A Survey on Comparative Analysis of Agile Software Development Methodologies

Ajith Jerom B[1*], Rajamohana SP[2]
PG Scholar[1], Assistant Professor[2]
Department of Information Technology
PSG College of Technology, Coimbatore, Tamil Nadu
***Corresponding Author's Email**: ajith.jerom.95@gmail.com[1*], monamohanasp@gmail.com[2]*
DOI: http://doi.org/10.5281/zenodo.2561994

Abstract

Software development methodologies is the process of dividing the software development process into different phases in order to achieve better greater planning and management of the software. For developing software's there consists of many methodologies available But Agile Methodology is one of the methods which were introduced to meet the customer's new requirements for developing software product. This paper is going to survey about the comparative analysis of agile software which contains different kinds of agile methodologies. Some of them are Scrum, Kaban, Extreme Programming, Lean, Feature Driven Development (FDD), Adaptive Software Development (ASD) and Dynamic Systems Development Method (DSDM), etc. and describes about the differences between them and recommends where to use and when to use. According to recent survey there were nearly 14 different Kanban boards were available and also the survey explains how the features are represented in the boards and it was compared with 22 software tools for implementing them to virtual Kanban boards, for analyzing there sources available on the web.

***Keywords**: - ASD, XP, DSDM, FDD, Scrum, Kanban, Agile Methodologies*

INTRODUCTION

Various numbers of agile methodologies [1] have been developed in the past fifteen years that is being differentiated to traditional trained approach for individual values and Co-operation over the processes

and tools. Before developing software the first thing to be done is customer coordination over meetings, and gathered some additional information to change the following plan [2] in the development based on the customer requirements. The word Agile which, means to move rapidly and easily and this is the reason why agile software development technology is mostly referred for the software development. Now a day most of the leading software leading companies use agile software methodology was accepted and used. Agile methodology is one of the kinds of iterative approach which makes to keep necessary action with dynamic development environments. Owing to the past decade, among various kinds of software development approaches a new software development approach were found applicable to new cultures in the software development either product based or service based companies. The objective for most of the software companies tries to produce high quality software within a short period of time with low costs, unstable, and changing environments with help of technology. So among them agile Methodologies were found to meet the new requirements given by the customer or stakeholders during development period in the software development companies [3]. When the Comparison has been made with traditional approach "Agile Methodology" founds to be work better for the current scenario for frequently changing requirements. It is one of the kinds of incremental approach for developing the product with a strong focusing with customer's involvement [4].During the creative process Agile software development (ASD) method expects the software to be in flexible and applies into a level of practicality for the complete delivery of the product to the customers. It keeps on focusing the product code to be simple to use and testing the code frequently, and the product will be delivered when functional bits of the application were made to be ready. The objective of ASD is to keep build upon the small scale customers approved parts for the project progresses, as they would opposed to deliver while, any one of the large application at the end of the project schedule. It is one of the lightweight software engineering frameworks that elevate iterative development during the life-cycle of the project. Agile software development is a group of software development methods that is based on the combination of iterative and incremental development process, where requirements and solutions for the product would be developed through connection between self-organizing and cross functional teams. Among many software development

models it seems to be one of the most popular models getting used in the software development industry. It provides the developers to help in speedy changes in the software development and produce great results in the forms of better quality and reusability. As it leads to upgrade in an adaptive planning, evolutionary development of the product delivery in a time-boxed iterative approach, and encourages quick and flexible response to change based on requirements. This survey paper mainly focus about the different type of Agile Methodologies in Software Engineering.

