International Journal of Engineering Applied Sciences and Technology, 2021
Vol. 5, Issue 12, ISSN No. 2455-2143, Pages 73-78
Published Online April 2021 in IJEAST (http://www.ijeast.com)

AGILE SOFTWARE DEVELOPMENT

Gopalkrishna Waja, Jill Shah, Pankti Nanavati
IT Department, KJSCE
Mumbai, India

Abstract—Agile Software Development plays a quintessential part in modern day software development. The term Agile refers to frequent reassessment and adaptation of plans and techniques and dividing tasks into shorter tasks for efficiency. Agile Software Development differs considerably from Traditional Software Development Methodology. Agile methodology aims to deliver features of a software project in small steps within a short duration of time (i.e., iterations). Hence, it becomes necessary to use agile software development methodology in todays' fast-paced revolutionizing software industry. This paper discusses the important subtopics of Agile Software Development which gathered by reviewing/surveying of research papers. First, is the Agile Planning Life Cycle which consists of various stages such as pre-planning, planning, release planning and product backlog management. In the next section, principles such as Scrum, Extreme Programming, Kanban and Lean are discussed. The last section comprises the impact of Agile principles on software quality

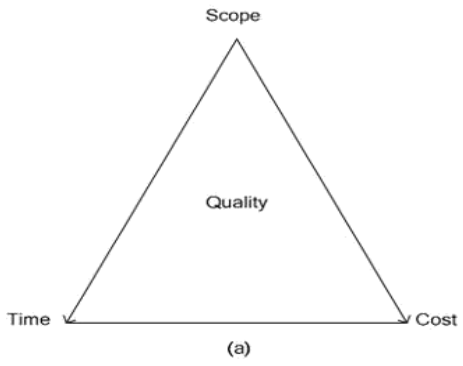
Index Terms—Agile, agile methodology, planning phases, quality impact, software development.

I. INTRODUCTION

In today's world the software industry is a fast-paced industry that requires rapid responses to ever-changing demands. Also, with the advent of globalization organizations, in today's world, are not limited to a single geographic location and have expanded their horizon to a world of infinite possibilities [1]. This increased globalization combined with large scale production and ever-changing demands which require rapid responses pose several challenges like failure of the organisation to cope-up with the increasing, rapidly-changing demands. This failure results because of the inability of traditional software development techniques or processes satisfy these demands. This inability of the traditional process models had created the circumstances which lead to the development of "Agile Process Model" and the software development using Agile Process came to be known as "Agile Software Development. Currently, agile process is the most renowned software development process/strategy [1]. In traditional software-development pro-cess models the requirement analysis and designing of the product is limited/restricted to the initial phase/stage of the process and usually they follow a sequential order. Adding new requirement at a later phase in the development can increase the cost of the project exponentially. Therefore, in traditional models like the waterfall or incremental process model it is almost mandatory for the customer to provide their requirements in the initial phase before the development has started, which means it is non-responsive to changes in the requirements [1].

Agile, on the other-hand, follows a short-cycled iterative-improvement model and incremental-delivery mechanism. This iterative and incremental model allows agile software developers to provide quick deliveries to the customers and improve customer satisfaction [2]. Agile primarily focus on giving highest priority to the customers and their needs, frequent deliveries in short incremental cycles, incorporating the customers as part of the team and collaborating with them, rapid reaction to changes, flexible short-lived plans, simplicity of the solution to a problem, face-to-face meeting and having "self-organising-cross-functional" teams. Also, activities like as planning, design/modelling, construction, and software-testing, rather than being sequential, are continuous activities. All these features and principles make agile process idle for use in fast-paced and sustainable-development facing rapid changes. In the up-coming sections, we have covered topics

Fig. 1. Cost in Traditional and Agile development adapted from [1]



International Journal of Engineering Applied Sciences and Technology, 2021
Vol. 5, Issue 12, ISSN No. 2455-2143, Pages 73-78
Published Online April 2021 in IJEAST (http://www.ijeast.com)

