Analytics of Agile Methodologies: An Empirical Review

Charul Nigam[1], Prof. (Dr.) Ajay Rana[2]

[1]Research Scholar, Dept. Of Computer Science & Engineering, Amity University, U.P, India
[2]Professor, Dept. Of Computer Science & Engineering, Amity University, U.P., India

---

Abstract – *Agile Methodologies have been widely accepted by majority of software companies. By following agile model it is easy to deliver a project on time and maintain its quality. Various agile methodologies followed are: extreme programming, scrum, lean software development, adaptive software development etc. All these methodologies can be followed depending upon the software requirements and budget decided for the project. This paper gives a review of various agile methodologies that have been adopted by the companies to deliver the software.*

***Key Words*: Agile Model, Adaptive Software Development, Crystal, Extreme Programming, Feature Driven Development, Lean Software Development, Scrum.**

1.INTRODUCTION

Agile models are based on continuous and iterative methodology in collaboration to deliver a cost effective and high quality software product which is developed in time and can adapt to changes very easily and quickly. Agile methodology focuses on delivering the smallest working module of the project as quickly as possible and then modifying it and adding new features to it as per the desire of the customer. Adopting agile methodology helps to reduce the overall risk on the project. It also allows the product or module to be modified quickly as compare to the other SDLC models. Work is done in repetitive cycles which take maximum time of five to six week. Most of the industries and organizations have adopted agile as it guarantees to give the result. Adopting any method of agile methodology stresses on documentation which can be understood easily by anyone. The important features of agile methodology are:-

- Early and frequent delivery of software product
- Iterative, continuous and collaborative development approach
- Reduced number of defects
- Incessant Testing
- Maximum returns

In recent years many researchers and scholars have proposed various agile methodologies. Some of these methodologies which we will discuss are as follows:-

- Extreme programming (XP)
- Scrum
- Lean Software Development (LSD)
- Adaptive Software Development (ASD)
- Feature Driven Development (FDD)
- Crystal

2. EXTREME PROGRAMMING (XP)

This methodology was given by Kent Beck, Ron Jeffries, and Ward Cunnigham in the year 1998. XP primarily focuses on building the software products whose requirements are not clear or products whose requirements are changing frequently. In this approach the software is modified quickly as per the requirements and feedback of the customer. In this approach we follow:-

- Collecting requirements from customers
- Making teams of programmers
- Creating easy and simple designs
- Continuous development of software product
- Continuous testing of the software product
- Frequently releasing the software product in cycles

This methodology appropriate for the projects where the customer is not sure of the requirements and cannot predict what he wants out of the software.

XP has six phases for developing a software product which are as follows:-

- Write story based on the current module.
- Do the planning to build the story such as estimate effort, resources etc.
- Perform the analysis, design, coding and testing of the current story.
- Perform rigorous testing in order to ensure defect free module.
- Release the software so that it can be used by the customer
- Finally the cycle ends when there is no requirement on the current story.

There are four key values of XP which are as follows:-

Communication: It is one of the most important factors for the success of any project. Because of lack of communication or poor communication any project can fail. So there should be proper communication among the team members of the same project in order to ensure the success of the project. It can be done by making the members of the team sit together; share the same workspace, regular meetings should be conducted to discuss the project status etc.

Simplicity: The final software should be made in a simple manner which can be easily understood by the customers and in which modifications can be made easily. While developing the product one thing should be kept in mind that the software developed should meet the customer's needs and requirements.

Feedback: It means that it is the duty of the developers and the project managers to obtain feedback from the customers timely so that changes can be done accordingly and the value should be given to the feedback of the customer. In every iteration the feedback should be taken in order to ensure the success of the project.

Courage: It is concerned with taking hard and corrective decisions which are in favour of the project.

XP has some limitations also. These limitations are as follows:-

- This methodology is not suitable for large projects.
- It needs to have good coordination among all the team members as a small mistake can lead to failure of the project.
- "Metaphors" are required to be developed carefully as all the terminologies related to the project should be known to all the team members.