RELATED WORK

Andreas Schmietendorf [5] defined an investigation with the effort estimation possibilities based on agile software development methodologies. Among the various software development process agile methods becomes more popular and increasingly used in the industrial projects. For many kind of application the effort estimation methods tends to be very difficult but it an important task in the project. So, the Classical estimation methods were needed for the well-defined requirements based on the customer's usage. It mostly provides the major investigation about possibilities in the product, particularly for the extreme programming worldwide. Most of the XP ventures were built on the basis of the effort estimation method. Xiaofeng Wang [6] in the last few years agile methods such as Extreme Programming (XP) and Scrum to lean software development were illustrated, which was being suggested as "agile to lean". The scope of this research was to explore how agile methods and lean approaches had been joined in software development process. The designs of joining those reports for agile methods and lean approaches were explained and sorted in a more global way. Pankaj Kamthan [7] explained as many number of different changes were demonstrated in many industrial software engineering, which enables the movement towards term agility. The word agile declaration is to identify the term "agile" which would lists the basic set of principles that forms a premise for agile methodologies that enables some which are Extreme Programming (XP), Scrum, and Open UP. For the success of project through agile methods, it was critical because the knowledge be managed effectively and efficiently. Fahad Almudarra and Basit Qureshi [8] illustrated a new idea that a mobile based cloud application development can be integrated with agile development methodologies for contents in the media management which, helps in reducing cost, time and gives better

software quality and changes can be made to improve software quality too.Among various methods results showed the positive indications for being applied with Extreme Programming (XP) which has capable of integrating for both mobile and web application with hybrid cloud. R. Steven Wingo and Murat M. Tanik [9] developed software development which makes difficult task with more challenges to get successful outcomes for complex problem domains. The XP Methodology seems to be one of the good choices for a skilled and disciplined software development team for use in creating software solutions mostly for complex problem domains. One of the critical problem domains such as core business systems, XP would be mostly preferred to be used for delivering the software with high quality that keeps with changing the environment.

Maria Paasivaara and Casper Lassenius [10] explained their common interest to the group of experts and want to deepen the knowledge to group of experts in practice. It illustrates about the Ericsson with 400 persons divided into 40 Scrum teams at three sites to adopt the use of Communities of Practice (COP) as a unit of transformation from a conventional plan driven based organization to lean and agile. If organization handles those case, COPs were the major part to support agile transformation and distributed Scrum implementation too. Sergio Galvan etal. [11] Established software process standards (e.g.ISO/IEC 12207, ISO/IEC 15504) and models (e.g. CMMI) which provides good guidelines and practices to improve the quality of software if they were any changes in the requirements suddenly. In this research, the particular issue has been studied in the compliance of Agile Software Development Methodologies like SCRUM, XP, and UPEDU and includes the new ISO/IEC 29110 standard too. The major findings indicate that the UPEDU and SCRUM methodologies were presented at the high level of compliance with the ISO/IEC 29110 in the project Management process. David P. Harvieet al. [12] described that these two fields software engineering and mission command were independent but with similar fields, as both were used for solving complex problem domains in the environments with rapid and ever changing requirements in the product. The research theory was one of the great modifications to agile software development based on motivations from mission command. Based on three motivation from the mission command modification of traditional scrum was made for targeted scrum: End State, Line of

Effort, and Targeting. Martin Tomanek and Tomas Klima [13] presented the penetration testing which, was embedded into the scrum framework that represents most of work has been done with the agile software development framework. One of the major advantages of scrum was to automate the penetration tests during the software development projects that incorporate some of the specific penetration tests into the regular software product release which would improve the overall resistance of developing software. The main focus on the research is to develop the penetration testing methods such as PETA. Bruno Antunes [14] proposed an analyzed waterfall vs. agile methodology which uses agile methodology that is nothing but scrum. It focuses on a solvable approach of virtual team, as well as tends to show some examples with use of Microsoft Visual Team Foundation server tools to identify the challenges. The main focus on the Software development environment is on the dictated speed, flexibility.

OVER VIEW OF AGILE METHODOLOGIES

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

- Scrum
- XP
- DSDM
- ASD
- FDD
- Kanban

SCRUM

Among various kinds of Agile methodologies Scrum was one of the most widely used agile development method [15] during the year 1996. Basically Scrum was mainly used to addresses the most important managerial issues in the product development instead of the technicality in the project work. Therefore more significance is given to the managerial skills of the team involved which include project managers as well as developers too.

