Advances in Computer Science and Information Technology (ACSIT)
Print ISSN: 2393-9907; Online ISSN: 2393-9915; Volume 2, Number 4; April-June, 2015 pp. 370-374

http://www.krishisanskriti.org/acsit.html

Review of Agile Software Development Methodologies

Manoj Wadhwa[1] and Nidhi Sharma[2]

[1],2Department of Computer & Science Echelon Institute of Technology Faridabad, India
E-mail: [1]manojkw@gmail.com, [2]nim.nidhi23@gmail.com

Abstract—*Showing all the software engineering principles from a historical perspective, we can see how software processing methodologies evolved since past 50 years, but probably the most discernible exchange to software business in recent years has been the introduction of evince "Agile". As numerous areas have overblown, there is a requirement to realize the components and narration, as easily as how the agile methodologies are diverse from the traditional one. Methods like SCRUM, Extreme programming (XP), Feature driven development, Crystal Methodology etc. are increasingly being used to develop software using an adaptation approach rather than a predictive one. But there is a scarcity of the resources which describe on how these resources can be integrated with the agile methodologies. This paper basically gives the review of all methodology*

Keywords: *Agile Software Development, Extreme Programming (XP), SCRUM, Feature driven development, Crystal.*

1. INTRODUCTION

A software process is defined as a set of methods, practices, activities and transformation that are used to acquire and affirm software and its related product. The issues of how software development should be organized in order to deliver faster ,better, and cheaper solutions has been discussed in software engineering circles for decades .Many remedies have been suggested, from the standardization and assessment of the software process to the software process to a multitude of concrete tools ,techniques and practices. Recently, many of the suggestions for improvement have come from experienced practitioners, who have labeled their methods as agile software development. This movement had a huge impact on how software is developed worldwide. However, though there are many agile methods, but there is limited knowledge about how these methods .are carried out in practice and what their effects are. Agile models do more changes in software process

This paper will show how agile movement is different from the traditional approaches.

2. BACKGROUND

Agility, for a software development organization, is the power of software to choose and react expeditiously and fittingly to various changes in its surround and to the demands by this surround. Agile process is more flexible for all the development phases. It is not depend upon the size but it depends upon the speed. Numbers of groups have independently developed methods and practices to act to the changes they were experiencing in software processing and development.

In this paper there are eight sections. In First section, there is an introduction about the software agile methods and traditional methods. Second section, shows the background of the software development organizations and further describe the agile manifesto and their twelve principles, also describe the definition of "Agile" and traditional & agile approaches. Sections third, show the agile software process and also describe Extreme programming, SCRUM, Feature driven development, Crystal methodology. After that section fourth, describes what happen when to use agile model. Section five, gives the advantages of agile model. Section sixth describes the conclusion & future work of software development methodologies. Further section seventh, includes the acknowledgement and finally section eight, shows the references of this paper.

2.1 Agile Manifesto

The Agile Manifesto combined with Extreme Programming (XP), Dynamics System Development

Methods(DSDM), Adaptive Software Development, Scrum, Crystal Methods, Feature Driven Development(FDD),Lean Software Development and others who saw the need for alternative to documentation driven ,heavyweight Traditional software development processes.
This process gives the value doing these approaches.

- Responding to change over following a plan.
- Individuals and interactions over processes and tool.

Above two values gives the twelve principles:-

1. Deliver working software frequently, from a couple of weeks to a couple of weeks to a couple of months, with a preference to the shorter timescale.
2. Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
3. Welcome changing requirements, even late development .Agile processes tackle change for the customer's competitive advantages.

4. Business people and developers must work together daily throughout the project.
5. Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
6. The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
7. Working software is the primary measure of progress.
8. Continuous attention to technical excellence and good design enhances agility.
9. Agile processes promote sustainable development .The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
10. Simplicity-the art of maximizing the amount of work not done –is essential.
11. The best architectures, requirements, and Design emerge from self-organizing teams.
12. At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

2.1.1 Define "Agile"?

The understanding of the word agile varies in practice .In addition, it is difficult to define agile methods, as it is an umbrella for well-defined methods, which also vary in practice .This section will show how this word was explained in literature by its defenders, as well as by other researchers.

