RESEARCH ARTICLE OPEN ACCESS

Comparative Analysis of Agile Software Development Methodologies-A Review

Kiran Hiwarkar [1], Aditya Doshi [2], Rahul Chinta [3], Manjula R[4]

[1,2,3]*(Post Graduate Students Department Of Computer Science Engineering, VIT University, Tamilnadu,India)*
[4]*(Associate Professor At Department Of Computer Science Engineering, VIT University, Tamilnadu, India)*

ABSTRACT

Software development methodology consists of dividing the software development process or work into different phases in order to achieve better planning and management. As the software development proceeds, factors such as requirements, needs, priorities, underlying technology may change. Thus development process must be highly dynamic and a good software development methodology must adapt to these evolving and changing requirements. Traditional software development models are unable to handle such dynamic requirements. To cope up with such dynamic requirements a set of software development methodologies referred as ‘Agile Software development methodologies’ are used. Comparison between different Agile software development methods will help in the selection of appropriate development model given a particular scenario.

Keywords: ASD, DSDM, FDD, Scrum, XP

I. INTRODUCTION

Today’s situation of software development with frequently changing requirements, where the traditional method fails in many scenarios. The most common traditional approach for developing software is “Waterfall Model”. The major drawbacks of this method are, it uses downward approach for development, development of complete software at one time, most of the energy is wasted in the documentation and taking signs and approvals and in the traditional approach we can’t move next until we complete previous stage, almost one third of the time and resources are spent on the testing [4] this scenario leads to delay in the delivery of the final product. Now a day’s software developing and maintaining plays an important role in all the industries. So the development need to more accurate and fast thus methodology need to be changed according to the changing scenario.

Converse to the traditional approach the “Agile Methodology” works better in the current scenario of frequently changing requirement.” Agile” means fast and incremental approach for developing projects with strong focus and customer’s involvement [3]. In this methodology documentation of the project considered as the non-productive work as it has no direct value at the end of the software development. Agile system is with little paper work oriented and more focused on the coding. In this methodology source code is considered as the way of communication and documentation between man to computer and people to people [3]. Agile is nothing but group of the methods which are mostly based on the iterative work flow. Main moto of the agile methodology is continuous and fast delivery of the project. This methodology is not prediction

oriented but it is more adaptive. Agile is designed in such a way that it works well in the frequently changing requirements [2]. The main advantage of this method over the traditional one are increase in the development speed and reduction in the cost.

Some of the method which follows the Agile principle but in practice works slightly different are “Extreme Programming”, “Lean Software Development”, “Scrum”, “Adaptive Software Development”, “Dynamic System Development Method,” “Crystal”, “Kanban”, “Feature Driven Development” [5]. This survey is mainly focused on the comparison of the widely used agile methods such as “Extreme Programming”, “Scrum”, “Adaptive Software Development”, “Dynamic System Development”, “Feature Driven Development”.

II. LITERATURE REVIEW

Software development methodologies are in use since the very inception of software industries. The term software engineering was result of “software crisis” [18]. The Software crisis presents several issues that were prominent in the software industry in 1960s, 1970s, and 1980s. Some of the important issues were: -

- Projects not being completed within given time and budget.
- Inefficient and low quality product.
- Product Unable to satisfy customer requirements.

Ensure that any miscellaneous numbering system you use in your paper cannot be confused with a reference [4] or an equation (3) designation.

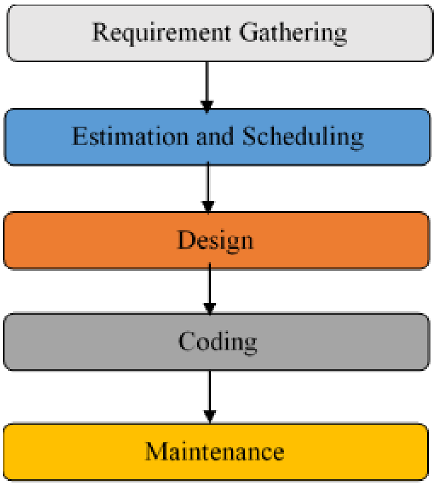

Fig 1. Traditional Software Development Life Cycle

To addresses these issues in a systematic and structured way, software development methodologies (SDM) were proposed in late 1960s. With SDM the software development process was viewed as a set of phases consisting requirement gathering, designing, implementation, validation and verification and maintenance. The systematic approach to the software development addressed (to some extent) the issues defined by "Software crisis". Several SDMS were proposed over the time, each having its own advantages and disadvantages. Each SDM is suitable for a particular kind of project based on factors such as type and complexity of project, time and available resources, team size etc. Traditional software development methodologies such as waterfall model, spiral model, RAD (rapid application development) are serial models in which development process moves step by step. It begins with gathering the requirements from the user. Then, the architecture of the product is pictured and finally coding phase begins. Following figure depicts the general steps followed by traditional software development models.