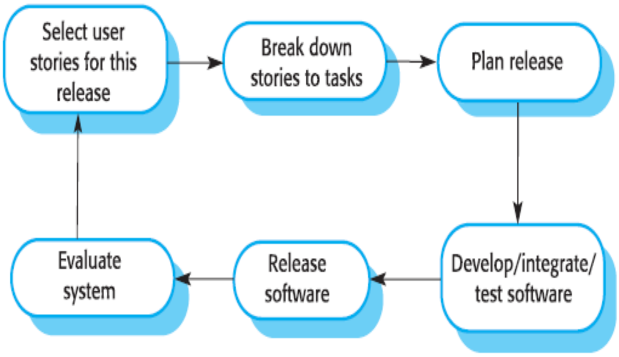

Select user stories for this release
Break down stories to tasks
Plan release
Develop/integrate/ test software
Release software
Evaluate system

Fig: 1 Extreme Programming Methodology

3. SCRUM

It was given by Jeff Sutherland, Ken Schwaber, and Mike Beedle in the year 1996. It emphasizes on agile project development rather than developing the project. Scrum focuses on enhancing the skills of the managers rather than enhancing the development of the project. In this approach the customer's requirements are completed quickly in about 2-4 weeks time in continuous iterative cycles. This continuous development helps in reducing the risk in the project and also helps to gain the client confidence. This approach is ideal for small scale projects.
Scrum has following artifacts:-

Product Backlog: It consists of all the requirements related to the project including-all the facts about products, its defects, amount of risk associated etc.

Sprint Backlog: It consists of all the tasks which are pending till date and needs to be completed in order to meet the client requirements and success of the project.

Burn Down Chart: This is a chart which shows the daily status of the tasks ie whether the tasks have been completed or they are still pending. It also helps in keeping track of the status of the project requirements. Once the requirements are completed they are removed from the chart.

Release Backlog: The requirements when completed form a module and when this module is delivered to the customer then the requirements are put in this release backlog. All the upcoming releases are also kept in this backlog.

The scrum methodology has following limitations:-

- Not suitable for large projects as it will require large team size which may lack coordination.
- Success of the project depends upon how well individual team members perform and complete their tasks.
- It does not take into account criticality of the project.
- Frequent customer communication and feedback is not possible as the clients are usually on-site.

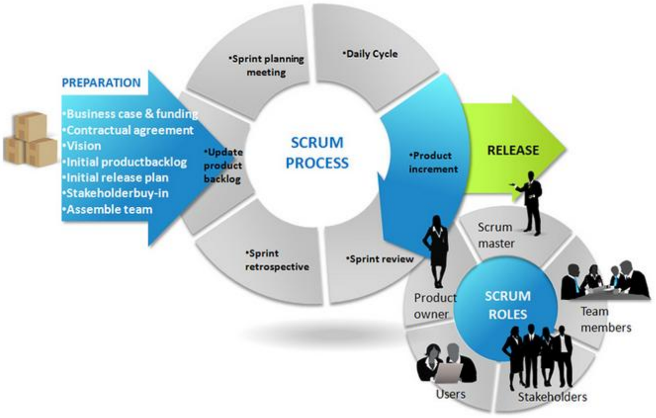


Fig: 2 Scrum Methodology

4. LEAN SOFTWARE DEVELOPMENT (LSD)

This methodology was adopted in the year 1980 from the production system of Toyota called as lean. It primarily stresses on the managerial front of the project development rather than focussing on the technical aspects of the project development. This approach works well with methodologies such as XP as they concentrate more on developing a project technically. Main idea behind this concept is to reduce the waste and optimizing the development process in order to have the best results.
LSD approach works on 7 principles which are as follows:-

Reduce Waste: This means we should delete those requirements which do not add to build the customer product.

Build Quality: We should focus on building the product which meets the customer requirements. If anything that does not meet the requirements it should be deleted from the project.

Knowledge Creation: The focus should be on building the software product as early as possible so that it is being given to customer for feedback. As per the feedback we can focus on the requirements which are more important from the customer's viewpoint.