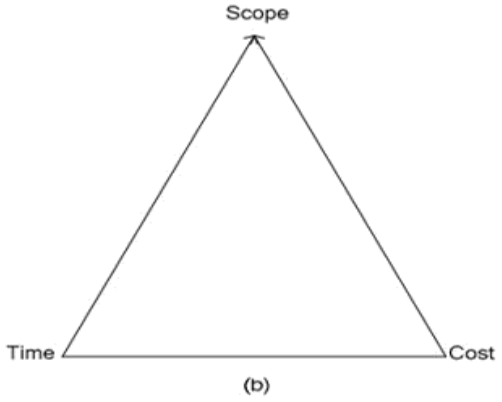


Fig. 1. Traditional Software Project Development (a) and Agile Software Development (b) adapted from [3]

which are relevant to Agile Software development whose content is been gathered by doing through review/survey of research papers. First, we have covered the general phases in Agile software development planning or the Agile planning life-cycle. This planning life-cycle consists of stages/phases like pre-planning, planning, release planning and manage product-backlog. Next, we have surveyed different papers regarding the different Agile Methodologies which are based on the principles mentioned in the Agile Manifesto's. In this section we see the principles and basis of Agile methodologies which include Scrum, Extreme Programming, Kanban and Lean. While reviewing these methodologies, we see all these models, despite their nuanced differences, follow or complement Agile Principles in a manner such that the overall soft-ware development process in a collaborative, rapidly changing environment becomes easy and efficient. Lastly, we see the impact of Agile principles/methodology on software quality. Reviewing the impact on quality is pertinent as the quality of software provides us answers to important questions about the robustness of the software design and its capability to work without faults. It also helps the project manager to track the quality of work done by the team members and allows him to make project's systematic development easy. It is a cost-effective approach and helps in developing a better-quality product in less time. The quality of product produced by a process depends on various factors, out of which we have reviewed the most relevant factors. So, having said that lets move on to the phases of agile planning.

II. AGILE PLANNING

Project planning helps in finding solutions to questions such as what is to be implemented, how much time will be required, costing of the project, what all features are to be added and who needs to be involved. During the earlier stages of the project, there is no clear picture as to what needs to be built, which features are to be included, time duration of completion and the estimates of the project. Planning of the project helps in alleviating such questions of uncertainty, the ultimate objective is to formulate a plan that helps in building a project from its rudimentary state to meet the final customer requirements and objectives.

In the figure 1, scope refers to the features or requirements associated with the project which is to be provided to the product owner. Time refers to the amount of time needed to completed the project model and the cost refers to the amount of money which will be required to implement the project. Quality refers to the satisfaction of the goals, objectives, aims, features of the project. Figure a) depicts the traditional project development while figure b) depicts agile project development [3]. The traditional project development states the project scope first and them calculates the time and costs involved i.e., top-down function decomposition. While in agile project development, scope is flexible and can be changed during the entire project implementation, i.e., it follows bottom-up approach. Traditional software development lifecycle implements the project as a single large model whereas the agile method breaks down the model into small iterations [4].

A. AGILE PLANNING LIFECYCLE

1) *Pre-planning*: The lifecycle starts with a preplanning step which includes identifying project vision, scope, roadmap, gathering as well as prioritizing business and technical requirements, forming the team by selecting members which have the required expertise, time and cost estimation. The product backlog is formulated which consists of all the requirements as well as a list of all desired work on the project. Each item on the product backlog will be assigned some priority by the product owner. A team of experts provide estimates for each feature to be included in the project and thus based on this a first rough time period, cost and scope of the project is determined [3]. Also, number of iterations are chosen by a team of experts.

2) *Planning:* Planning involves deciding the release of iterations, the sequence of iterations, the length of iterations, development speed to meet the requirements of
the customer. Also, depending on the user story scale, project size will be estimated.
3) *Release Planning*: The product backlog which contains all the features and the various tasks which are to be implemented in the project is fragmented to form iterations by the project team and the customers. These iterations are assigned to each release, every release would provide the customer the working model of the project having the features listed down in that particular iteration. On the basis of release planning of the iterations, the dates for further iterations and an approximated date of the final project model can be estimated. The costs involved in the entire

International Journal of Engineering Applied Sciences and Technology, 2021
Vol. 5, Issue 12, ISSN No. 2455-2143, Pages 73-78
Published Online April 2021 in IJEAST (http://www.ijeast.com)

project implementation can also be roughly estimated based on the number of iterations, iteration lengths, overhead costs etc [5].