The important factor which needs to be considered with traditional models such as waterfall model is end user requirements. For the success of the project it is important to have clear understanding of the customer requirements. All the requirements must be specified clearly at the beginning. Vagueness or uncertainty in the requirement specification may lead to bad quality product and any lateral chances in the requirements are not entertained. In most of the traditional models it is not possible to move to next phase or step until the previous phase is completed and the work is validated. They also involve heavy documentation.

To deal with the changing requirements the concept of Agile software development was proposed [16]. Agile software development methodologies follow iterative incremental approach to develop a high quality product within the assigned time and budget and are capable of handling changing requirements. Agile software development views the software product as combination of small modules that communicate and collaborate with each other to realize a large software structure [5].

The Manifesto for Agile Software Development, also known as the Agile Manifesto, was presented in 2001(agile alliance). In February 2001, 17 software developers published the Manifesto for Agile Software Development which says "uncovering better ways of developing software by doing it and helping others do it" [16]. The agile manifesto presented 12 principles to ensure delivery of valuable software product among which core principles were:

1. High priority to customer satisfaction.
2. Adapting to changing requirements.
3. Promoting communication within the development team
4. Frequent interaction and with the customer.

Agile software development emphasizes on quick and valuable delivery of the small working units of product and improvising it by adding features or functionality as required in the subsequent iterations or increments. Unlike traditional models it promotes the communication within the teams and among individual involved rather than documentation. It is a result oriented methodology which adapts to the changing requirements, ensures delivery of a high quality product through continuous unit and integration testing, and promotes parallel development and testing of the features.

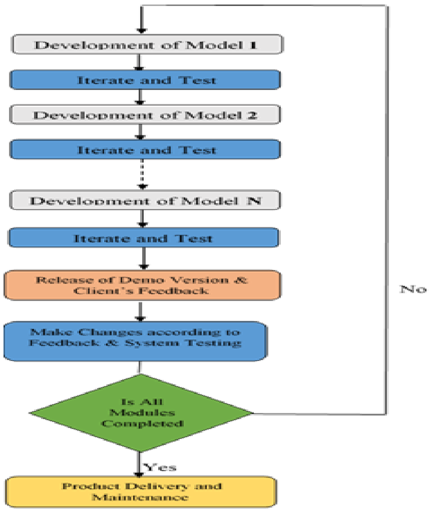

Fig 2.Agile Software Development Life Cycle

Several Agile methodologies were proposed over time such as XP, Scrum, FDD, DSDM, ASDM, Crystal, Lean etc. each of which is suitable for developing a specific kind of product.

III. OVERVIEW OF AGILR METHODOLOGY

Different Agile Methodologies have been presented over the time that is suitable for a particular domain. Popular ones are:

a) Scrum
b) XP
c) DSDM
d) ASD
e) FDD

3.1 Scrum

Presented by "Jeff Sutherland, Ken Schwaber, Mike Beedle" in 1996 Scrum is the most widely used agile development methodology [19]. Scrum basically addresses the managerial issues of product development rather than the technicality of the project. Thus more emphasis is given to the managerial skills of the personnel involved such as project managers as well as developers.

Scrum is light weight methodology that follows an incremental approach for software development [5]. It proceeds by implementing small requirements within 2 to 4 weeks of iteration known as "sprints" and iterate the process to achieve the core objective. Prime Importance is given to regular standup meetings in order to effectively manage the sprint. Each sprint is followed by testing and risk analysis and thus reducing the overall project risk. Scrum teams usually consist of 6 to 10 members and communication among the team members and between the teams are important. Scrum suits well to small projects.

3.2 XP

XP (extreme programming) model was presented by "Ward Cunnigham, Ron Jeffries, Kent Beck" [5]. It is a simple development model that effectively handles the uncertain of frequently changing customer requirements. It emphasizes on the engineering practices in order to develop a quality product and quickly adapt to the unstable or changing requirements.

XP developer teams follow the concept of pair programming. Pair programming refers to a pair of developers or programmers working together on a single workstation where one (the driver) is responsible for coding and other (the observer) is responsible for observing the code line by line as it is typed in and suggest improvements or changes if any [1]. Due to the practice of pair programming success of the project largely depends on the communication among the team members. Teams are collocated and project requires on site customer so that regular feedback can be gained.

