International Journal of Research in Advent Technology, Vol.7, No.5, May 2019
E-ISSN: 2321-9637
Available online at www.ijrat.org

A Path for Assessing Better Agile Methodology

Pooja Nagpal
Department of Computer Science and Engineering, Amity University
Email: poojabatra9@gmail.com

Abstract- Agile Methodology is a practice that promotes continuous iteration of development and testing throughout the development lifecycle of that specified software. Agile development has emerged as a promising prospect for software development lifecycle models with popularity increasing in recent times with more and more companies adopting this methodology. This is due to the fact that Agile provides practices of several varieties which can be adopted for their relevance to the current environment of the respective organization. The objective of this paper is to provide an insight into the various Agile methodologies such as Extreme Programming, Scrum, Feature driven Development, Dynamic Systems Development Method and Crystal. Furthermore these methods will be compared on the basis of the following factors; nature of the project, development team skills, customer involvement, iteration duration, as well as project constraints.

Index Terms- Agile, SDLC, Crystal, FDD, SCRUM, Software Development.

1. INTRODUCTION

Currently businesses are engulfed in a rapidly changing environment for most answer to a global clientele because they have to respond to new opportunities and markets, changing financial conditions, and emerging competition. This leads to rapid development being a critical requirement to fulfill the ever growing demands of the customers or clients[1]. Main functionality of all rapid software development processes is to produce useful software quickly, that's why incremental development process is chosen for software development which will support frequent and sudden changes. Agile methods are incremental development methods with the increments being typically small and where the new releases of the system are made available to customers every two or three weeks. Customer involvement is crucial for the development process because with rapid feedback it is easier to manage changing requirements. Documentation is minimized through the use of informal meetings rather than formal meetings with written documents. Following figure shows agile development process. It is kind of normal software development life cycle which includes more customer satisfaction, faster delivery, happy users and teams.

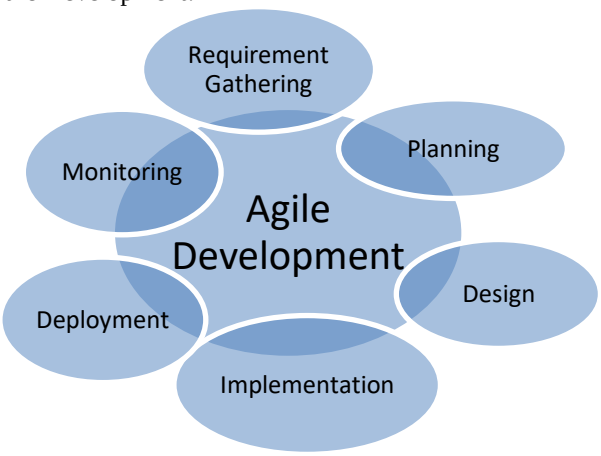

Fig 1. Agile Development Process

2. AGILE METHODOLOGIES

Various Agile Methods are working in software industry like Extreme Programming[2], Scrum[3], Feature driven Development[4], Dynamic Systems Development[5], Adaptive Software Development[6] and Crystal[7], Kanban[8] and many more. In my paper XP, Scrum ,FDD ,DSDM and Crystal technologies will be discussed in detail and comparison of these methodologies will help in taking decision to adapt which methodology in different situations.

2.1 Extreme Programming:

Extreme Programming (XP) is the most popular among Agile methods in which fresh versions may be built a number of times per day, increments are supplied to customer every two weeks, and complete set of decided tests should be successful for each build before being supplied to the customer. It involves the steady increase of functionality with each increment, smaller releases, and automated unit testing. This

technique for deciding tests for functionality prior to its implementation makes retesting after addition of new code easier. All developers collectively take responsibility for the complete code. The integration of the modules is performed continuously as they develop, and most importantly, an on-site full time customer representative is required for communicating the desired requirements to the development team to ensure the project stays on track. release. Features of XP are user stories, small releases, coding standard, continuous integration, on-site customer, simple design, testing, refactoring, pair- programming, 40-hour workweek and collective ownership. Following figure shows the process of extreme programming.

