

# AGILE: Boon for today's Software Industry-A Review

Ashish Agrawal\*, Malay Tripathi\*, Sadhana Singh\*, L.S.Maurya\*\*

\* Master Of Technology, SRMS CET Bareilly

\*\* CS/IT Deptt., SRMS CET Bareilly

**Abstract**— Agile software development or agility is a sound concept of today's scenario which comprises the methods like scrum, Kanban, pair programming, extreme programming etc...Agile methods work with collaboration, competence, predictive planning and achievement and so that they let a project to be completed in its optimum state i.e. within time and cost and with producing best results. Agile methods are not bound to apply only in field of software development but they can be used in each and every field for getting best results in least efforts. This paper is an attempt to provide a study on agile methods as well as the chances to use them in other fields of engineering.

**Keywords**- Agile, Agile manifesto and values, quality, Agile Methods.

## I. INTRODUCTION

From the day, the word software developed; many software development strategies or methods have been evolved for developing mature and cost effective software. The basic was the Waterfall model but due to the limitation of sequential pattern, it became unsuitable for the changing scenario of software industry. After that many models like iterative waterfall model, prototype model, spiral model etc. have been developed but every model had its own limitations.

In recent decades, there is a number of techniques evolved that are like 'best meal' to the starvation of quality. Some of these techniques are Scrum, extreme programming, Kanban etc. commonly referred to as "Agile". These agile techniques are serving as a boon for software developers and so software communities are supporting agile and awaking the organizations for using these techniques. For example, in 2002, Mark Paulk, promoter of CMM stated (Paulk 2002)

"Many of the practices in the agile methodologies are good practices that should be thoughtfully considered for any environment. While the merits of any of these practices can be debated in comparison with other ways of dealing with the same issues, none of them should be arbitrarily rejected. Perhaps the biggest challenge in dealing effectively with both agile and plan driven methodologies is dealing with the extremists in both camps who refuse to keep an open mind."

In this paper, authors first look at agile manifesto values and principles and then they discuss about the agile techniques as well as their uses.

## II. LITERATURE REVIEW

Agile methodologies and principles place emphasis on incremental software development with short iterations, adaptation to changing requirements, close communication, self-organizing teams, and simplicity [8][5]. Agile methods are a subset of iterative and evolutionary methods and are based on iterative enhancement and opportunistic development processes. In all iterative products, each iteration is a self-contained, mini-project with activities that

Span requirements analysis, design, implementation, and test [7]. A key difference between agile methods and past iterative methods is the length of each iteration. In the past, iterations might have been three or six months long. With agile methods, iteration lengths vary between one to four weeks [5]. According to Agile Adoption Rate Survey [11] performed by Dr. Dobbs Journal in 2008 agile teams report significant improvements in productivity, quality, and stakeholder satisfaction, and reasonable improvements in cost. A similar survey conducted by Version One [8] additionally reports enhanced ability to manage changing priorities and significantly improved project visibility. For this reason, agile methods are especially suitable for development of information systems with changing and emergent user requirements [6][3].

On the surface, the advantages of agile appear to satisfy the major drawbacks of the traditional approaches. However, although the software experts that formulated agile had the best intentions, a key component in the development process was superficially considered – the user. The primary stakeholder that was given attention in agile was the customer, which in many cases is different than the users. Hence, by only understanding the customer's perspective, the software is still likely to not satisfy those who are purchasing and using the software. Only the users can provide the user's perspective, and in some cases, the customer is merely guessing what the user wants or what they want without regard for their users[1].

## III AGILE MANIFESTO VALUES AND PRINCIPLES

In February 2011, 17 software developers discussed and published "agile manifesto" that defines basic values and principles for Agile software development process. Some of these authors established a non-profitable organization named as, 'Agile Alliance' that promotes Agile software development. The Agile Manifesto values are as follows:

- V1: Individuals and interactions are more important than processes and tools.

- V2: Working software is more valued than comprehensive documentation.
- V3: Customer collaboration are emphasized over contract negotiation.
- V4: Responding to change is emphasized over following a plan.