3.3 DSDM

"Dynamic System Development Methodology" was firstly presented in 1995 by "Stapleton" [5]. In 1994 very huge number of projects combined which were required to be rapidly developed. DSDM mainly focus on delivering the projects as early as possible without affecting the quality of the project. This methodology is the agile methodology thus it also works on the same principle as agile that is incremental as well as iterative method. The project development cycle of this methodology is modified such a way that cycle of project management combined with project development cycle [5]. This method creates prototype as early as possible to make sure that to proceed to next step or not. Function prioritizations, analysis of the risk, documentation of prototype of a function are outcomes of the phase of this method [6].

This methodology is more suitable for the projects whose requirements are very frequently changed rather than project with well defined requirement, project which are real time. This methodology has good control over quality of product, risk, cost and time[5]. This method's iteration length and size of team varies depending on size of the project. Testing in this methodology is done throughout the projects life cycle. The most famous prototyping technology "MoSCoW" [5]. This method is quit heavier as compared to other agile methodologies as it is more restrictive than others.

3.4 ASD

In the year 2000 Jim Highsmith and Sam Bayer presented Adaptive software development methodology [5]. It basically addresses the managerial activities of the project and thus limited to the same has evolved from RAD (rapid application development) model and gives prime importance to rapid or quick development of software product [7]. It is an incremental and iterative development strategy which involves constant prototyping of the product according to the user requirement. ASD follows three important phases: Speculate which involves initialization and planning out the project, collaborate which involves concurrent development of the products functionality or features and Learning which means reviewing the quality of each functionality or feature before and after integration. It is preferred for large and complex projects.

3.5 FDD

"Feature-Driven Development" designed by "Jeff De Luca" and "Peter Coad" in 1999. "Felsing"and "Palmer" describes about FDD in 2002 [5]. This models working principle is also same as the other which is nothing but incremental and iterative but varied in some of the factors. These model mainly focused phases are building and design. Development of overall model, building a list of features, Plan of development, package design, completion of client valued functions. This model includes fast deliveries as well as perfect monitoring. FDD is mostly not suitable for vary large project it woks good for small and average type of projects. Team size changes depending on the size of the project. Length of iteration is up to fourteen days. If iteration length exceeds over the given period, then models are broken down in to sub modules.

IV. COMPARATIVE ANALYSIS

Even though several Agile methodologies are available, each applies to a specific set of project. A Software development project has several factors associated with it such as project size, complexity, allotted time and budget etc. Selecting appropriate methodology for software development depends on such factors. Thus a comparative analysis of agile methodologies will help to decide which can be used under a given situation.

4.1 Documentation

One of the main principle of agile software development is to reduce the amount of time and effort spent on documentation. But documentation being important cannot be completely removed.

- While using methods like Scrum, XP and ASD documentation is of least importance.
- Compared to others projects under FDD requires more documentation.
- DSDM requires moderate level of documentation which is still less than FD.

4.2 Interaction with Customer

Agile gives prime importance to frequent communication with the end users. Still the degree of involvement is different in each methodology.

- XP and Scrum have high involvement of customer in the development process.
- In ASD and DSDM customer or end user involvement can be seen during the start and end of the iteration.
- While FDD uses reports to communicate with the customers.

4.3 Meetings

Communication is one of the core principles stated in the Agile manifesto[16]. Success of the agile methodologies depends on the effective communication among the team members.

- The meetings are of informal nature and no documentation is maintained.
- Due to use of pair programming technique Success of XP largely depends on the communication.
- FDD and DSDM relies on reports and documentation for the communication.
- Face to Face meetings are used for communication in ASD.

4.4 Size and complexity of the project

Each methodology is suitable for a particular kind of project

- XP and ASD are usually preferred for small and less complex projects.
- XP is suitable for projects where there is constant change in the product specification.
- Scrum, FDD DSDM can be applied to any size project.

Table 1 compares above methodologies with respect to different parameters.

V. CONCLUSION

Agile methodologies are gaining popularity and are now preferred over traditional software development methodologies which have several shortcomings such as inability to cope up with the constantly changing user requirements and exceeding the allotted time and budget. With traditional software development models product requirements must be clearly specified beforehand. Considering the current business environment, it is important that the development methodology used easily adapts to the frequently changing end user requirements.

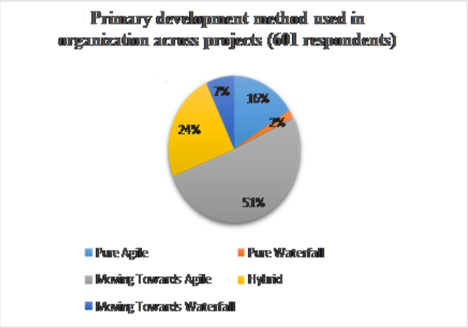

Fig 3. Survey on Agile Methods

Agile software development methodologies handle the evolving customer requirements through iterative and incremental approach. It has shorter development iteration with each iteration or increment followed by testing and risk analysis which results in faster development and delivery of a quality product.