- Mental quick – Able to think fast and clearly way.
- Physically quick – Able to move body quickly and flexible.

Barry Boehm described agile methods as-
"An outgrowth of rapid prototyping and rapid development experience as well as the resurgence of a philosophy that programming is a craft rather than an industrial process"

Table 1: How Agile is different from traditional approaches

Traditional Approaches	Agile Approaches
Developers do waiting until the architecture is ready.	The whole team is working at the same time on the same iteration .Good coordination between team members.
Too slow to provide fixes to user.	Provide quick responds to user feedback.
No communication within the team, novices stay in their rooms and try to understand things.	High level of communication and interaction, reading groups, meetings.
Documents and review meetings are needed to solve an issue.	5 minutes discussion may solve the problem.
Everything is up front, everything is big before you stat.	The focus is on whether customer requirement are met in the current iteration.
Restricted access to architecture.	The whole team influences and understands the architecture. Everybody will be to do a design presentation.
In this type the environment is taken as stable and predictable.	In this type the environment is taken as turbulent and difficult to predict
Sequential and synchronous process.	Concurrent and asynchronous process.
Optimization is the goal.	Adaption, flexibility, responsiveness is the goal.
It is not flexible.	It is more flexible

3. AGILE SOFTWARE PROCESS

In" Agile" large number of approaches are comes. They solved the all problems that occur in the customer requirements. These approaches are Extreme Programming, Scrum, Lean Software Development, and Test Driven Development, Feature Driven Development.

3.1 Extreme Programming (XP)

Extreme Programming is the first and good agile approach. It provides the successful software practice .Whole team selected best form of the result. The team may consist of developers, who develop the software, testers who are responsible for providing Quality Assurance, and analysis who help in design and the customer representative who provides the feedback. The customer representative may be the actual end user of the system.

Planning is important part of the XP; it shows how much effort and cost is developing project. These planning are very effective because the product is visible all time .There are two types of planning –

1. Release Planning
2. Iteration Planning

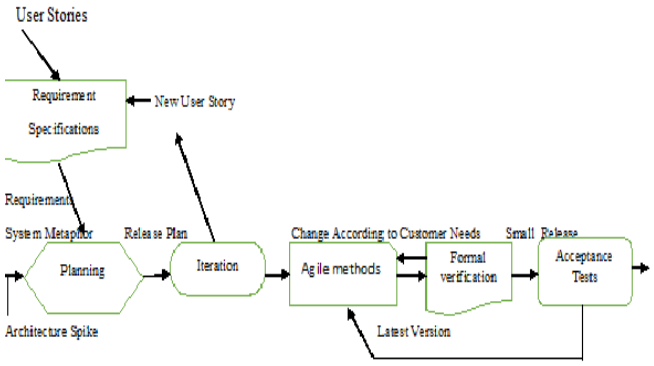

Fig. 1: The flow in a project using Extreme Programming

XP is accepted all the phases that's lies in software development cases. Designing is most important phase in software development, it will shows the functionality of the system. To achieve simple design, XP puts an emphasis on using refactoring techniques such as removing duplicated code, improving the existing design. Programmers must verify

that the system is still operational after refactoring activity takes place. Phases The XP process requires that all the phases of software development viz. design , implementation and testing of the system should be carried out by a pair of programming sharing one computer .This helps programmers to spend more time on finding solutions to challenging problem and less time doing routine debugging.

Pair programming means code is written by two programmers on a single machine. This code is checked by other programmer .This process gives the better design, testing and code.

Small Release is Define the features that have to be in project, represented as stories, which need to be developed. Every story represents the smallest increment to which new features of the system can be added, which usually takes only a few weeks to be developed.