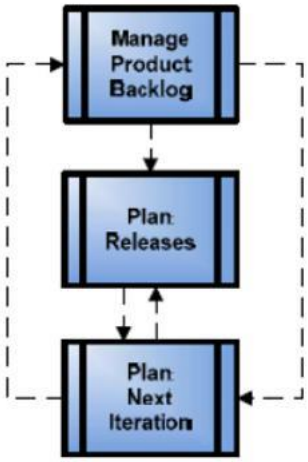


Fig. 3. Planning Lifecycle

4) *Iteration Planning*: The process of planning is iterative. The meeting of the team members of the project is held after each iteration to formulate the scope of the next iteration. In the meeting, the team receives feedback on the current iteration of the project, also they discuss the changes that have to be made in the product backlog including the re-prioritization of features in the product backlog. Once this step has been accomplished, the team goes forth with allocating the various tasks (user stories) discussed in the meeting to the team members. Each iteration is allocated some time; hence the team can remove or add the features discussed in the meeting in order to complete the implementation within the specified duration of time [3].

5) *Manage Product Backlog*: Manage Product Backlog summarizes the changes in the scope, priority and estimates of the project; it can be changed after each iteration as required to enhance the features or remove them [4]. The priority of the item in the product backlog can be changed after each iteration according to the needs and feedback of the customer. Since, after each iteration there are changes made in the project implementation, the changes are also reflected in the time period of project completion, hence it is estimated again. The release planning is also affected after each iteration, since they are dependent on each other.

III. AGILE METHODOLOGIES

Agile methodologies comprise of a set of frameworks that are developed, for the use of organisation and software development team, keeping in mind the Agile Manifesto, published in 2001. Agile process is a multitude of principles or manifestos which can be achieved using of agile methodologies. Despite the differences between various agile methodologies these methodologies find their common foundations in the principles of rapid-adaptability to changes, flexible planning, maximum customer satisfaction, early delivery and continuous improvement through various increments and iterations. From this point on we will be focusing our attention on Scrum, Extreme Programming, Kanban and Lean.

A. *Scrum*

Scrum is an agile, light-weight framework. It provides steps to control and govern the software and product development process. The concept of Scrum was born when Jeff Suther-land and his team, during 1990s, defined it as a process “to energize, focus and add clearness to project developing systems” [6]. Scrum mainly comprises of 3 components: Roles, Ceremonies and Artifacts. The roles in Scrum Process comprise of product/project owner/stakeholder, scrum master and the product/project developmental team. The project-owner is the person who decides the features, budget, priority of the project. The Scrum master is leader or the manager responsible for upholding the values and practices of Scrum. Lastly, the team is comprised of a 4-7 members of cross-functional and self-organising people which actually work on the product. The Ceremonies include the Sprint planning, during which the goals and the sprint-backlog, chosen from the product-backlog, is decided [7]. The product-backlog is a list/collection of the features/functionality required, defined and prioritized, according to their level of importance, by the product-owner/stakeholders in form of user stories. While the sprint-backlog consists of user stories, to be worked upon during the next sprint, chosen and prioritized by the team. Both product and sprint-backlog along with burn out charts are the Artifacts. Other ceremonies include Daily scrum meeting, which is a short stand-up meeting attended by the team members and the stakeholders. Sprint Review, during which the team members and the client, is a survey meeting which involves the demo of the product developed in during the sprint. Sprint Retrospective involves discussion among the co-workers regarding the problems they have faced in the previous sprint and how to avoid them in upcoming sprints [8].

A variation in Scrum is “Scrum without a Scrum master”. According to Zoran Ereiz, Scrum role is subtle, he is responsible for teaching and mentoring the team and removing any complexities and problems. In cases, where there an existence of an experienced team it is practical to have Agile process without a master. Having said that, in every other situation the requirement of a Scrum Master is necessary [9].