Characteristics	**XP**	**SCRUM**	**DSDM**	**FDD**	**ASD**
Approach	Iterative, Incremental	Iterative, Incremental	Iterative	Iterative	Iterative, Incremental
Iteration Cycle Period	1-6 weeks	2-4 weeks	In 20% percent of total time 80 % of product.	2 days-2weeks	4-8 weeks
Suitable project size and complexity	Small and simple project	For large and complex problems.	All types of complex and as well as simple project.	Large scale projects	Smaller and simple projects
User Involvement	Actively involved	Through product owner	Through frequent releases	Through reports	Through frequent releases
Documentation	Basic Documentation	Basic Documentation	More than XP and Scrum	Highest among all	Basic Documentation
Major Practices	Simplicity, Pair programming Test driven development	Scrum meetings	Time boxing, MoSCoW, Prototyping	Object Modeling, Development by feature, use of UMl diagram	Time boxing, Risk Driven, Feature based.
Concurrent feature development	Possible	Possible	Possible	Possible	Possible

Table 1: Comparison of Agile Methodologies

According to a survey conducted by HP (Hewlett-Packard) around 51% of the industry is leaning towards the Agile software development methodologies [17]. Selection of appropriate agile methodologies is important to maximize the probability of delivery of a high quality of product that meets the end user requirements. The comparison done in this paper can be used to decide which methodology can be adapted to a particular project.

REFERENCES

[1]. M. Jameel Qureshi, "Agile software development methodology for medium and large projects", IET, vol. 6, no. 4, pp. 358-363, 2012.

[2]. R. H P and A. P Patil, "An Improvised Prespective to Agile Methodology", IEEE 3rd International Conference on System Engineering and Technology, 2013.

[3]. A. Sharma, V. Deep and N. Garg, "An Efficient way of Articulation or Suppression in Agile Methodologies".

[4]. A. Sultania, "Developing Software Product and Test Automation Software Using Agile Methodology", IEEE, 2015.

[5]. H. K Flora and S. V Chande, "A Systematic Study on Agile Software Development Methodologies and Practices", International Journal of Computer Science and Information Technologies, vol. 5, 2014.

[6]. A.,AdilaFirdaus, S. Jeong and I. Ghani, "A Review on Software Development Security Engineering using Dynamic System Method (DSDM)", International Journal of Computer Applications, vol. 69, 2013.

[7]. M. Alnoukari, Z. Alzoabi and S. Hanna, "Applying Adaptive Software Development (ASD) Agile Modeling on Predictive Data Mining Applications: ASD-DM Methodology", IEEE, 2008.

[8]. M. Sadiq and T. Hassan, "An Extended Adaptive Software Development Process Model", IEEE, 2014.

[9]. D. Rover, R. Scheel, C. Ullerich, J. Wegter and C. Whipple, "Advantages of Agile Methodologies for Software and Product Development in a Capstone Design Project", IEEE, 2014.

[10]. K. Kaur, A. Jajoo and M, "Applying Agile Methodologies in Industry Projects: Benefits and Challenges", IEEE_ International Conference on Computing Communication Control and Automation, 2015.

[11]. A. Begel and N. Nagappan, "Usage and Perceptions of Agile Software Development in an Industrial Context: An Exploratory Study".

[12]. J. Noll, "A Survey of Empirical Studies of Extreme Programming", Computer Engineering Department, Santa Clara University.

[13]. D. Karlström, "Introducing Extreme Programming – An Experience Report".

[14]. M. HNEIF and S. HOCK OW, "REVIEW OF AGILE METHODOLOGIES IN SOFTWARE DEVELOPMENT", International Journal of Research and Reviews in Applied Sciences, vol. 1, no. 1, 2009.

[15]. K. Pathak and A. Saha, "Review of Agile Software Development Methodologies", International Journal of Advanced Research in Computer Science and Software Engineering, vol. 3, no. 2, 2013.

[16]. AgileManifesto, "Principles behind the Agile Manifesto". [Online]. Available: http://www.agilemanifesto.org/iso/en/principles.html.

[17]. Agile is the new normal, [Online]. Available: http://www8.hp.com/h20195/v2/getpdf.aspx/4AA5-7619ENW.pdf?ver=1.0.

[18]. Wikipedia, [Online]. Available: https://en.wikipedia.org.

[19]. Schwaber, K. and M. Beedle, “Agile Software Development with Scrum”, Prentice Hall PTR, 2001.