3.2 SCRUM

SCRUM has been used with the objective of simplifying project control through simple processes, easy to update documentation and higher team iteration over exhaustive documentation.

SCRUM shares the basic concept and practices with the other agile methodologies, but it comprises project management as part of its practices. These practices guide the development team to find out the tasks at each development iteration.

In addition to the practices defined for agility, one main mechanism recommended by SCRUM is to build a backlog .A backlog is a place where one can see all requirements pending for the project, sized based on complexity, days or some other unit of measure the team decides. Inside a product backlog, there is a simple sentence for each requirement, something that will be used by the team to start discussions and putting details of what is needed to be implementation by the team for that requirement.

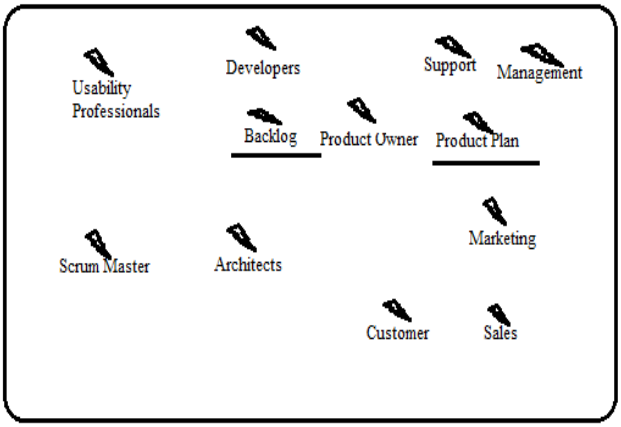

Fig. 2: Key role and interaction artifacts in SCRUM

For the team of SCRUM, three main roles are defined as shows in fig 2.The first role is the product owner, who mainly would be the voice of business. The second ole is the SCRUM team which comprises developers, testers, and other roles. This team would make initial contact with customer and identify the need for a new product. SCRUM master, the third role, is responsibilities for keeping the team focused on the specific goals, and help the team members to solve problems when they appear.

The process of development using SCRUM divides the project into phases .In each phases, one feature is fully developed, tester, and become ready to go to production. The team does not move to a new phase until the current phase is completed. Whether what is being done adds value to the process or not, is the main concern of each phase.

Current studies on traditional SCRUM development have shown that despites its advantages, it is not best suited for products where the focus is on usability. It fails to address usability needs of the user, because product owner keep their focus mainly on business issues and forget about usability. Since product owners usually come from business background, they lack the experience, skills, and motivation to design for user experiences. Moreover, traditional agile methodologies are not concerned about the user experiences vision, which drives the architecture and is essential for ensuring a coherent set of user experiences. According to Nidhi Sharma, U-SCRUM is an agile methodology for promoting usability.

Briefly, SCRUM is considered an iterative, incremental methodology of software development. It was proposed for software development projects, and at the same time, it can use as a program management approach.

3.3 Feature Driven Development

The Feature Driven Development approach focuses on the software feature of the system. They are the main driver of the entire development process. It differs significantly from the other agile processes because they put a strong emphasis on planning and upfront design. In FDD process is to build a detailed model of the system, which captures all the stakeholders' assumptions and requirement. Once the domain model is build, the team members print a list of the system.

Each feature has to be development in a few hours or days, but no longer than two weeks. Using FDD, development teams are divided according to design and to implement a particular feature. The development work is performed in parallel on all the features. Each team is headed by a feature owner, who is responsible for the code segment that implements those features. This is contrasted with the XP approach to the whole development where the ownership of the code belongs to the whole development team and not to a specific member.