International Journal of Engineering Applied Sciences and Technology, 2021
Vol. 5, Issue 12, ISSN No. 2455-2143, Pages 73-78
Published Online April 2021 in IJEAST (http://www.ijeast.com)

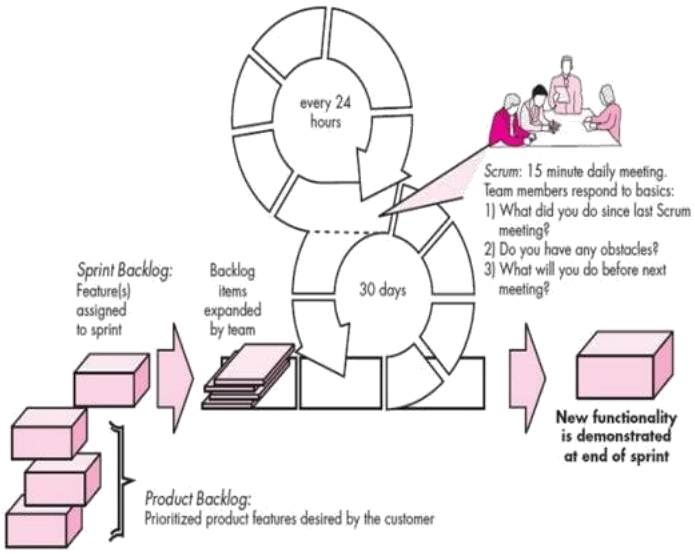


Fig. 4. Scrum process flow adapted from [4]

B. *Extreme Programming*

XP is an agile methodology that helps in management of changes in the code during the software development process. Extreme Programming (XP) mainly focuses proactive and automated testing. Testing forms an integral part of Extreme Programming (XP) and it begins for a very rudimentary stage of the development. XP uses quick and brief development-cycles incorporating incremental design and planning. Another key aspect of XP is it requires high customer involvement at every stage of project/product development. Also, in XP the code is continuously integrated and deployed on a daily basis [10].

It utilizes a light-weighted methodology can be used for any type of software big or small. Concepts of paired-programming, comprehensive reviewing of the code, and lucidity in code form the basis of XP. It also allows adaptability to changes late in the project development allowing quality improvement and better responsiveness of the software to changes made in the requirements by the customer [11]. It also forces on "Open-workspace" and "Small-Release" that is the development takes place in presence of the client onsite and validated results, after testing, are released in small increments [11].

C. *KANBAN*

Kanban work-management system works with the aim to save the team from performing tasks assigned by the administration in a fast passed environment improvement. Kanban allows visualization of in tangible task which are normally difficult to substantiate. This method enables the team to have a better view of the current project by dividing the things-to-do into small chunks and then providing a visualization of them through the Kanban-board. Unlike Scrum which puts a restriction on time, Kanban restricts the amount of works done at a given moment also known as putting a restriction on the measure of work-in-progress (WIP) [8]. Also, in Scrum, fixed roles are assigned to the members while in Kanban this is not the case, in Kanban there are no specialized roles or responsibilities assigned to the members, which means everyone works according to their preference. In Kanban, unlike scrum where the scrum master oversees the project, the administration is done by the product owner. The main concepts in Kanban are: -

Visualization of the workflow through Kanban-board.

Limiting work at a given moment to keep the team focused on a single goal at a moment.
Managing the workflow by extrapolating the time re-quired to prevent wastage of time.
Persistent Feedback to improve the product quality [11].
Continuous and Iterative software/project development.

D. Lean

Lean Agile software development has a greater flexibility as a methodology when compared with Scrum or XP, this is because it has minimal restriction on guidelines, methods, or rules. Lean mainly comprises of 7 values. These values are complemented by other agile-methods since lean pays primary attention on reducing the wastage during the soft-ware process [11]. Another objective of Lean is to try and provide greater value in a limited/short amount of time by re-moving the unnecessary activities during the software/project development-process [11]. Since Lean has minimalistic rules the development process is optimized mainly by the use of specialized tools. Unlike XP, which primarily defines practices mainly customer and developer by trying to ease the tension between them which might arise due to conflicting aims [12], Lean defines values/principles for upper-management of a company/organization and is used for optimization of the entire organization itself, which means it follows a top-down-approach [12].

The 7 main values which form the core of Lean are: First, eliminate of anything which does not provide additional utility to the product to be developed. Second, learn from each and every iteration on the basis of the feedback provided by the user. Third, only make important decision regarding software-development when maximum possible information is available which mostly at later stages of the project. Fourth, make deliveries as quickly as possible. Fifth, provide an environment in which the developers can thrive. Sixth, the primary focus should be the customer. Lastly, view the bigger picture and try to understand the flow of and business and how the product fits in it [11].

