Vol. 2 No. 1, 2016
ISSN 2412-303X

International Interdisciplinary Journal of Scholarly Research (IIJSR)

Enhancement in Agile Development Methodologies Using Extreme Programming to Overcome Large Scale Firm's Implementation Problem

Muhammad Zeeshan Ali
MS in Computer Science, University of Agriculture, Faisalabad, Pakistan
Tel: +92-0332-7155977 Email: mazeeshi@gmail.com

Abstract

Extreme programming is one of the most effective and commonly used agile development methodology among software development methodologies. But in extreme programming, quality activities were executed consecutively along with the functional requirements. It reduces the rapidity in continuous iterations and makes inverse relation between quality and agility. Agile development worked with the small scale projects along with few experts team but not for large project. It could not be implemented properly due to scaling and estimation problem. Extreme programming (XP) with enhancing capabilities enabled agile methods to work efficiently for large scale organizations. XP was best suitable to be documentation and easier to implement. System engineering approach was considered as incremental change related to coding. Extreme programming was enabling hundreds of programmers and software engineers to work on large projects using agile development methodologies.

Keywords: Estimation, Scaling, Extreme Programming, Hierarchical approach, Process Measurement

1. Introduction

As agile methodologies become more famous as the "modern" change of the traditional methodologies like waterfall model, but its accomplished and distributed roots go back years (Abdullah & Abdelsatir, 2011). Software engineering was a highly versatile practice and can take many forms from small tasks to huge projects, lasting for years (Alshehri & Benedicenti, 2013). Technologies, complexity, criticality and business domains may vary greatly. However, one piece is always a part of the game recognized as customers. It may be a single person, an organization or a larger diverse group. As there were many theories, strategies and methodologies (Asif, 2015) for software engineering, there are many ways of engaging the customers.

The best practice of extreme programming is in telecommunication sector where changing requirements and enhancement in functionality could must be done under pressure (Dixon & Dudman, 2004). The tough competition among mobile companies required fast and reliable solutions for their large projects. A process model that could be useful for both developer as well as customer which initiate the concept of extreme programming. But due to lack of necessary documentation, it was helpful to communicate developers with each other and established interfaces. Designing, coding, testing and deployment are the major activities during any software development (Henderson & Sellers, 2003). Due to light weight approach, XP eliminate the documentation and requirement analysis in detail. Therefore, documentation should not consider as an essential part because our main focus in agile development is on requirement specification with small but continuously iterations.

During development, no one from the team knows the direction of development even about the final product. We would pick up the test cases near to original and implement them on software which were later recognized as final acceptance test for working. Roles of users could help to identify the test cases according to their skills and goals that they want to achieve.

When proper acceptance and quality assertion had to occur, test-cases could be "fleshed up" to cover precise references to the real executed user interface along with correct test data. Our observation would must be accurate at the finished product due to test-driven development and individual testing. In iterative development, either we don't know what the end product will be but must have confidence about its accuracy. Continuing on a project without knowing what the people roles were for the product and what actions they needed to perform, now feels as risky as writing code without unit tests. This paper basically consist of five portions which are discussed as the 2nd portion of this paper describes the initiation of extreme programming among different organizations. 3rd portion stated the management of agile development methodologies over large projects. 4th portion describes the impact of moderation applying on extreme programming. 5th portion describes the results of implementing hierarchical structure and moderation concept to large scale organizations. The last portion contain the conclusion of overall research.

Vol. 2 No. 1, 2016
ISSN 2412-303X

International Interdisciplinary Journal of Scholarly Research (IIJSR)

1. Initiation of extreme programming

Extreme Programming (XP) is the most noticeable among new generation of light-weight software development models as it has small teams to develop software with changing requirements (Laubacher & Malone, 2002). Extreme programming also proved to be the solution of complex problem with randomly changing requirements same like rational unified process and open modeling languages. A traditional methodology doesn't work on small iterations as compared to extreme programming technique by following iterative and agile development (Musa & Norwawi, 2011). XP is basically design for small sized iterations with the help of 10-12 persons team.

The weakness of XP is to use it for large scale organization. The best practice of Extreme Programming is in telecommunication sector where changing requirements and enhancement in functionality could must be done under pressure (P. Meso & Jain, 2006). The tough competition among mobile companies requires fast and reliable solutions for their large projects (Qumer, 2008).