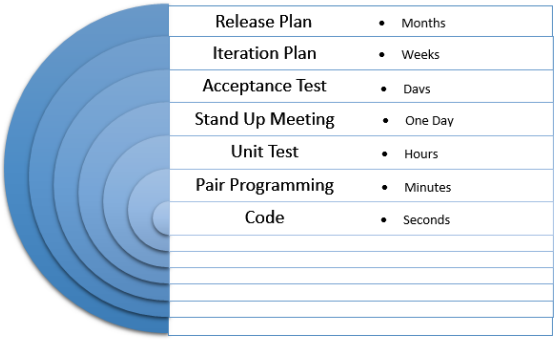

Fig2. Extreme Programming

2.2 SCRUM

Scrum approach emphasizes on the incremental development of a project. It operates in three stages with the first, being a general planning stage where the objectives are specified, and planning of the software architecture design is underway.. Second stage involves increment development through succession of sprints. Third stage consists of project closure with all the necessary documentation and evaluation of the project. Each sprint in accordance with its release is of a fixed duration usually lasting 2-4 weeks. It starts by deciding the functionalities that need to be performed with its appropriate staffing. The Scrum master arranges meetings within the development team and carries out the negotiation as a middle man between the Alteration to any particular class can only be done with the permission and presence of said class owner. There are five main tasks which are performed iteratively in this type of development method. The first task is preparing an overall model; this is where the domain under consideration is clearly identified. Second, a list of the features is prepared, more specifically groups of related subject matter and related areas are set. The third task deals with planning by feature, which means creating plans for development. The last two tasks are designing and building by feature. As the names suggest, it deals with extensive designing which is followed by coding testing after which the system is then packaged.

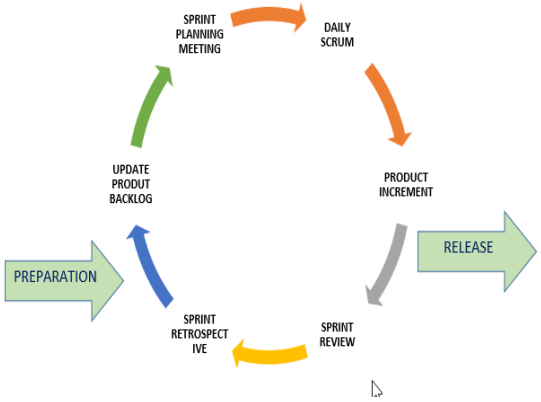

Fig 3 SCRUM

FEATURE DRIVEN DEVELOPMENT

As its name suggests its essential property is feature. It is client centric process and clients requirement is features. Following figure shows 5 process of FDD. All are feature oriented.FDD is architecture centric approach [9].

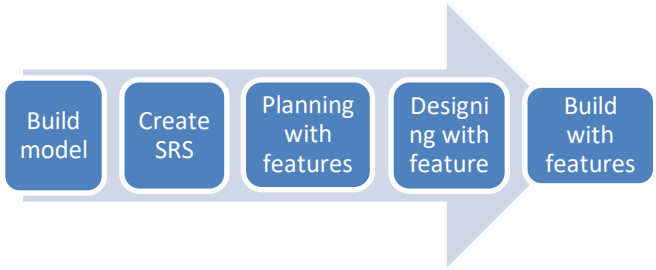

Figure 4 Features Driven Development

2.3 Crystal Methodology

Crystal is an agile method that is basically a family of approaches which vary according to the size of the project and its complexity. All specific approaches are named so after a color, which essentially represents the criticality and size of the project. Even though the Crystal implementations may be specific, the fundamental principles of each implementation include:

• Frequent Delivery: Intermediates of the complete versions are delivered every few months.

• Continuous Feedback: Regular project team meetings and meetings with stakeholders are carried out enhancing the communication shared within members of
the project as well as with project team and client.

• Access to Users: Project team maintains contact with one or more individuals who might be actual users of the system.

• Constant Communication: Frequent communication among team members is present to ensure the project stays on track and that there is no confusion.
• Automated testing and integration: Several verification, frequent integration measures and automated testing are carried out.
• Focus: Tasks that are of high priority are known to each of the team members with the goal being the timely completion of them with no interruptions.