Scrum is light weight methodology that follows an incremental approach for software development process [16]. For small requirements only 2 to 4 weeks were need for implementing the iterations that are commonly referred as “sprints”. The regular standup meetings were conducted to effectively and efficiently manage the sprint phase. Each and every phase was tested thoroughly by the testing tram in order to reduce overall project risk. Actually the scrum has only 6 to 10 members in the range. In this method communication is very important among

the team members and between the teams. So, Scrum is more suitable to small projects.

XP

XP (extreme programming) method which seems simple for development and it was mostly used to handle the frequently changing requirements form the customer. This model is suitable for quickly changing unstable requirements during the software development and it is more flexible. It underscores the engineering practices in order to develop a quality product.

The development team follows the pair programming concept. It refers the pair of developers working together in a single workstation. Among the team one of them will be responsible for coding part and other is responsible for viewing the code line by line in the program and suggestions will be given if any necessary to improve the product. [17]. In order to this kind of practice using pair programming concept the victory of project majorly depends upon the communication between team members and the teams are collaborated with customer to get regular feedback.

DSDM

DSDM is another kind of software development method which, focus on product delivery as soon as possible without affecting the product quality. This method works as the principle as agile that is incremental as well as iterative model. In this method a slight modification were made in the development cycle that is in this methodology cycle of project management can be combined with project development cycle [16]. One of the biggest advantages of this method creates the prototype as early as possible that helps to proceed to move on next step or not. Risk analysis, Function prioritizations and documentation for the prototype are the outcomes of this phase. [18]

Furthermore this method is mostly suitable for the projects whose customer requirements are very habitually changed rather than project with well-defined requirement. This method has a greater control over product quality, risk factor, effective with cost and time [16]. The length and size of the team varies based on the size of the project. Testing can be done throughout the entire project life cycle. While compare to other methods it is little bit quite heavier than other agile methodologies and also it is more restrictive than others.

ASD

In the year 2000 Adaptive software development was introduced into the software industry [16]. This method mainly focuses in the managerial activities which, is related to Rapid Application Development model having some limitations and gives more important to quick development of the product [19]. This model is also based on incremental and iterative approach and so the prototyping were made according to the user's wish.ASD mainly follows only on three important phases they are Speculate which process in initialization and planning for the project the second is collaboration which focus in the concurrent development of the products or features and the final one is learning which means reviewing the quality of the product. This kind of methods is preferred only for large and heavy complex software projects.

FDD

Feature-Driven Development model was designed by Jeff De Luca, Peter Coad. Felsing, Palmer 2002 [16]. This model works similar like other model that is nothing but incremental and iterative but having variations in some of the factors. This model mainly focuses on building and design phases. By using this model the delivering of the products were quick and also well suited for perfect monitoring. This model is mostly not suitable for very large project because it works better only for small and average type of projects. The size of the team changes depends upon the size of the project.

KANBAN

Kanban has become popular in recent years because, of its ease of implementation, usage of visual controls, works well in progress management, and relentless focus on the continuous process improvement.

GENERIC KANBAN BOARD

The aim of the generic Kanban board is to provide a full range of activities in the development process and approaches to board modeling during the working states, buffers etc., which must be taken into account while deciding about the most suitable board structure.

COLUMN DESCRIPTOR

The proposed Kanban board consists of the following columns factors:

- The "Product Backlog" column which includes all user stories that are being currently known. The product owner will prioritize each user stories from the customer; stakeholders and estimated it using planning poker [22, 23]. Whenever a new story enters into the

product development that will be added at the end of the product backlog.

- The “Sprint Backlog” column contains the user stories belonging to the current Sprint. The content inside this column will be initiated during the Sprint planning meeting when the Product Owner and the development team agree which stories have to be developed in the next Sprint. During Sprint, the stories are moved to subsequent columns in according with Kanban pull mechanism. If it is executed properly, then this column becomes empty at the end of the Sprint.

- The “Next” column contains limited number of high priority user stories. When the member in the development team is ready to start working on a new item, he/she can take a user story from the “Next” column based on the high priority and move it to “Analysis & Design”.

