International Journal of Scientific and Technical Advancements
IJSTA
ISSN: 2454-1532

Review and Analysis of Applying Agile Methodology in Software Development

Syed Nisar Bukhari, Ashaq Hussain Dar, Muneer Ahmad Dar
National Institute of Electronics & Information Technology, NIELIT, J&K, India

***Abstract*—**In this era of digital world Software development plays a major role. Earlier software development teams used to follow traditional software models. In the last 30 years a number of approaches have been announced but only a few have survived to be used today. Using any of these approaches we need to find out as how successful we are in developing software? The primary goals in delivering a software are (1) Deliver software that meets the requirements and specifications(feature complete) (2) Deliver the software within the time frame promised (on time) (3) Ensure that the software is of high quality and bug free(failure free) (4) Ensure that the software is within budget. Achieving these four goals is not easy. It requires your entire software team to have a defined development / project management methodology, and tools to make implementing the methodology easy. When choosing your development methodology, you have many to choose from, each having advantages and disadvantages. As classical method of software development have many disadvantages, a new way of software development is explored here which is agile software development-an Iterative and Incremental development. In this paper we will explore the agile methodology to provide a clear understanding and complete overview of Agile software development culture and practices. For testing purpose we will use the online tool called Version One. Then one of the agile development methodologies 'Scrum' will be proposed. Later some findings/observations from this scrum methodology will be presented.

***Keywords*—** Agile, agile software development, cadences, scrum, software, sprint, team.

I. INTRODUCTION

When a computer software succeeds—when it meets the needs of the people who use it, when it performs flawlessly over a long period of time, when it is easy to modify and even easier to use—it can and does change things for the better [9]. The problem with most of the projects following traditional software development approach (sequential development) is that the requirements are always changing which is what we can say as poor requirement engineering. Also in sequential approach every phase of the project must be completed before the next phase can begin. So firstly developers gather all the projects requirements, then complete its architecture and design, then write the code and so on. There is hardly any communication between the groups that complete each phase of work. It assumes requirements of a system can be frozen (i.e., baselined) before the design begins [5]. Clearly it is hard to tell to a team of developers everything that needed to be in a final product before it was up. It follows the ―big bang‖ approach-the entire software is delivered in one shot at the end [5]. This entails heavy risks, as the user does not know until the very end what they are getting [5]. So a team might have built the software it was asked to build, but, in the time it took to create, business requirements have changed so dramatically that the product is totally irrelevant. Ultimately company is in loss because it has spent time and money but the output is of no use With agile our focus is on the repetition of abbreviated work cycles as well as the functional product they yield and that is why agile methodology is described as ―iterative and incremental‖ approach. In waterfall, development teams only have one chance to get each aspect of a project right. In an agile paradigm, every aspect of development — requirements, design, etc. is continually revisited throughout the lifecycle. When a team stops and re-evaluates the direction of a project every 2-4 weeks, there's always time to steer it in another direction. The results of this ―inspect-adapt‖ approach to development greatly reduce both development costs and time to market [7]. In essence, it could be said that the agile development methodology helps companies build the right product [7]. Agile software development methodologies accommodate frequently changing requirements in a tight schedule. That is agile process imples a light and adaptive process, nimble in response to changing needs [5]. There are number of agile Software Development methods which include mainly extreme Programming, Scrum, Agile Modeling etc and it has been observed and experimented that all the aforesaid methods are based on agile manifesto and they have their own life cycle for improving productivity and quality. AGILE SOFTWARE DEVELOPMENT PROCESS

II. AGILE SOFTWARE DEVELOPMENT PROCESS

The word agile denotes ―quickness, lightness, well coordinated and ease of movement.‖ It means that Agile Software development is about fast delivery of software with more ease of development. If for a project with frequently changing requirements and client ready for his involvement and to invest time agile methodology is the best option for that project. It is an incremental concept (iterations) to software development, performed in a highly shared manner by self-organizing teams that produces high quality software in a cost effective and timely manner, which meets the changing needs of its customer. An Iteration is nothing but a small set of the functionalities build on top of previous iteration. As per my work experience with agile way of software development is concerned there is lot of client involvement and face-to-face interaction. Each iteration is tested by client. Then after testing, the feedback from the client is implemented in subsequent iterations; thus minimizing risk and ensuring higher client satisfaction. Features are prioritized depending