A process model that could be useful both by developer and customer which initiate the concept of extreme programming. But due to lack of necessary documentation related to development that was helpful to communicate developers with each other and established interfaces. Therefore, scaling up of XP will perhaps crucial in the way to adopt systematic practices from development models. By discussing about software engineering, basically three approaches are appropriately conducted to reorganizing a project. First is total system approach in which all the requirement are already mentioned and then start implementation that will be introduce a new organization.

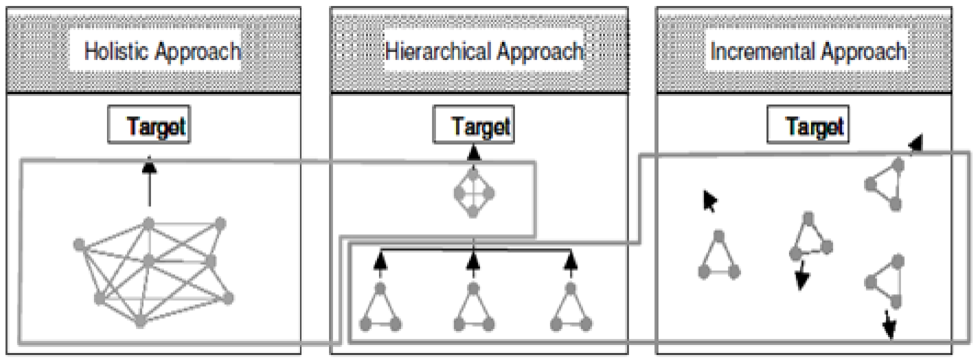


Figure 2.1. Agile Development Method Approaches

Second approach is incremental system approach in which changes could be made along with each iteration that leads to final optimization of project as shown in figure 2.1. These small iterations and evolution in changes steadily define the internal structure of the organization and leads to optimal final project (Richard P, 2005). These both approaches couldn't found successive. Another approach as the combination of both above discussed approach is defined which are hierarchical structuring or mixed scanning. This approach provides the better rearranging of projects and their optimization with the passage of time. By combining iterative and evolutionary development with extreme programming, the above approaches shifted to software engineering method to software discipline which contain the following advantages,

- Due to hierarchical structuring of teams in larger projects, it enables the extreme programming to scale up for large organization.
- It provides a successful approach for the firm of the classified approach which can be shifted to the software engineering discipline.

By scaling up the extreme programming for large projects, more people are required to complete project requirements even project become more complex. Another major benefit is to reorganize project and also enhanced hierarchical methodology that describe the discipline of software engineering.

Vol. 2 No. 1, 2016
ISSN 2412-303X

International Interdisciplinary Journal of Scholarly Research (IIJSR)

2. Large scale project management

Eextreme programming requires main focus on team in hierarchical approach in which project management handle the issues arises related to steering committee. The basic concern of project management is to divide the whole project into small iterations and define work breakdown structure (WBS) which could not be practiced in reality for controlling and monitoring the projects. Earlier, the project management works on the running projects but extreme programming contradict it by requiring intensive and straightforward project modeling and designing before the project properly starts. Success of every project using extreme programming could be done only on the behalf of communication and early feedback (Rycroft *et al*, 2004). Both communication and response from customer over project become tougher to get it with every single further team member to elaborate in the team.

We found the solution of how extreme programming could be enabled for every size of team either it is large or small with the help of communication and early response. The success of project based on the project manager (Shahriar & Sohrabi, 2009). In extreme programming, project manager is not only the project supervisor or chief handler but work as a moderator among team members and consumers. Project manager shifted the large project responsibilities to each team member individually.

3. The Effect of Applying Moderation on Extreme Programming

Project manager in extreme programming could also work as an intermediate and moderator person who communicates with the whole team as well as developers and steering committee (T.S.Ramya Krishna, 2011). It only works to control and monitor the team management and took iteration in sequential and timely manner. Moderation is a complete separate job in software development in large scale organization to develop software using Extreme Programming.