Deferred Commitments: Decisions should be taken after doing a complete research on the process of development. A decision can only be taken if we have understood the problem correctly and then design solutions for it.

Faster Delivery: As soon as the requirements are gathered they should be converted into a module in order to ensure the quick delivery of the software product or module.

Respect Fellow Team Members: All the members of the team should respect each other in order to work as a team for the development of the software product.

Optimization: Suggestions should be taken from across teams to build a defect free product which meets the customers' needs.

The **limitations of the LSD** approach are as follows:-

- Team building plays a major role as if all the team members are working in collaboration then only results can be achieved.
- If any requirement or fact is missing then it could result in failure of the project as that missing requirement can be essential for the customer.

5. ADAPTIVE SOFTWARE DEVELOPMENT (ASD)

This agile methodology was given by Jim Highsmith and Sam Bayer in the year 2000. This methodology was considered to be a part of rapid application development and primarily stresses on creating the software systems rapidly. It provides solutions to the complex software projects through repetitive and continuous cycles of project development.

It consists of starting the project development, iterative planning of development cycles, and continuous development of the product features, reviews of product requirements and lastly the quality assurance and release of the product. Initially at the start of the project the customer do not have all the requirements or we can say that he is not clear with the requirements. He just only has a concept which gradually develops into a software project.

This approach is limited to managerial aspects of the projects and do not take into account the technicality of the project. The team size in this approach depends upon the type of project being constructed but larger the team size the level of agility goes on reducing.

6. FEATURE DRIVEN DEVELOPMENT (FDD)

This approach came into existence in the late 1999 as a collaborative work between Jeff DeLuca and Peter Coad and was later given by Palmer and Felsing in the year 2002. This approach primarily emphasizes on five domain activities:-

- Build a complete outline of the model
- Develop a set of features
- Plan according to those features
- Design according to those features
- Build the product with those features

FDD fixates on the design and building phases, accentuates quality aspects throughout the process and includes frequent and tangible distributions, along with precise monitoring of the progress of the project. It is simple to understand but potent approach to build the product or solutions. FDD is inhibited to diminutive to medium sized teams. The methodology deals with sceptical requisites and center on Object Oriented modelling.

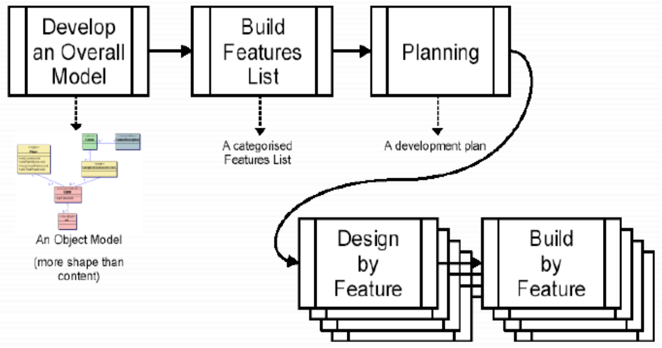
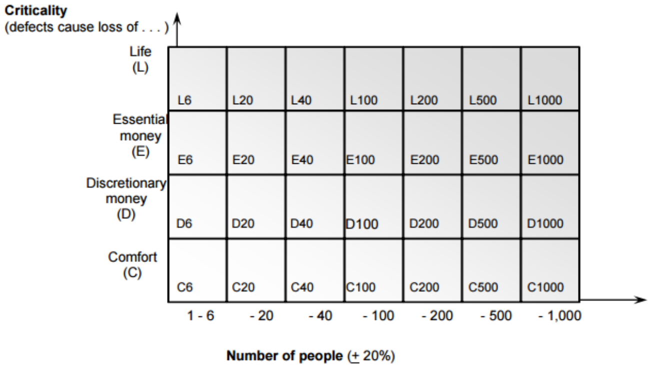


Fig: 3 FDD Methodology

7. CRYSTAL

Crystal method says that there is no single method which can define the development process for all the projects. There has to be different approach followed for developing a specific type of product. For this it has defined various crystal categories such as Crystal Clear, Crystal Yellow, Crystal Orange and finally Crystal Red. The most agile method is crystal clear and least agile is crystal red.