Syed Nisar Bukhari, Ashaq Hussain Dar, and Muneer Ahmad Dar, "Review and analysis of applying agile methodology in software development," *International Journal of Scientific and Technical Advancements*, Volume 2, Issue 4, pp. 187-190, 2016.

on customer need, development risk etc. High priority features are developed first. After every iteration, the project priorities are again evaluated. The project teams are generally small and a lot of interaction and communication is there among team members. As entire team is actively involved, team is empowered to take decisions and no separate team is there to do the project management. Emphasis is on keeping things as simple as possible and being open to change. More emphasis is on developing the application only, and not on documentation. Simple and minimal documents are used to exchange the views. Customer gets to know regular and frequent status of the application. Since the methodology emphasizes rapid delivery, time-to-market is less. So the key functionalities can be available to use sooner. Delivery is defined by fixed timescale. So customer is assured of receiving some functionality by a fixed time period. More Testing is done, so better software quality is delivered. Agile approach to managing software projects is based on giving more value to the developers than to the process [4].To manage agile projects you can use an online product called VersionOne (http://www.versionone.com/).Today there are around 30,000 teams in world using this product everyday to plan and track their projects.

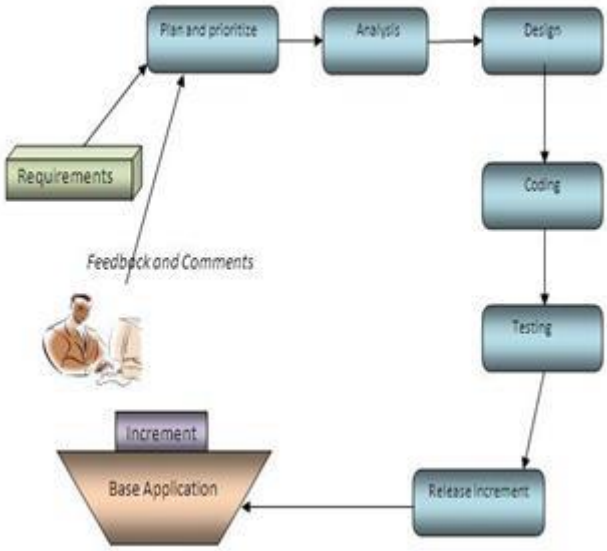

Fig. 1. Iteration cycle stages

From the above figure the iteration stages are repeated for each cycle and accordingly agile software development process involves the following:

- It gets initiated with kick-off meeting.
- Requirements which are known are understood and prioritized. Then development plan is drawn accordingly.
- Complexity of each requirement is estimated.
- Design using simple diagrams is done.
- An approach called Test Driven Development (TDD) approach may be used. This approach emphasis on ―writing test first and then writing code to pass the test‖. So it can help in avoiding over-coding
- Then development is done, sometimes in pairs, with lot of team interaction. Ownership of code is shared when pair programming is done.
- The code is tested more frequently.
- Depending on the feedback received, code refactoring is done. There is no impact on external behavior of the application due to refactoring, but the internal structure may be changed to provide better design, maintainability. Some ways of refactoring may be add interface, use super class, perform database to business object mapping in separate class etc.

III. SCRUM

Scrum is one of the agile development methodology that allows us to focus on delivering the highest business value in the shortest time. Scrum is unique because it introduced the idea of ―empirical process control‖ [9].That is, Scrum uses the real-world progress of a project — not a best guess or uninformed forecast — to plan and schedule releases. In Scrum, projects are divided into succinct work cadences, known as sprints, whose typical duration is a calendar month (at most) [9]. Designing, coding and testing of the product is done during the sprint. At the end of each sprin his document is a template. An electronic copy can be downloaded from the Journal website. For questions on paper guidelines, please contact the journal publications committee as indicated on the journal website. All title and author details must be in single-column format and must be centered. clients and team members meet to assess the progress of a project and accordingly plan its next steps. As my research and development experience with scrum is concerned I found it to be an incredibly effective management methodology for everyone involved, including developers and stakeholders. At the end of the iteration the system is demoed to the stakeholders to verify that the work that the team promised to do at the beginning of the iteration was in fact accomplished.