All moderators participate in a specific large scale XP projects meet each other throughout the whole mean day in leaning way to exchange project knowledge from their teams (Vidgen. 2006). It is feasible to provision the team of developers by project managers. Each manager of a team will be responsible to their assigned responsibility along with acceptance testing. Moreover, development team and testing team should meet in total during every four weeks in order to clarify, discuss, and solve programming techniques, coding, designing and testing. These discussions and efforts also should offer a policy for know-how exchange (Syed-Abdullah, Holcombe & George, 2007.).

4.1 Actions Performed by Moderator

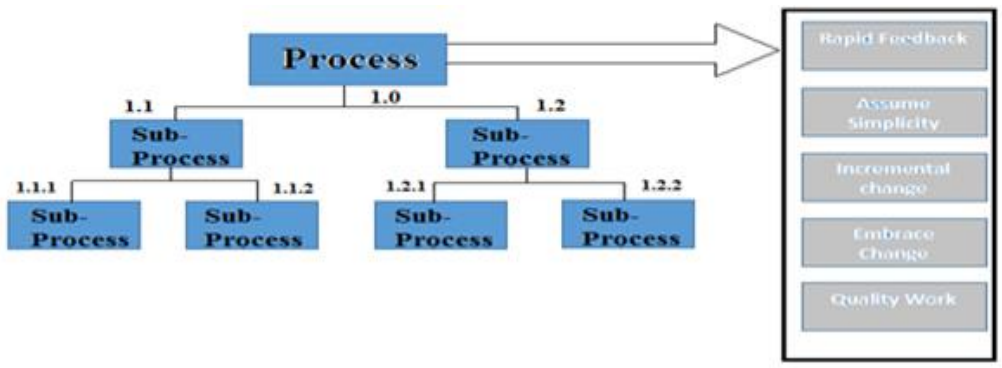
Project manager support their team in a way that they complete their tasks by themselves. Feedback and enhancement could be positively impact with respect to software efficiency. The results of the project could be acceptable by both developers as well as manager. Project manager must have some technical and strategic knowledge about the software development related to coding and implementation through the guide, the developers will work (Shahriar & Sahar, 2009). A good project manager must have technical knowledge and completely understanding about Extreme Programming (Salo & Abrahamsson, 2007) use. He must be able to define the structure and iterative manner of project by dividing whole project in sub systems. At the end, he or she takes precaution that the probability of each project member can be shattered in an ideal way. Project manager manage the team efficiently with the help of communication, coordination and convincing (Richard, 2005). By using hierarchical approach, sub teams will be decided even during development that enables the organization to manage them in appropriate manner.

4. Hierarchical Structuring in XP

The hierarchical structuring in extreme programming is the combination of total system approach and incremental system approach (Neil & Robert, 2015) as shown in fig 5.1.

Vol. 2 No. 1, 2016
ISSN 2412-303X

International Interdisciplinary Journal of Scholarly Research (IIJSR)



5.1. Hierarchical Structuring along with effective factors

As agile development known as light weighted methodology, it basically consist of some values which are,

- Allow the programmers and customers to communicate with each other, easiness in design, and response over product feedback encourage the programmers to do that seem to be necessary.
- Four steps of product development cycle included designing, coding, testing, listening and their sequence relevant to development practices.
-

5.1. Rapid feedback

In extreme programming, feedback response from the development team could be discussed on daily and random basis that enable the developers to focus on special feature of software. Iterations are very short but maximum team coordination of developers and customers causes very low percentage of failure.

5.2. Assumed simplicity

Extreme programming basically focus on simple and straightforward designing which could be easily programmed. XP doesn't focus on future changes but fulfil the today's customer requirement completely and more accurately. In other words, it couldn't plan to reuse.

5.3. Incremental change

Iterative and agile development using extreme programming doesn't allow large change consecutively. It accommodates changes in short iterations with incremental changes to fulfil the final specification of the product. This idea could be initialized by refactoring.

5.4. Embracing change:

The best approach is the one that reserves many choices while truly solving your most persistent problem. Mostly unspecified problem will be realize that never accommodate that much quickly. Extreme programming contains many alternate solutions as a backup plan for randomly change requirements.