Criticality	1 - 6	- 20	- 40	- 100	- 200	- 500	- 1,000
Life (L)	L6	L20	L40	L100	L200	L500	L1000
Essential money (E)	E6	E20	E40	E100	E200	E500	E1000
Discretionary money (D)	D6	D20	D40	D100	D200	D500	D1000
Comfort (C)	C6	C20	C40	C100	C200	C500	C1000

Fig: 4 Crystal Methodology Graph

In the graph shown above the team size is depicted along x-axis. A team grows bigger in size we cannot manage face to face communication all the time and therefore the need arises of some documentation about the project. The y-axis tells the amount of damage which can be in the form of comfort loss, money loss etc. Every method defined in crystal has its own set of procedures and practices to be followed.

8. CONCLUSIONS

After studying the various agile methodologies I conclude that we can separate the agile methodologies in to two broad variants: Technical Aspect and Managerial Aspect.
Some of the methodologies focus on the technical aspect of the product development while others focus on the managerial aspect of the product development.
I would suggest depending upon how the project is; its requirements, its cost, its time schedule etc decision should be taken. In order to take a rational decision both the aspects of the product should be considered.

REFERENCES

[1] Agile Alliance. “Principles behind Agile Manifesto”. www.agilemanifesto.org/principles.html, 2006.

[2] Beck, Kent, "Manifesto for Agile Software Development". Agile Alliance, http://agilemanifesto.org, 2001.

[3] Beck, K. and C. Andres, “Extreme Programming Explained” (2nd Edition), Addison-Wesley Professional, 2004.

[4] Beck, Kent, “Extreme Programming Explained: Embrace Change”, Addison-Wesley, ISBN 0-201-61641-6, 2000.

[5] Beck, Kent, “Extreme Programming Explained: Embrace Change”, second ed., Addison-Wesley, ISBN 978-0321278654, 2004.

[6] D. Cohen, M. Lindvall, P. Costa, “An introduction to agile methods, in: M.V. Zelkowitz (Ed.)”, Advances in Computers, Advances in Software Engineering, vol. 62, Elsevier, Amsterdam, 2004.

[7] Schwaber, K. and M. Beedle, “Agile Software Development with Scrum”, Prentice Hall PTR, 2001. [8] K. Schwaber, M. Beedle, “Agile Software Development with Scrum”, Prentice Hall, Upper Saddle River, 2001.

[8] M. Poppendieck, T. Poppendieck, “Lean Software Development: An Agile Toolkit for Software Development Managers”, AddisonWesley, Boston, ISBN 0-321-15078-3, 2003.

[9] Highsmith, J., “Adaptive software development: a collaborative approach to managing complex systems”, Dorset House Publishing Co., Inc, 2000.

[10] Coad, P., J. deLuca, et al., “Java Modeling Color with Uml: Enterprise Components and Process with CDROM”, Prentice Hall PTR, 1999.

[11] Palmer, S. R. and M. Felsing, “A Practical Guide to Feature-Driven Development”, Pearson Education, 2001.

[12] S.R. Palmer, J.M. Felsing, “A Practical Guide to Feature-driven Development”, Prentice Hall, Upper Saddle River, NJ, ISBN 0- 13- 067615-2, 2002.

[13] A. P. Framework and I. DSDM, “Introduction Introducing the DSDM ® Agile Project Framework”.

[14] Agile Modeling (AM) Home Page, “Effective Practices for Modeling and Documentation” (http://www.agilemodeling.com/).

[15] Ambysoft. “The Agile System Development Life Cycle (SDLC)”, http://www.ambysoft.com/essays/agileLifecycle.html#Cycle0, 2011.

[16] A. Cockburn, “Crystal Clear: A Human-Powered Methodology for Small Teams”, Addison-Wesley, ISBN 0-201-69947-8, 2004.

[17] Waters, John K, "Agile lands role in games and business software", 2008.

[18] http://www.ambysoft.com/unifiedprocess/agileUP.html.