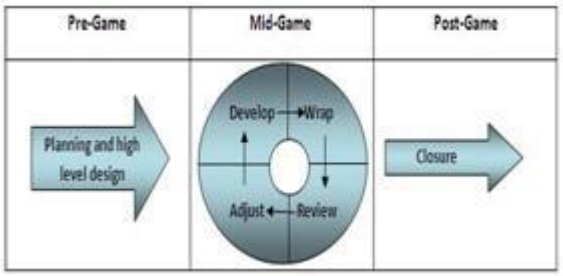

Fig. 2. The general scrum process.

So from the above figure scrum process generally has three main phases.

- *Planning*: Here project planning is done and also high level design decisions are made.
- *Sprint cycle*: An iterative cycle of one month duration at the most, in which the actual development of the product is done (see figure 3 below).Generally it starts out with a Sprint Planning Meeting to decide what features to be pulled from product backlog to sprint backlog (features to be completed in

Syed Nisar Bukhari, Ashaq Hussain Dar, and Muneer Ahmad Dar, "Review and analysis of applying agile methodology in software development," *International Journal of Scientific and Technical Advancements*, Volume 2, Issue 4, pp. 187-190, 2016.

this sprint). Sprint backlog is a table we derive from the product backlog that formalizes the schedule for the sprint (short- range schedule) [3]. Then the development is done. And sprint Review Meeting is held where the progress made in the last sprint is demonstrated, and adjustments are made to the project as necessary and with this sprint is closed. The sprint cycle is repeated until the product's development is complete .The product is complete when the variables of time, quality, competition, and cost are at a balance [9].

- *Develop* - code, test, and document.
- *Wrap* - Integrated the work and make it ready for evaluation.
- *Review* -do the review on the work done in this sprint.
- *Adjust changes* (if any) in requirements or plans.

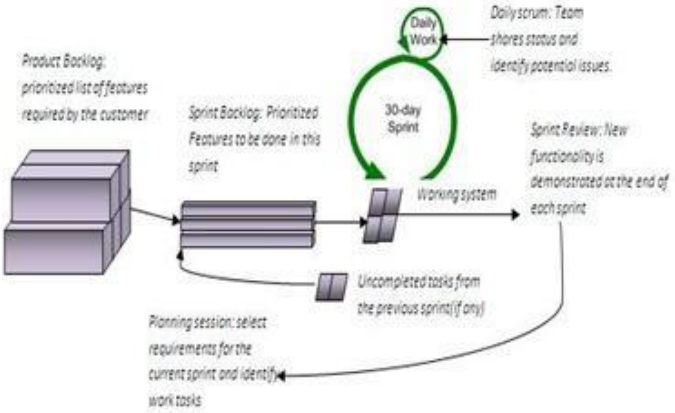

Fig. 3. The scrum sprint cycle

- *Closure*: software is released and development is brought to a close.

A) Scrum Roles

- Product Owner: Product owner defines the features of the system and decides on release date and content. Product owner also prioritize features according to market value and adjust features and priority every iteration, as needed. He also accepts or rejects work results. Product owner can be a customer or marketing representative or proxy between customer and the team.
- The Scrum Master: Scrum master Represents management to the project. He is responsible for enacting Scrum values and practices and removes impediments if any. The scrum master has to ensure that the team is fully functional and productive and shields the team from external interferences. Also scrum master enable close cooperation across all roles and functions
- Scrum team: Scrum usually consists of 5-9 people. Members in a team can be cross functional. That is they can be Programmers, testers, UI Designers, etc. Members must be full time but exceptions may be there (e.g., Database Administrator).Membership should change only between sprint.