The Agile Manifesto principles are as follows:

- P1: *Customer is everything increase his satisfaction by rapid delivery of software;* understand customer’s needs and provide him his software as soon as possible.
- P2: *Welcome and inclusion of late changing requirements;* try to incorporate all requirements even if they came in late phases of development.
- P3: *Continuous delivery of working software (in small time periods);* deliver software modules in weeks or months so customer can see development process is going on as well as can check the modules.
- P4: *Enhancement of technical excellence and good design by keeping continuous attention;* keep an eagle eye on whole process and make it more effective.
- P5: *Simplicity is essential;* software development process should be simple upto an acceptable level.
- P6: *Progress measurement through working software;* any system that can perform some functions is always most prominent measurement parameter.
- P7: *Face to face communication is the best communication;* because it provides sound and unmanipulated ideas.
- P8: *Develop projects in healthy environment with trustworthy motivated employees;* these employees know their duties very well always show cooperative nature.
- P9: *Have self organizing teams;* self manageable and self decision makers for small changes.
- P10: *Self judgment at regular intervals to become more effective;* managers should judge themselves as well as their development process for adapting advancements.
- P11: *Sustainable development, ability to maintain a constant pace;* maintain a constant growth during development process. Try to escape from situations like- ‘do half of project in three months and then forget it for a year’.
- P12: *Co-operation between developers and business persons;* all developers and managers should appreciate other’s ideas and advice them for betterment. They should understand that they are the pillars of ongoing

project and only with their combined efforts they can lead towards success.

These are basic principles that every organization should follow while adapting agile methods because somewhere all agile methods are based on these principles.

#### IV. AGILE KEY CONCEPTS AND METHODS

Figure 1 show how agile methods like scrum, pair programming, crystal clear and Kanban are related to each other. Crystal clear deals with communication between teams, Kanban controls the production process and scrum helps to incorporate changing requirements. Finally they produce the final product and commonly called Agile. These methods are further described as follows-

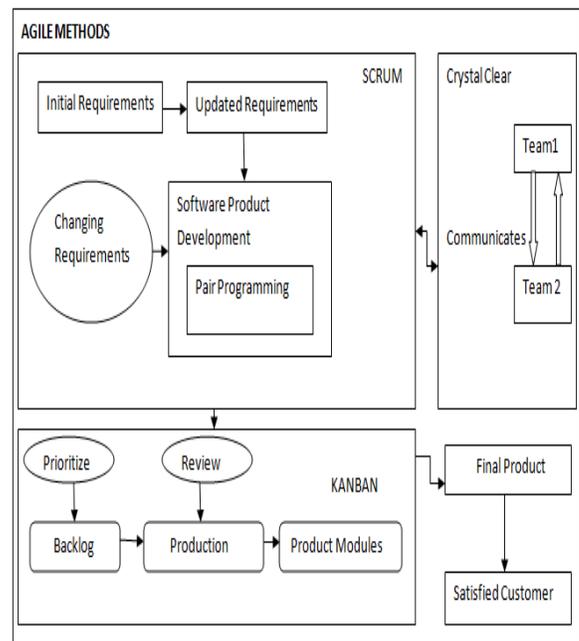


Figure 1: Communication between Agile Methods

A. *Scrum-* Scrum framework (developed by Hirotaka Takeuchi and Ikujiro Nonaka) is an empirical approach which works in iterative and incremental manner and let the changing mind of customer to be accepted. Scrum supports co-location of team members and so creates self organizing teams.

- *Sprint-* Sprint, basic development unit of scrum, is time bounded effort that is restricted to some days, weeks, or months. The wished-for functionality is written as *User Stories* and prioritized in a *Product Backlog* by a *Product Owner* representing the customer view. The highest prioritized functionality becomes the target functionality for a Sprint. The Sprint starts with a *Sprint Planning Meeting* where the targeted functionality is broken down and estimated (sized)[2].
- *Daily stand up meetings-* These meetings are minutes based meeting organized every-day. During this time

everyday's tasks are discussed and planned. These meetings are held at same place and at same time. It provides at least a partial solution to one of the biggest challenges The daily scrum meeting normally takes minutes, during which each team member answers the three scrum questions (Schwaber and Beedle 2001)[4][12]:

- What did you do since the last scrum meeting?
- Do you have any obstacles?
- What will you do before the next meeting?

B. *Extreme Programming (XP)* - XP, developed by Kent Back, focuses on reviews, continuous integration and testing, pairing and simplification. XP addresses risk and value of software at all levels of the development process. According to Beck customers (or managers) can pick three out of four control variables (these are cost, time, quality and scope) and the development team decides on the fourth (Beck, 1999). Technical people are responsible for work estimates, technical consequences of business decisions, development process and detailed scheduling within a release. Team size should be in maximum about 12 designers and the software not excessively complex. (Beck, 1999)[13].

