

# Review of an Agile Software Development Methodology with SCRUM & Extreme Programming

Purvi Sankhe  
IT Dept., TCET,  
Mumbai, Maharashtra.  
India

purvi.sankhe@tcetmumbai.in

Shruti Mathur  
IT Dept., TCET,  
Mumbai, Maharashtra.  
India

Shruti.mathur@tcetmumbai.in

Tasneem Bano Rehman  
SAGE University,  
Bhopal, Madhya Pradesh.  
India

Tasneem.bano@gmail.com

Mukesh Dixit  
SAGE University,  
Bhopal, Madhya Pradesh.  
India

Mukesh.d@sageuniversity.edu.in

**Abstract** – Software development methodologies offer a methodical way to working on projects in accordance with client expectations. In the software development process, the customer is seen as the main entity. In order to build valuable software quickly and within a given budget, firms are now seeking to implement customer-centric techniques. Customer satisfaction plays a crucial part in the development and deployment of software products. One software development process that helps to best fulfil changing customer requirements is agile software development. In the following part, a table of comparisons between the various agile modelling methodologies with regard to requirements and the role of the customer follows a discussion of agile methodology. The Scrum & Extreme Programming framework is presented, along with how different firms use the approach for projects of various sizes.

**Keywords-** Software development approaches, agile methodology, agile modeling, scrum.

## I. INTRODUCTION

We are all aware of how vital software has become to our daily lives. Software has been created over time using a variety of platforms, including mainframe computers, personal computers, tiny embedded systems, ubiquitous computing, even automobiles and aeroplanes.

However, if we look back at the early days of software development, it had numerous catastrophes or failures that led to fruitless projects, wasting all valuable resources like time, money, and effort. Many developers and academics attempted to produce products of higher quality using new tools and technologies, better models and techniques, better processes, or by developing new technologies that were suitable for development. However, when they were faced with the need to work on large-scale projects and multiple environments, they were forced to use a systematic approach of engineering systems and methods.

For the proper & high-quality development of software projects, software development methodologies are fundamental. This is necessary in order to provide direction for managing and coordinating the task lists in order to produce better software and achieve the project's goals. However, the most appropriate techniques are chosen depending on the available resources and other factors, which are not always ideal for that circumstance and may result in project failure.

For the creation of software products, two main methodologies are used: the classic waterfall approach to

software development and the agile software development approach. Since the early 1990s, agile software development methodologies have evolved and improved.

For circumstances of product development where the client is unsure of all the needs at the beginning of the process, this technique, as shown in Fig. 1 [1], iterative & incremental strategy of development life cycle. [2].

Agile techniques, which are now used by several firms throughout the globe, are founded on a set of principles created by diverse specialists. [3]. Agile Manifesto was developed based on 12 predefined agile principles and values, and this led to the establishment of agile associations that assisted teams and businesses in putting the agile manifesto's concepts into practice. [4]

One of the features of this technique, constant customer interaction in the software development process and rapid, incremental software delivery, which includes simple adaptability to changes, is the mechanism that draws developers to it for the development of software products.

By breaking down the main application system or project into smaller increments or stories, agile approach satisfies consumers while readily adjusting to their modifications or changes caused by any situation [5].

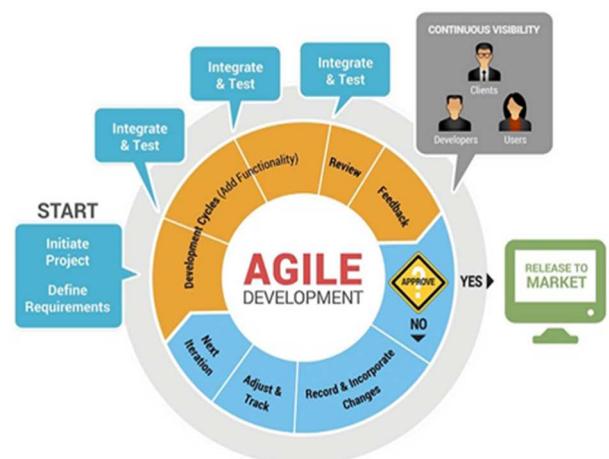


Fig.1. Agile System Development Structure

## II. LITERATURE REVIEW

In the paper [6], All the essential points and details about the software development methodologies are reviewed in this paper as well as the advantages and disadvantages of each model is also discussed. So, the key findings in this paper is, Many companies and software developers want to increase the progress of software development by investigating the advanced techniques and methods to minimize the complicity of the development process. They are having difficulties when choosing an appropriate software development methodology.