2.4 DSDM (Dynamic System Development Method)
This approach concentrates more on project management related activities. It focus on mapping of requirements to deployment of project to provide better customer satisfaction and on time delivery. Mainly it has four phases iterative development, facilitated workshops, time boxing and prioritization. This method is very adaptable and customizable.

Characteristic	XP	Scrum	DSDM	FDD	Crystal
Development Approach	Iterative Increments	Iterative Increments	Iterative	Iterative	Incremental
Iteration Period	1 to 6 weeks	2 to 4 weeks	80 %solution in 20 % time	2 days to 2 weeks	Dependent on method
Size	Smaller Projects	All types of Projects	All types of Projects	More complex Projects	Dependent on method
Customer Involvement	Customer Involved	Customer through the role of Project owner	Customer through frequent releases	Customer through reports	Customer through incremental releases

3. WHY AGILE APPROACH IS SO MUCH POPULAR

Reduced risks, higher customer satisfaction, increased project control, high quality product, early and predictable delivery, focus on business values, predictable cost and schedules are some of the benefits of this approach. This list does not end here. It is better than traditional approaches in every manner. That's why Agile approach is so much popular in software industry.
The Agile Method emerged from the experience of leading software professional with their real- life projects that they worked on in the past. This led to the challenges and limitations of traditional development to be discarded. Subsequently, the Agile Method was accepted by the industry as the better solution to tackle project development.

This method offers a lighter framework which assists the teams in their respective fields. The main focus always being the proper delivery of the project in rapid succession. The Agile Method ensures that the value of the project be optimized through its development process with the use of iterative planning and proper feedback results. It adapts to changing requirements throughout the process with relative ease. A relatively important aspect A relatively important aspect
is that Agile focuses on empowering the teams to optimize their releases during development. This leads to building a proper product, for it doesnt try to market software before its written, rather than it deals with upcoming demands or requirements from the client. As a result the product is allowed to be as competitive as possible in the market. This feature is what draws many developers and stakeholders to this approach. Regular feedbacks are always necessary here to implement updates and changes for fast delivery.

Continuous alignment of the delivered software with the desired business needs leads to teams easily adapting to the changing requirements. This helps with managing the risk a project deals and that is because of the iteration based approach. Each iteration is handled with its own risk analysis and hence there is a much better chance of a good product because through each iteration, risks are analyzed and dealt with

4. COMPARISON

The following table consists of a comparative analysis of the different methodologies that were mentioned earlier on in this paper on the basis of four factors.

5. CONCLUSION

The popularity that Agile has gained in recent times is due to the flexibility and openness it provides with requirement changes. This makes it a very suitable approach for the current business needs of any small or medium sized project.

It's not recommended for large scale projects but that will change with time for this approach handles a lot of issues efficiently. Agile offers a variety of different methodologies, each of which contains its own perks and flaws. Software companies, based on their environment, project types, resources and other constraints can choose which method would suit their needs best. This paper is designed to shed insight into Agile methodology and its array of methods through comparative analysis of the mentioned methods.

REFERENCES

[1] I. Sommerville, Software Engineering, Addison-Wesley, 2010.
[2] K. Beck, "Embracing change with extreme programming," *Computer*, vol. 32, no. 10, pp. 70-77, 1999
[3] K. Schwaber and M. Beedle, Agile Software Development with Scrum, Pearson Education International,
[4] J. Stapleton, DSDM, Dynamic Systems Development Method: The method in practice, Addison-Wesley, 1997.
[5] S. R. Palmer and J. M. Felsing, A practical guide to Feature driven Development, Practice Hall,2002
[6] J. A. Highsmith, Adaptive Software Development: A Collaborative approach to managing complex systems, Dorset House, 2000.,
[7] A. Cockburn, Crystal Clear: A human-Powered methodology for small teams, Addison-Wesley
[8] D. J. Anderson, Kanban, Blue Hole Press, 2010.
[9] S. W. Ambler, "Feature Driven Development (FDD) and Agile modeling," Ambysoft Inc.,