5.5. Quality work

The most desirable thing throughout the whole development is quality which is the most important part of the successful development. Extreme programming basically focuses on the major and initial requirements to enhance the maximum quality. It also motivates the programmers to raise it much more they can. Designing, coding, testing and deployment are the major activities during any software development. Due to light weight approach, XP eliminate the documentation and requirement analysis in detail (Maruping & Venkatesh, 2009). Therefore, documentation should not consider as an essential part because our main focus in agile development is on requirement specification with small but continuously iterations. Coding is the major part of the development as compared to interface designing. XP ensure the quality of product and their accuracy with the help of Junit framework (Lawrence & Yslas, 2006). End user or customer could check the software according to functional and essential requirements. XP developers understand the need of customers in daily coding and but this wouldn't a daily routine. It will works only in complicated condition of requirements.

Vol. 2 No. 1, 2016
ISSN 2412-303X

International Interdisciplinary Journal of Scholarly Research (IIJSR)

6. Implementation of Moderation Concept to Large Scale Organization

Mostly in extreme programming to provide efficient work, a project manager could handle 6 to 8 pair of developers, which will be according to our research is able to cover three teams of large projects. But as we concerned with the problem that, how extreme programming will be helpful for 100 or more than 100 developers. The solution is, the whole firm is divided into sub teams and every team having 6 to 8 pair of developers along with project manager. Each project manager will be responsible for the progress, performance and quality of its project. This will enable the intra team knowledge sharing.

After changing hierarchies in extreme programming and software development models, some questions were asked from 30 people at random software houses. The detaile of people are describes in table 1.

Role of the Person	N	Role of the Person	N
Programmer	15	Tester	4
Project Manager	3	Owner	2
Customer	3	Marketing Personal	3

Table 1 : Role of the persons and their strength

The expertises of the persons relevant to their fields are described in table 2.

Experience in Years	N	Percentage
Fresh	6	20
1-2 years	12	40
3—5	8	27
6—10	4	13
Total	30	100.0

Table 2 :: Experience of the employee and their strength

The questions that are generally asked from the people are,

- What was the knowledge and expertise level of team members at the beginning?
- How many people involved as development company/individual consultants?
- How much the customer involvement was, during the development and what are the levels of their involvement?

Vol. 2 No. 1, 2016
ISSN 2412-303X

International Interdisciplinary Journal of Scholarly Research (IIJSR)

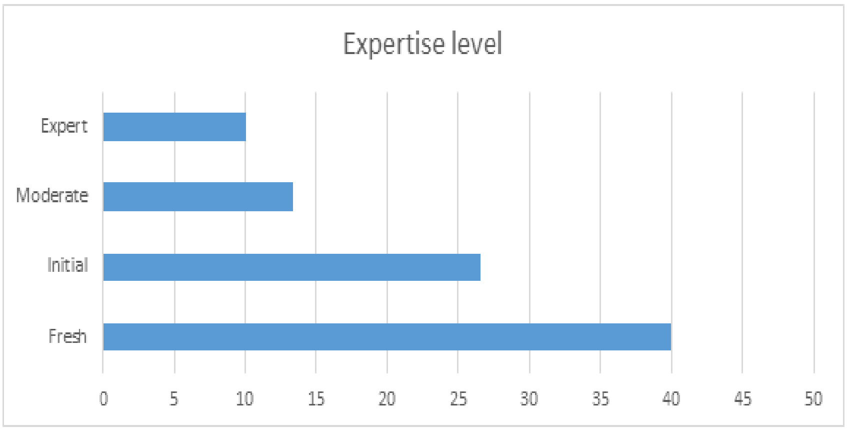


Figure 6.1 Expertise level among software developers at the beginning of firm

According to collected consequences, 40 percent people just take initiate related to development. Only 10% people are most expert in agile development that works as team leader as well as developer. These are the people who already have work experience about the project using XP along with agile development.

