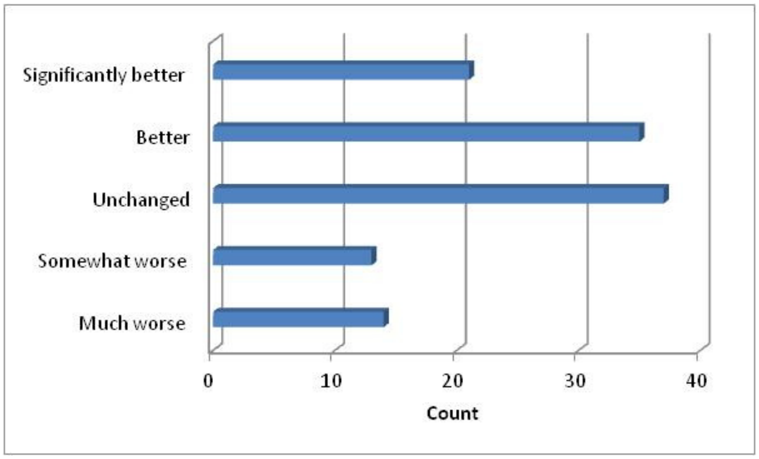

Fig. 6. Management commitment.

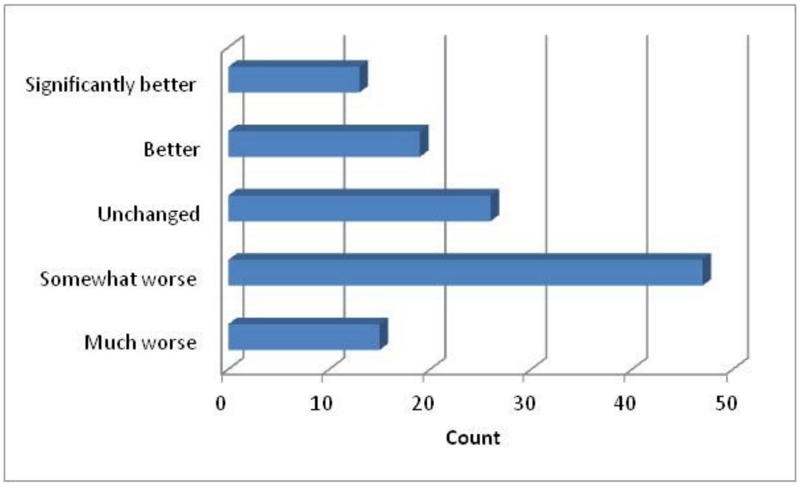

Fig. 7. Delivering a working product.

In this question we believe that it can make relation between organization factor and quality of products. This question will estimate success or failure of first hypothesis which indicate in previous chapter.

Question 5: Has management commitment in organization dimension of agile process effect total estimated cost and effort?

This question clearly states that the relation between organization in agile process and total cost of projects. Mixed responses for all categories have been seen and it is not possible to comment on statement without further statistical analysis.

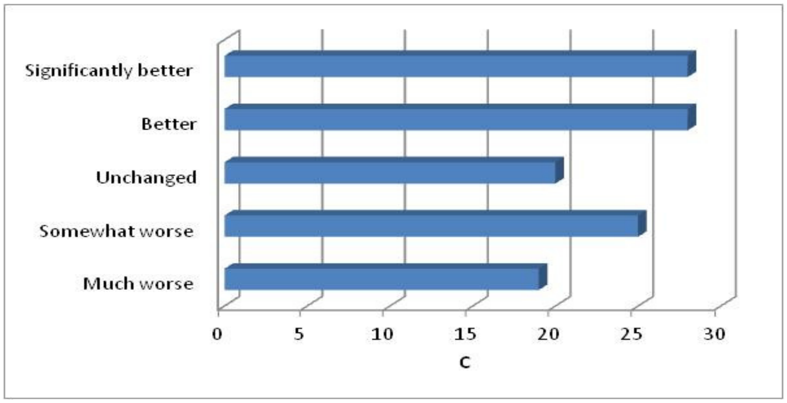

Fig. 8. Total estimated cost and effort.

VIII. CONCLUSION

The main aim of performing this study has been to investigate and understand the relationship between the different success factors of agile software development methodologies
1 Team environment in organizational subject
2 Team capability in people subject
3. Project management process in process subject & Agile software engineering in technical subject
4 Scheduler in project dimension subject
In terms of:
□ Quality □ Scope □ Timeless □ Cost
Further, the inclusion of agile methodology characteristics (i.e., perceived usefulness, perceived ease of use, perceived compatibility, result demonstrability, and perceived maturity) can add to the understanding of acceptance of agile methodology. This conceptual framework can have both theoretical and practical contributions. In terms of theoretical contributions, the results build on a knowledge management perspective can provide a fresh view synthesizing various factors that can potentially influence acceptance of agile methodologies. In terms of practical contributions, the framework can consolidate current knowledge on acceptance of agile methodologies which can provide guidance to organizations interested in getting their developers to use these new methodologies.