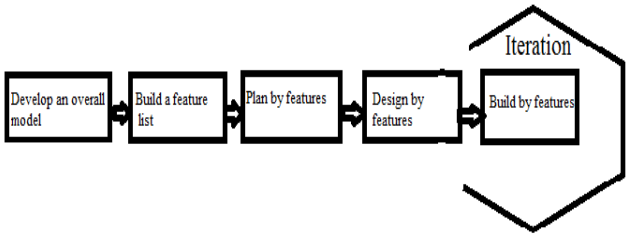

Fig. 3: The simple steps of Feature Driven Development

The FDD process enforces rigorous guidelines in order to find defects in the system. It also enforces coding standards and encourages regular build on a daily or week's basis in order to add freshly designed features to the base system. Since the features are developed in parallel, it is mandatory to have a configuration management system that allows calmly integrating of the changes that are made to the system. In FDD approach there is a tracking and papering mechanism that is used to show the project status based on the number of features that have been implemented as well as the overall progress of the design, coding, and testing activities. Each feature is scored using a value ranging between 0 to 1 and anything in between refers to a feature in progress.

3.4 Crystal Methodology

The Crystal methodology is one of the most lightweight adaptable approaches to software development.

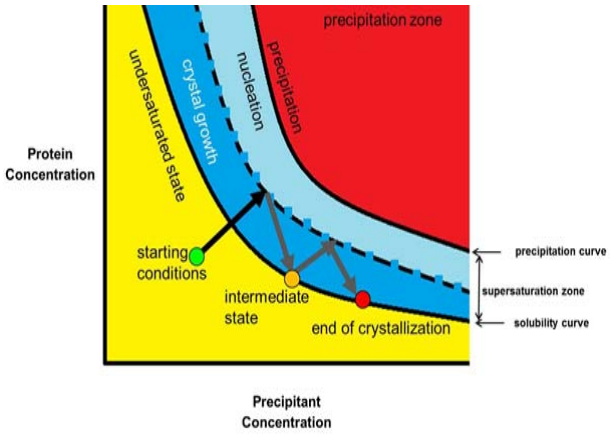

Fig. 4: Show the Crystal growth

- Crystal is actually comprised of a family of agile methodologies such as Crystal clear, Crystal Yellow, Crystal Orange and others, whose unique characteristics are driven by several factors such as team size, system criticality and project priorities.
- It addresses the realization that each project may require a slightly tailored set of policies, practices, and processes in order to meet the project's unique characteristics.
- Several of the key tents of Crystal include teamwork communication and simplicity as well as reflection to frequently adjust and improve the process.
- Like other agile methodologies Crystal promotes early, frequent delivery of working software, high user involvement, adaptability and the removal of bureaucracy or distractions.

4. WHEN TO USE AGILE MODEL

- When new changes are needed to be implemented. The freedom agile gives to change is very important. New change can be implemented at very little cost because of the frequency of new increments that are produced.
- To implement a new feature the developers need to lose only the work of a few days, or even only hours, to roll back and implement it.
- Unlike the waterfall model in agile very limited planning is required to get started with the project. Agile assumes that the end user's needs are ever changing in a dynamic business and IT world. Changes can be discussed and features can be newly affected or removed based on feedback. This effectively gives the customer the finished system they want or need.
- Both system developers and stakeholders alike, find they also get more freedom of time and options than if the software developed in a more rigid sequential way. Having options gives them the ability to leave important decisions until more or better data or even entire hosting programs are available; meaning the project can continue to move forward without fear of reaching a sudden standstill.

5. ADVANTAGES OF AGILE MODEL

- Customer satisfaction by rapid, continuous delivery of useful software.
- People and interactions are emphasized rather process and tools .Customers, developers and testers constantly interact with each other.
- Working software is delivered frequently (weeks rather than months).
- Face to Face conversation is the best form of communication.
- Close daily cooperation between business people and developers.

6. CONCLUSION & FUTURE WORK

Agile Methodologies came into existence after the need for a light way to do software development in order to accommodate changing requirement environment. Agile methodologies provide some practices that facilitate communication between the developer and customer, and under go develop-deliver-feedback cycles, to have more specific view of the requirement, and be ready for any change

at any time. The main aim of agile methodologies is to deliver what is needed when it is needed.