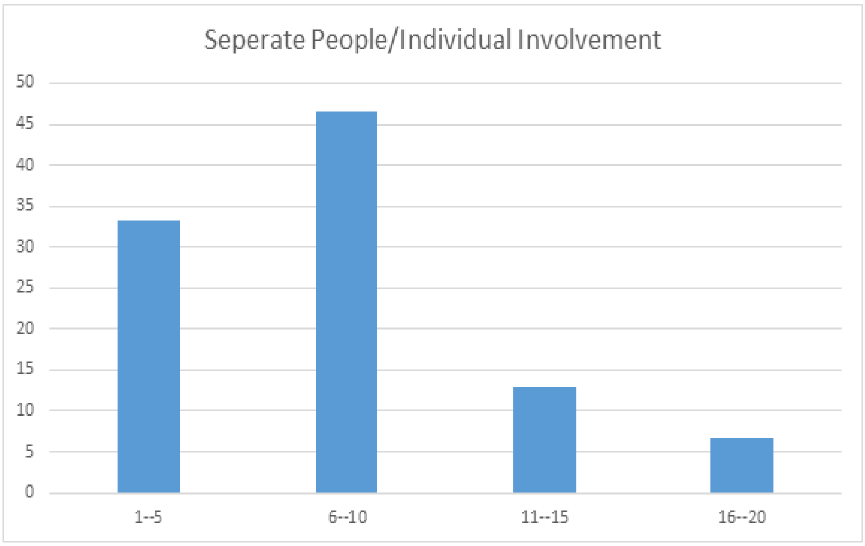


Figure 6.2: Involvement of people as consultants other than the company/Virtual Team

Each organization has its own strategies to achieve a goal. Online project development team hires the individuals or separate team as consultants to improve the software quality, reliability and customization. Figure 6.2 describes that mostly the people involved during software development was 6-10 people.

At least, the project managers act as moderators of the steering committee meet frequently by forming "heures fixe": All moderators participate in a specific large scale XP projects meet each other throughout the whole mean day in leaning way to exchange project knowledge from their teams.

6.1. Estimation and Process Measuring

The managers working on agile methodology uses different methods and techniques to estimate the project cost and number of team members (John & Narti, 2003). They schedule their work in different manners for estimation. Large organizations having their own methods to calculate the risk related to software development, how much time required to complete the whole project, what type of project is, and which type of technologies will be used to achieve this goal.

By implementing the enhancement in Extreme programming provide the following results,

International Interdisciplinary Journal of Scholarly Research (IIJSR)

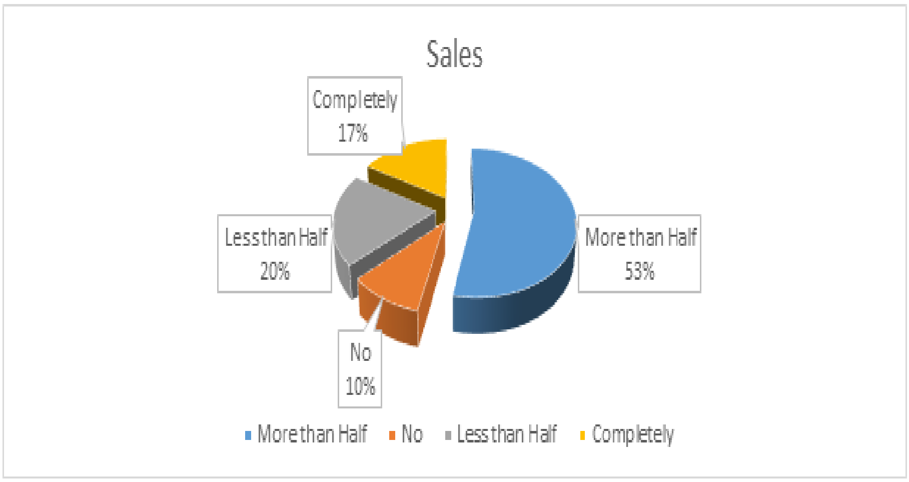


Figure 6.3: Project Successive rate Using Extreme Programming (Agile Development)

By adopting extreme programming, more than 50% projects result successfully completion as shown in figure 6.3 which clearly describes that usage of extreme programming reduces the maximum failure rate. According to conducted research and results, extreme programming along with agile development consider much better for software development for the practical approach

Wrong estimation and process measuring causes project failures and flop the whole organization. Mostly the effort required to complete a goal using agile development method could also calculate the time and cost required to complete a project. In agile development methods, we are unable to calculate the total time required to calculate the resources, cost and also the time required due to randomly updating changes. It is very difficult either impossible to estimate the iterations length and about final product. It could never be done in iterative and incremental manner.