International Journal of Engineering Applied Sciences and Technology, 2021
Vol. 5, Issue 12, ISSN No. 2455-2143, Pages 73-78
Published Online April 2021 in IJEAST (http://www.ijeast.com)

IV. IMPACT ON PRODUCT QUALITY

The quality is an important factor for a product to be successful. The idea of quality varies from person to per-son. Hence it is necessary to group the quality in different aspects like perspectives of customer, developer, tester and on specification-based, quality assurance based and manufacturing-based. The main two categories are the product quality and the process quality. The product quality is dependent on the process quality. More the process quality, more the product quality. Product quality means the universal quality of product that makes the user or a customer overjoyed and also results in more customer satisfaction and profit. A product with good quality fulfils all the requirements of the user. On the other hand, the process quality means the method adopted to make the product a quality product. The development team defines the process quality by defining and requiring the specifications and then the code is sent to users, who define the product quality and hence a software product is obtained [13].

When we see the traditional approach, the software development cycle follows the phases which are to be executed in a sequential manner like customer requirements, planning, developing, deployment. The entire process is monitored by project manager. This has a drawback that it may not be able to response to user changes and even if done, they may take more time for the quality of product to be maintained. But when we see the agile approach, being an iterative method, all phases get executed with small cycles called sprints and then user feedbacks are taken after every sprint is completed. This allows the quality of product to be efficient as after every small task or phase, feedback is taken from user and requirements are done. It is easy to change during development itself which helps to create a good product. Other stages in agile like concept, release, inception, production, etc. are not there in any traditional methods. This method being totally focused on customer satisfaction, it is developing a good quality product [13].

The quality of software product while using the agile approach is measured in every phase of the development cycle. In the requirement gathering phase, quality attributes like functionality, reliability, scalability and maintainability are seen depending on agile attributes like user stories, complete-ness and correctness of functionalities and consistency. In the designing phase, quality attributes like efficiency, functionality, maintainability, re-usability, reliability, portability and flexibility are achieved based on the release date of design, number of sprints, story points, etc. The software quality attributes like maintainability and reliability are obtained from the productivity and sprint factor in software implementation. The user stories, review effectiveness and pre delivery dates tell us about the reliability, functionalities, portability and performance of software while testing. The removal of defects and maintaining the efficiency of product after delivery of product is also an important part of agile which has an impact on quality of product to the customer. It shows efficiency, maintainability and reliability of the product. [13] By adopting agile, organizations have increased to customer satisfaction to more than 90 percentage and the defect ratio is less than 20 percentage. Every quality attribute has a positive effect on the quality of the software product [14].

The quality of the software can be measured by counting the defects that are found in production or mentioned by the clients and also by counting the total number of users this affects. We should not let the catastrophic defects go till the production phase. We have to follow the agile metric to find the proper quality of the product. The metric can be of two types i.e., functional and non-functional. Functional metric can be like how many user stories are accepted, what is the ratio of defects in review and in production or the amount of code that was covered while testing, etc. The non-functional metric includes portability, security, usability, reliability, performance etc. This metric is used to detect future problems [13].

As a part of metric, the user stories play a powerful role to define the quality of the product. A user story includes the type of user, the goal and an action that the specific user can perform. User stories should be written in both positive and negative way. The positive user stories help us to identify the functionalities of the system while the negative stories can help to identify the functionality which a user can perform in a wrong manner or an attacker to attack the system. The structure of positive user story involves all positive goals, user and action. The structure of negative user story involves positive or negative user and goal with an action which is negative. Good user stories mean proper definition of user requirements and good requirements means good quality of product [15].

Some of the issues that are faced by many organizations while adopting agile are cultural issues, lack of requirements, communication between the team, software project management. The key barriers like inadequate training for agile adoption, increase in risk of project failure, lack of guidelines, lack of peer support, organizational resistance, etc. are also faced by organizations [14].