C. *Pair Programming*- Pair programming is one approach which works on the principle *two minds working on same module are better than one*. As we know pair means a group of two persons; in pair programming "driver" and "navigator" makes pair. Where driver do source coding and navigator keep close watch on code being written.

Erik Arisholm, Member, IEEE, Hans Gallis, Tore Dyba, Member, IEEE Computer Society, and Dag I.K. Sjoberg, Member, IEEE defines in their conceptual model, the effects (in our case given by duration, effort, and correctness of the maintained program) of PP (versus individual programming) will depend on the moderating variables, system complexity and programmer expertise[5].

D. *Kanban*- Kanban is a process that controls the development and production process. This process ensures that every activity is on time and at the end of each activity we'll get a working module. Kanban prioritizes the task with the help of production backlog and help in completing them on time . it measures and manages the process flow and find out the opportunities for improvement.

E. *Crystal Clear*- This family of methodologies is developed by Alistair Cockburn and crystal clear focuses on people and communication between them. The communication between small teams should be transparent and good so that they all activities can be performed optimally. If the communication is bad, no one would be able to do better work and to create a healthy environment.

## V.CONCLUSIONS AND FUTURE WORK

Agile methodologies are the new age of software development, it's methods are serving their best, commonly they are focusing on people relations, customer satisfaction, producing best

product, cost benefit analysis and much more. Though it's more than a decade, agile manifesto published but still these methodologies or agile are the buzzing words of software industry.

Every process have a corner for improvement so for future work we recommend is to enhance these methods so that they can also be used by small organizations not only software development organizations but also other ones.

## REFERENCES

- [1] Jeremy T. Barksdale and D. Scott McCrickard, Software product innovation in agile usability teams: An analytical framework of social capital, network Governance, and usability knowledge management, Int. J. Agile and Extreme Software Development, Vol. 1, No. 1, 2012
- [2] Lene Pries-Heje, Jan Pries-Heje, Why Scrum works-A case study from an agile distributed project in Denmark and India, IEEE computer society, 978-0-7695-4370-3/11, 2011.
- [3] V .Manhic, A Case Study on Agile Estimating and Planning using Scrum, ISSN 1392 – 1215, 2011.
- [4] Maria Paasivaara,, Sandra Durasiewicz and Casper Lassenius,Using scrum in a globally distributed system:a case study, Wiley Interscience, 13: 527–544,2008
- [5] Erik Arisholm, member, IEEE, Hans Gallis, Tore Dyba , Member, IEEE Computer Society, and Dag I.K. Sjoberg, Member, IEEE , IEEE Transactions On Software Engineering, Vol. 33, No. 2, February 2007.
- [6] G. Canfora, A. Cimitile, F. Garcia, M. Piattini, and C. A. Visaggio, "Evaluating Performances Of Pair Designing In Industry," Journal Of Systems And Software, Vol. 80, No. 8, Pp. 1317–1327, 2007.
- [7] Laurie Williams, A Survey of Agile Development Methodologies, 2007.
- [8] Outi Salo, Kari Kolehmainen, Pekka Kyllönen, Jani Löthman, Sanna Salmijärvi, and Pekka Abrahamsson, Self-Adaptability of Agile Software Processes: A Case Study on Post-Iteration Workshop, Springer Verlag 2004.
- [9] A. Cockburn, Agile Software Development. Boston: Addison-Wesley, 2002
- [10] M. M. M'Uller, "Two Controlled Experiments Concerning the Comparison of Pair Programming".
- [11] Ambler S. W., Has Agile Peaked? Let's look at the numbers.
- [12] Schwaber K, Beedle M. 2001. Agile Software Development with Scrum. Prentice Hall: New York.
- [13] Beck, K. *Extreme Programming Explained: Embrace Change*. Reading, MA:, Addison-Wesley1999.

AUTHORS

**First Author**– Ashish Agrawal, M.Tech. Scholar, Shri Ram Murti Smarak College of Engineering & Technology, Bareilly, agarwal.ashish01@gmail.com

**Second Author**- Malay Tripathi, M.Tech. Scholar Shri Ram Murti Smarak College of Engineering & Technology Bareilly,malaytripathi406@gmail.com

**Third Author** –Sadhana Singh, M.Tech. Scholar, Shri Ram Murti Smarak College of Engineering & Technology Bareilly,ssadhana2008@gmail.com

**Fourth Author**- L.S.Maurya, HOD(CS/IT), Shri Ram Murti Smarak College of Engineering & Technology Bareilly , lsmaurya@gmail.com