Conclusion

The whole research comprises the enhancement in agile development methodologies and their efficient use to make them implementable on larger scale firms. Hierarchical structuring Extreme Programming devised the whole project into smaller iterations along with the simplest structure. When the company will be reorganized, this seems to be reconstructing the whole developing structure along with the software products, databases and their network infrastructure. The main aim within the organization is to optimize the product quality related to short iterations. By enhancing the Extreme Programming, it enable us to maximum cooperate with customers, reduce the risk of failure and enhance the quality and capability of softwares. Companies that already work on large scale can't be able to develop software more effectively without usage of agile development methodologies. Practical software development related to extreme programming having a spectrum of agility. At the extreme level of spectrum, we are completely able to predict the future requirements and goals without disturbing the software development model stages due to dynamically experiencing the already rapid changed requirements. The agility of a process could be determined by accommodating the changing requirements with the help of customers and developers. Practical approach lies among the completely agile and completely predictive.

References

Abdullah, E.T., and Abdelsatir. 2011. Extreme programming applied in a large-scale distributed system. *Computing Electrical and Electronics Engineering International Conference*. 442 – 446.

Alshehri, S., and L. Benedicenti, 2013. Prioritizing CRC cards as a simple design tool in extreme programming, *Electrical and Computer Engineering*.

Asif Q.G., 2015. Distributed Agile Development: Applying a Coverage Analysis Approach to the Evaluation of a Communication Technology Assessment Tool.

Dixon, M., and K. Dudman., 2004. Utilization Diagrams: An agile modelling technique for Information Systems, *international association for development of the information society.*

Vol. 2 No. 1, 2016
ISSN 2412-303X

International Interdisciplinary Journal of Scholarly Research (IIJSR)

Henderson and Sellers. 2003. Method engineering for OO systems development. *Communications of the ACM, 46*(10): 73-78.

Laubacher, R., and T. W. Malone., 2002. Temporary assignments and a permanent home: A case study in the transition to project-based organizational practices. *MIT Center for Coordination Science*.
Musa, S. and N. M. Norwawi., 2011. Secure E-commerce web development framework. *Information Technology Journal*, *10*(4): 769-778.

P.MESO and R. Jain. 2006. Agile software development: Adaptive system principles and best practices, *International journal of Information system management.*

Qumer A. 2008. A framework to support the evaluation, adoption and improvement of agile methods in practice, *The Journal of Systems and Software.*

Richard V., 2005. A complex adaptive system perspective of agile software development process, *International Journal of agile software development and technology,* 9(8):13-17

Rycroft, Robert W and E. Don., 2004. Self-organizing innovation networks: implications for globalization, *Journal of Informatics,* 24(3): 187-197.

Shahriar M. and S. Sohrabi. 2009. Challenges of user Involvement in Extreme Programming projects, *International Journal of Software Engineering and Its Applications,* 3(1):19-32

T.S.Ramya Krishna, 2011. Survey on Extreme Programming in Software Engineering, *International Journal of Computer Trends and Technology.*

Vidgen. 2006. Organizing for agility: A complex adaptive systems perspective on agile software development process, *Information Systems Research, 20(3): 355- 376.*

Syed-Abdullah., S. M., Holcombe, and M. George, 2007. The impact of an agile methodology on the well-being of development teams. *Empirical Software Engineering*, 11(1), 143-167.

Shahriar M. and S. Sahar. 2009. Challenges of user Involvement in Extreme Programming projects, *International Journal of Software Engineering and Its Applications,* 3(1):19-32.

Salo, O., and P. Abrahamsson. 2007. An iterative improvement process for agile software development, *Software Process: Improvement and Practice*, 12(1): 81-100.

Richard P. 2005. Triggers and Practice: How Extremes in Writing Relate to Creativity and Learning, *Journal of Systems and Software,* 32(2): 29-37.

Neil E. and L. Robert. 2015. Enabling Incremental Iterative Development at Scale: Quality Attribute Refinement and Allocation in Practice, *Software Engineering Institute*.

Maruping, L.M., and V. Venkatesh. 2009. A control theory perspective on agile methodology use and changing user requirements, *Information Systems Research*, *20*(3): 377-399.

Lawrence, R., and B. Yslas. 2006. Three-Way Cultural Change Introducing Agile within Two Non-Agile Companies and a Non-Agile Methodology, *Cutter IT Journal*