REFERENCES

[1]. Abrahamsson, P., Warsta, J., Siponen, M.T., Ronkainen, J.: New directions on agile methods A comparative analysis. In: *International Conference in Software Engineering* (ICSE) 2003. pp. 244-254 (2003).

[2]. Arisholm, E., Gallis, H., Dyba, T., Sjoberg, D.I.K., (2007). Evaluating pair programming with respect to system complexity and programmer expertise. *IEEE Transactions on Software Engineering*, **33**, 65–86.

[3]. Beck K., Cockburn A., Jeffries R., Highsmith J.: Agile manifesto. http://www.agilemanifesto.org, pp. 12-4 (2001)

[4]. Beck, K., Andres, C., 2004. Extreme Programming Explained: Embrace Change. Addison-Wesley Professional, Boston, MA.

[5]. Beck, K., Boehm, B.: Agility through Discipline: A Debate. *Computer.* vol. **36**(6), pp. 44-46 (2003)

[6]. Boehm, B. and Turner, R.: Balancing Agility and Discipline: A guide for the perplexed, Addison–Wesley, USA, first edition, pp. 165–19 (2004)

[7]. Boehm, B., Turner, R., 2003. Using risk to balance agile and plan-driven methods. *Computer* **36** (6), 57–66.

[8]. 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.

[9]. 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.

[10]. Bytheway, A.J., 1999. Successful software projects and how to achieve them. *IEEE Software,* **16** (3), 15–17

[11]. Cao, L., Mohan, K., Xu, P., & Ramesh, B. (2004). How extreme does extreme programming

[12]. 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.

[13]. 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.

[14]. Cohen, D., Lindvall, M., Costa, P., (2004). An introduction to agile methods. In: Zelkowitz, M.V. (Ed.), Advances in Computers, Advances in Software Engineering. Elsevier, Amsterdam.

[15]. Conboy, K., Fitzgerald, B.: 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 (2004)

[16]. Dulewicz, V. & Higgs, M.J. (2003). Design of a new instrument to assess leadership dimensions and styles. Reading, UK, University of Reading (Henley Working Paper Series, (HWP 0311)).

[17]. Fowler, M.: The Agile Manifesto: where it came from and where it may go, http://martinfowler.com/articles/agileStory.html (2002)

[18]. Goldman, S.L., Nagel, R.N., Preiss K.: Agile Competitors and Virtual Organizations: Strategies for Enriching the Customer. Van Nostrand Reinhold (1995)

[19]. Gunasekaran, A.: Agile manufacturing: a framework for research and development. *International Journal of Production Economics.* vol. **62**, pp. 87–105 (1999)

[20]. Hackman, J.R. (1990). Groups that work (and those that don't). San Francisco: Jossey-Bass. A have to be? Adapting XP practices to large-scale projects. In System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on (pp. 10-pp). IEEE

[21]. Highsmith J.: Agile Software Development Ecosystems, Addison–Wesley, Boston, MA, (2002)

[22]. Highsmith, J. (2003). Agile project management: Principles and Tools. Agile Project Management Advisory Service Executive Report.

[23]. Highsmith, J., 2002. Agile Software Development Ecosystems. Addison- Wesley, Boston, Massachusetts.

[24]. Highsmith, J., Cockburn, A., 2001. Agile software development. 1. The business of innovation. *IEEE Computer,* **34**, 120–127.

[25]. Hoegl, M. & Gemuenden, H. G. (2001). Teamwork quality and the success of innovative projects: A theoretical concept and empirical evidence. *Organization Science,* **12**(4), 435-449

[26]. 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)

[27]. 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)

[28]. 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

[29]. Jones, M. C., & Harrison, A. W. (1996). IS project team performance: an empirical assessment. Information & Management, *31*(2), 57-65.

[30]. Koskela, T.T. V.: Software configuration management in agile methods, VTT Technical Research Centre of Finland (2003)

[31]. M. Fowler, "The New Methodology," http://www.martinfowler.com/articles/newMethodology.html Accessed on 12/12/201

[32]. Medinilla, Á. (2012). Agile Management: Leadership in an Agile Environment. Springer.

[33]. Melnik, G., & Maurer, F. (2004). Direct verbal communication as a catalyst of agile knowledge sharing. In Agile Development Conference*, 2004,* IEEE, 21-31.

[34]. Misra, S. C., Kumar, V., & Kumar, U. (2006). Success Factors of Agile Software Development. In Software Engineering Research and Practice, 233-239.

[35]. 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.

[36]. Nerur, S., Mahapatra, R., Mangalaraj, G.: Challenges of migrating to agile methodologies.

[37]. Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P., & Still, J. (2008). The impact of agile practices on communication in software development. Empirical Software Engineering, 13(3), 303-337.

[38]. Rockhart, J.F., Crescenzi, A.D., (1984). Engaging top management in information technology. Sloan Management Review 25 (4), 3–16.

[39]. The Standish Group International, "The CHAOS Report ," http://www.standishgroup.com/sample_research/chaos_1994_1.php Accessed on 2/2/2005

[40]. Visconti, M., Cook, C.R.: An Ideal Process Model for Agile Methods. In: PROFES 2004. Pp 431-441 (2004).