- The “Analysis & Design” column reflects in the Scrum approach to just-in-time design where each user stories are splitted into constituent tasks. After that the “Analysis & Design” column is split into two sub-columns Ongoing and “Done” in order to distinguish between user stories being still under in a working state that is represented as “Ongoing" and user stories for which analysis and design were completed done. In such a way, that the sub-column Done acts as an inter-process buffer providing information which user stories can be pulled to the next step in process. The same would be hold for Ongoing and Done sub-columns within Development, Testing, and Documentation.

- The “Development” column shows what is being developed at the moment. It is assumed that development phase includes coding and unit testing performed by programmers. Therefore, development is considered to be done when all unit tests were passed and functional testing get started.

- The “Testing” column decides which user stories have to be tested first This step have a part of functional and integration tests performed by the development team in order to verify whether the user stories comply with the Scrum concept of done. Stories in the sub-column “Done” are considered to be completed and ready for documentation.

- The "Documentation" column contains user stories to convert them s document is done the documentation phase. Scrum requires that the user operation of the new functionality is documented, in user documentation [24]. Stories in the "Done" sub-column are considered to be completed and ready for presenting to the Product Owner and other stakeholders during the Sprint review meeting.

- The "Acceptance" column is responsible for final acceptance, after the testing performed by the Product Owner. According to Scrum rules, the Product Owner will be responsible to decide whether the user stories implementation meets all requirements or not and can be put in production.

- The "Deploy" column would show which user stories are going to be deployed.

- The final "Done/Live" column contains the user stories that were accepted by the Product Owner Done and made available for use ("Live.

COMPARATIVE ANALYSIS

Even though there were lot agile methodologies are being available for the product development process, but each method applies only for specific set of project. A Software development projects would have several factors associated with the size of the project, complexity, allotted duration of time, budget for the project etc. Selecting suitable methodology for particular project is very important in the software development. Thus the comparative analysis of agile methodologies will help the project team members to decide which can be used in a given situation.

A) DOCUMENTATION

One of the main principles in agile software development process is to reduce the time and effort spent on documentation.

- Using some methods like Scrum, XP and ASD documentation is to be given less importance.

- Compared to others projects FDD require more documentation.

- DSDM requires a moderate level of documentation which is less than FDD.

B) INTERACTION WITH CUSTOMER

Agile methods give more importance for frequent communication with the end users.

- During the development process methods like XP and Scrum have highest involvement with the customer.
- In ASD and DSDM the involvement with the customer or end user can be seen only at start and end of the iteration.
- FDD uses the reports to communicate with the customers or end user.

C) MEETINGS

Communication is the core principles stated in the Agilemanifesto [20]. The success of the agile methodologies depends only on the efficient communication between the team members.

The meetings are done informal in nature and no documentations were maintained.

- While the use of pair programming technique success of XP only depends on the communication.
- FDD and DSDM mainly rely on reports and documentation for the communication.
- In ASD Face to Face meetings are used for communication.

D) SIZE AND COMPLEXITY OF THE PROJECT

For particular kind of project certain methodology will be suitable

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

CONCLUSION

Agile methodologies are becoming more popular in the recent years and are preferred over traditional software development methodologies which have several shortcomings such as inability to cope up with the constantly changing with the requirements from the user. With traditional software development models product requirements must be clearly identified before to the hand. While considering with the current business environment, it is important that the development methodology were used for easy adapting to the frequently changing of the requirements from the user.Agile software development methodologies are used to handle such a kind of frequently changing requirements through iterative and incremental approach. It has shorter development iteration with the each

iteration or increment followed by testing and risk analysis which delivers the results in faster development and delivery of a quality product. According to recent survey conducted by HP (Hewlett-Packard) around 51% of the industries are moving towards the agile software development methodologies [21]. Selecting an appropriate agile methodology is important which would maximize e the probability for product delivery with a high quality of product that meets the requirements of the end user's. The comparison has been done with various approaches in this paper which would be more useful to decide which methodologies can bemore suitable to a particular project.

REFERENCES

I. L. Williams, Agile Software Development Methodologies and Practices, Advances inComputers, Vol. 80, 2010, pp. 1–44.

II. Manifesto for Agile Software Development, http://www.agilemanifesto.org (last accessed 7.11.2013).

III. Agile Alliance. Manifesto for Agile Software De velopment. [Online] Retrieved 16th March 2009. Available at: http://www.agilemanifesto.org.