B) Scrum Ceremonies

- • Sprint planning meeting: A meeting at the beginning of a sprint where the sprint is planned. Items from the Product Backlog are selected to be completed in the sprint, based on the priorities set by the Product Owner and each is estimated in hours (1-16).
- Daily scrum: A meeting is held on a daily basis called scrum meeting .Here preference is given to face-to-face communication over written documents when the team consisting of 5 to 9 members is all in the same location. However facilities like voice (Skype is an example), videoconferencing, e-mail etc are used to maintain daily contact if team is working in different locations. This meeting is not a problem solving session but team share the status and potential issues are identified .Here team members make commitments to each other and to the Scrum Master. The duration of this meeting is 15 minutes and every team members answers three questions in standup position.
- What did you do yesterday?
- What will you do today?
- What obstacles are in your way?

While one is giving his scrum others must remain silent and listen to him carefully. Only team members, scrum master and product owner can talk. Daily scrum is a good way for a Scrum Master to track the progress of the Team.

- Sprint review meeting: In this informal Sprint review meeting team presents what it accomplished during the sprint. It typically takes the form of a demo of new features or underlying architecture. The participants in this meeting are customers, management, product owner, other engineers.

IV. FINDINGS AND OBSERVATIONS

1) Customer involvement throughout life cycle is must.
2) If product owner is not available at the time of scrum call, scrum master can hold the scrum and send the scrum note across the product owner.
3) While the scrum is going on, product owner note down everything whatever you say as an answers to three questions using a preformatted scrum document.
4) Every team member must first go through that document on next working day before starting his day
5) While one is giving his scrum others must remain silent and listen to him carefully.
6) While working on the particular requirement if there is need for example to create an API ―XYZ‖ but it was not in the sprint backlog as a requirement for you, you can discuss it with your scrum master and even product owner and add a requirement for this with exact time in product backlog.
7) Every team member must daily update the status (closed,open,inprogress,done) of his/her tasks/requirements in an online system (e.g., VersionOne (http://www.versionone.com/))
8) Communication is more important with product owner and between team members. If any requirement is ambiguous or not clear to you, clear it at that moment only with product owner.
9) You can also fix a meeting with P.O or set up a call with him to clear this ambiguity.

Syed Nisar Bukhari, Ashaq Hussain Dar, and Muneer Ahmad Dar, "Review and analysis of applying agile methodology in software development," *International Journal of Scientific and Technical Advancements*, Volume 2, Issue 4, pp. 187-190, 2016.

10) You can also include your team lead in call/discussion and create the minutes of meeting (MOM) document and share it with the entire team.
11) Every team member must finish his/her work on time (time allocated to each requirement which are on his name) so that there are no backlogs against that particular team member in the next sprint.
12) Synchronization of development team with the testing team is must

V. CONCLUSION

Agile software development stresses quick and rapid iterations, small and frequent releases, and evolving requirements facilitated by direct user involvement in the development process. Agile has made it easier to cope with the changes because of short sprints and constant feedback. The reason why agile methodology have been adopted by many IT industries is that you are delivering the quality product in a scheduled time[Time which is an important factor in software development] and overhead cost in terms of process and management is minimal.

REFERENCES

[1] Cockburn, A., Agile software Development, Pearson Education, Asia, Low price edition.
[2] W Craig Larman, –Applying UML and patterns (An introduction to object oriented analysis and design and the unified process) Second edition
[3] Kim H. Pries, Jon M. Quigley, Scrum project management
[4] Loannis G.Stamelos, Panagiotis fetsos -Agile software development quality assurance.
[5] Pankaj jalote -An integrated approach to software engineering.
[6] Roger S Pressman - Software Engineering: A Practitioner's Approach. Fifth edition McGraw-Hill
[7] http://www.agilemethodology.org/
[8] http://www.codeproject.com/KB/architecture/scrum.aspx.
[9] http://martinfowler.*com/articles/newMethodology.html*

Syed Nisar Bukhari, Ashaq Hussain Dar, and Muneer Ahmad Dar, "Review and analysis of applying agile methodology in software development," *International Journal of Scientific and Technical Advancements*, Volume 2, Issue 4, pp. 187-190, 2016.