V. CONCLUSION

The software industry has numerous changing demands in today's world. As per the agile manifesto, the four values/principles are interactive teams, working software, customer collaboration and responsiveness to changes. Hence, the adoption of agile development has been a crucial part for many organizations. Agile, being an iterative process, changes are done and errors are rectified from iterations to iterations. Quick delivery and customer satisfaction is the major benefit of agile process. So, for this benefit, the agile has to be followed in different phases like inception, construction, etc. as discussed.

These steps play a vital role to in the agile development process. Agile has different frameworks like scrum, Kanban, lean, XP. Each of them has its own benefits. Scrum is always used when the project consists of lots of features and has large goals/milestones. Kanban is used when the project needs small improvements of bug/defect fixes. XP is used when there are small deliveries of software involved and when rapid development is needed. Lean is used when only important functionalities are to be developed. But above all, mostly scrum is used by many organizations. The product quality of the software is always measured in defects. Hence, lesser the defects, more the quality of the product and vice versa. Hence, agile makes huge impact on the quality of the software product by reducing the maximum number of defects while developing the product itself. Agile is therefore used by various organizations both at small scale and at large scale levels. Due to all this, the future of the project management will be the agile development.

VI. REFERENCES

[1] Gonen B. and Sawant D. (2020). Significance of Agile Software Development and SQA Powered by Automation. In 2020 3rd International Conference on Information and Computer Technologies (ICICT) (pp. 7-11).

[2] Jain P., Sharma A., and Ahuja L. (2018). The Impact of Agile Software Development Process on the Quality of Software Product. In 2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO) (pp. 812-815).

[3] Shi Zhong, Chen Liping, and Chen Tianen (2011). Agile planning and development methods. In 2011 3rd International Conference on Computer Research and Development (pp. 488-491).

[4] Khong Loo, Yu Beng Leau, Yip Tham and Fun Tan. (2012). Software Development Life Cycle AGILE vs Traditional Approaches. In 2012 International Proceedings of Computer Science and Information Technology (IPCSIT Volume 37) (pp. 162-167).

[5] Pawar R.P. (2015) A Comparative Study of Agile Software Development Methodology and Traditional Waterfall Model. IOSR Journal of Computer Engineering (pp. 1-8).

[6] Faniran V., Badru A., and Ajayi N. (2017). Adopting Scrum as an Agile approach in distributed software development: A review of literature. In 2017 1st International Conference on Next Generation Computing Applications (NextComp) (pp. 36-40).

[7] Srivastava A., Bhardwaj S., and Saraswat S. (2017). SCRUM model for agile methodology. In 2017 International Conference on Computing, Communication and Automation (ICCCA) (pp. 864-869).

[8] Saleh S., Huq S., and Rahman M. (2019). Comparative Study within Scrum, Kanban, XP Focused on Their Practices. In 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE) (pp. 1-6).

[9] Ereiz Z., and Musiˇc´ D. (2019). Scrum Without a Scrum Master. In 2019 IEEE International Conference on Computer Science and Educational Informatization (CSEI) (pp. 325-328).

[10] Sharma P., and Hasteer N. (2016). Analysis of linear sequential and extreme programming development methodology for a gaming application. In 2016 International Conference on Communication and Signal Processing (ICCSP) (pp. 1916-1920).

[11] Herdika H., and Budiardjo E. (2020). Variability and Commonality Requirement Specification on Agile Software Development: Scrum, XP, Lean, and Kanban. In 2020 3rd International Conference on Computer and Informatics Engineering (IC2IE) (pp. 323-329).

[12] Wang X. (2011). The Combination of Agile and Lean in Software Development: An Experience Report Analysis. In 2011 Agile Conference (pp. 1-9).

[13] Jain P., Sharma A., and Ahuja L. (2018). The Impact of Agile Software Development Process on the Quality of Software Product. In 2018 7th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO) (pp. 812-815).

[14] Ruk S., Khan M., Khan S., and Zia S. (2019). A survey on Adopting Agile Software Development: Issues Its impact on Software Quality. In 2019 IEEE 6th International Conference on Engineering Technologies and Applied Sciences (ICETAS) (pp. 1-5).

[15] Chopade M., and Dhavase N. (2017). Agile software development: Positive and negative user stories. In 2017 2nd International Conference for Convergence in Technology (I2CT) (pp. 297-299).