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

V. Andreas Schmietendorf, Martin Kunz and Reiner Dumke, "Effort Estimation for Agile Software Development Projects", Proceedings 5thSoftware Measurement European Forum, Milan 2008, pp. 113-126, 2008.

VI. Xiaofeng Wang, "The Combination of Agile and Lean in Software Development: An ExperienceReportAnalysis", IEEE Computer Society, Agile Conference, Lero – the Irish Software Engineering Research CentreLimerick, Ireland, pp. 1-9, 2011.

VII. PankajKamthan, "On the Role of Wiki forManaging Knowledge in Agile Software Development", Department of Computer Science and Software Engineering Concordia University Montreal, Canada, pp. 622-623, 2013.

VIII. FahadAlmudarra and BasitQureshi, "Issues in adopting Agile Development Principles for MobileCloud Computing Applications", Elsevier B.V., 1st International Workshop on Mobile

Cloud Computing Systems, Management, and Security (MCSMS-2015), College of Computer and Information Sciences, Prince Sultan University, Riyadh, Saudi Arabia, Vol. 52, pp.1133-1140, 2015.

IX. R. Steven Wingo and Murat M. Tanik, "Using anAgile Software Development Methodology for aComplex Problem Domain",Proceedings of the IEEE SoutheastCon, University of Alabama Birmingham, 2015.

X. Maria Paasivaara and Casper Lassenius, "Communities of practice in a large distributed agile software developmentorganization – Case Ericsson", Elsevier B. V., Aalto University, Finland,pp. 1556-1577, 2014.

XI. Sergio Galvan, Manuel Mora, Rory V. O Connor, Francisco Acosta and Francisco Alvarez, "A Compliance Analysis of Agile Methodologies with the ISO/IEC 29110 Project Management Process", Elsevier Ltd., Conference on Enterprise Information System, pp. 188-195, 2015.

XII. David P. Harvie, Graduate Student Member, IEEE, andArvinAgah, Senior Member, IEEE, "Targeted Scrum:Applying Mission Command to Agile Software Development", IEEE Transactions on Software Engineering, Vol. X, No. Y, pp.1-14, 2015.

XIII. Martin Tomanek and Tomas Klima, "Penetration Testing In Agile Software Development Projects", International Journal on Cryptography and Information Security (IJCIS), Vol. 5, No. 1, pp. 1-7, March 2015.

XIV. Bruno Antunes, Diogo Santos, Eurico Lopes, FilipeFidalgo and Paulo Alves,"Blisstrail: An Agile Project Business Case Study", Conference on Enterprise Information Systems, Escola Superior de Tecnologia, InstitutoPolitecnico de CasteloBranco, Branco, Portugal, Vol. 64, pp. 529-536, 2015.

XV. Schwaber, K. and M. Beedle, "Agile Software Development with Scrum", Prentice Hall PTR, 2001.

XVI. 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.

XVII. M. JameelQureshi, "Agile software development methodology for

medium and large projects", IET, vol. 6, no. 4, pp. 358-363, 2012.

XVIII. 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.

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

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

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

XXII. J. Grenning, Planning Poker or How to Avoid Analysis Paralysis While Release Planning, 2002, http://www.renaissancesoftware.net/files /articles/PlanningPoker-v1.1.pdf (last accessed 28.10.2013).

XXIII. V. Mahnic, T. Hovelja, On Using Planning Poker for Estimating User Stories, Journal ofSystems and Software, Vol. 85, No. 9, 2012,pp. 2086–2095.

XXIV. E. Corona, F. E. Pani, A Review of Lean-Kanban Approaches in the Software Development. WSEAS Transactions onInformation Science and Applications, Vol. 10,No. 1, 2013, pp. 1–13.

Cite this Article

Ajith Jerom B & Rajamohana SP (2019) **A Survey on Comparative Analysis of Agile Software Development Methodologies** Recent Trends in Computer Science and Software Technology 4(1), 10- 12
http://doi.org/10.5281/zenodo.2561994