Agile methodologies include a set of software development approaches. They have some variations, but still they share the same basic concepts. The main agile methodologies that being used include XP, SCRUM, Crystal methodology, Feature driven development, .XP is the coding of what the customer specifies and the testing of that code, SCRUM support management role in software development.

Agile methodologies are not best suited for all projects. When communication between the developer and the customer is difficult, or when the development team includes mainly beginners, agile methodologies will not give the best result. These methodologies exhibit optimum result when there is a strong communication between the developer and customer, and the development team compromises skilled team members. When there is a big chance for misunderstanding the exact customer's requirements, or when the deadlines and budgets are tight, then agile methodologies are among the best software development approaches to apply.

As a future work, there is a need to review other agile processes not covered in this paper such as Dynamic Systems Development Method (DSDM), Adaptive Software Development (ASD).

7. ACKNOWLEDGEMENTS

We would like to convey my thanks message to Dr. (Professor) Manoj Wadhwa. Their advices and support make us able to complete our work and also thankful to Echelon Institute of Technology for providing us all the facilities and services during our studies.

REFERENCES

[1] Agile Documentation: Pattern Guide to Producing Lightweight Documents for software project.

[2] Agile Processes in software Engineering and Extreme Programming .9th International Conference.XP 2008.,Limerick,Ireland,june 10-14,2008:Proceedings.

[3] Aoyama, M,"Agile software process model", IEEE software 2002.

[4] Beck, Kettle al. (2001)."Manifesto for Agile Software Development". Agile Alliance. Retrieved 14 June 2010.

[5] Beck, Kettle al. (2001)."Principles behind the Agile Manifesto", Agile Alliance. Archived from the original on 14 June 2010.Retrieved 6 June 2010.

[6] Cristal, Dwelt and R.Priklandnicki, Usage of SCURM Practices within a Global Company. Global Software Engineering, 2008.ICGSE 2008.IEEE International Conference on, 2008, 222-226.

[7] D.B.Cao,"An Empirical Investigation of Critical Success Factors in Agile Software Development Project". PhD thesis, Capella University, USA, 2006.

[8] G.Cugola and C.Ghezzi,"Software processes: a retrospective and path to the future". In pros of software process improvement and practice conference, 1998, pp.101-123.

[9] High smith, "Agile software development the business of innovation"IEEE software 2001

[10] J.Erickson ,K.Lyytinen and K.Sian,Agile Modeling, Agile Software Development, and Extreme Programming .The State of Research .In Journal of Database Management ,16(4),2005,88-100.

[11] J.Ferreira, J.Noble, R.Biddle, Up-Front Interaction Design in Agile Development. In Agile Processes in software Engineering and Extreme Programming, Springer, Berin/Heidelberg, 2007, 9-16.

[12] Kieran comboy and Sharon Coyle, "People over processes: Key Challenges in agile development"IEEE software.

[13] Mikio Anoyama,"Agile software process model", IEEE software, pp 454-455.

[14] P.Schuh, Integrating Agile Development in the Real World, Charles River Media, 2004.

[15] Qasaimeh, M,"Comparing Agile software processes based on the software development project requirement", IEEE computer society, 2008.

[16] S.Nerur and V.Balijepally,"Theoretical Re-flection on Agile Development Methodologies,"Comm. ACM, vol50, no.3, 2007, pp.79-83.

[17] S.Nerur, R Mahapatra, and G.Mangalaraj,"Challenges of Migrating to Agile Methodologies," Comm.ACM, vol, 48, no 5, 2005, pp.72-78.

[18] T.Dybad and T.Dingsoyr,"Empirical Studies of Agile Software Development: A Systematic Review, "Information and software Technology, vol.50, no's 9-10, 2008, pp 2-4.

[19] What is Extreme Programming, Ron Jeffries, available http://www.xprograming.com/xpmag/whatisxp.btm,extracted on 11/02/2006