Research gap identified is, it has been observed that There is no any methodology to fulfil all the requirements and satisfy developers & customers to gain a perfect final project. After considering all the parameters, the best and proper methodology should select with the help of experienced team members and project managers. Instead of selecting one methodology, a combination of software methodologies could choose for the development process to gain a better outcome.

The author [7] intends to set the phase for the formalization of a product advancement system devoted to development-oriented IT anticipates. Programming improvement procedures are thought about by highlighting qualities of Software development methodologies characteristics, strengths & weaknesses. Key findings are, choosing a software development methodology need to check profile of project owner, project details, complexity of project, developers' technical skill and budget .in some cases there is no other methodology is suitable for profile of project. The experienced team and project, manager together gives a best methodology. A new methodology is needed to make innovative software project [7].

Software development methodology uses two methods: Heavyweight and lightweight. Heavyweight requirement cannot be changed and software requires detailed planning. Lightweight is easy to change .it provides delivery of working code, self-organizing teams and adaptive planning. Choosing a software development methodology need to check profile of project owner, project details, complexity of project, developers' technical skill and budget .in some cases there is no other methodology is suitable for profile. of project. The experienced team and project, manager together gives a best methodology. A new methodology is needed to make innovative software project.

### III. AGILE MODELLING TECHNIQUES

By using any of the agile process models, it is feasible to take use of the appealing features of agile methodology, such as speedy product delivery and ease of adaptation to changes [3]. These include Scrum, Lean Software Development, Adaptive Software Development, Kanban, Dynamic System Development Methods, Feature Driven Development, and Extreme Programming. Each of these types has its own advantages and disadvantages. The use of the process model is based on the scenario and problem statement for which a product-based solution must be built [8].

A comparative analysis regarding how different agile process approaches manage requirements was provided by Baruah in [9]. In terms of requirements and administration, Table I compares and contrasts the eight agile software development methodologies[9]

TABLE I. DIFFERENT AGILE MODEL'S COMPARISON BASED ON ROLE OF CUSTOMER & REQUIREMENTS

Sr. No.	Agile Model	Requirements	Role Of Customer
1	Extreme Programming (XP)	Both cards and chats are used to write user stories. For implementation, there must be precisely defined and comprehensive requirements. Written cards are optional for implementation. As soon as implementation is complete, written stories are destroyed.	Customers should participate in the requirements estimating and requirements prioritization processes.
2	Scrum	Communication between the development team and consumers' representatives is how requirements are gathered.	The software's owner is crucial and essential in determining requirements.
3	Feature Driven Development	Project is divided into features. UML diagrams are used to represent needs & requirements. Customers and the development team create a method for modeling requirements according to a domain. Requirements are assessed by peers after modeling is completed.	Customer Centric, continuous communication with customer.
4	Lean Software Development	Requirements are collected using Just-In-Time policy. On cards, requirements are listed. Developers who estimate the amount of time needed for each card use sample screens.	Continuous communication with customer,
5	Adaptive Software Development	In the speculation phase of an ASD, the customer's mission statement, requirements, and constraints are gathered.	Continuous communication with customer,
6	Kanban	User stories are used to define each sprint main goal. Each sprint handles a single user story. Each story is divided into server-side and client-side task.	Not Specified
7	Dynamic System Development Methods	The feasibility phase involves gathering and prioritizing requirements.	Not Specified
8	Agile Unified Process	Requirement phase consists of the following activities: recognizing the participants, Understanding the issue, defining the user interface, and establishing a foundation for estimating User stories are used in	Not Specified

	the construction phase. Requirements are presented as Business Use Case Model.	
--	---	--

#### IV. DIFFERENCE BETWEEN AGILE SOFTWARE DEVELOPMENT AND TRADITIONAL SOFTWARE DEVELOPMENT APPROACH

Traditional approaches follow sequence steps like Requirement elicitation, planning, modeling, construction & implementation, testing and deployment. The client needs are first meticulously outlined in a document called a software requirement specification. After that, the software's general structural design is completed, which will give users a visual representation of the software development process. After implementation is complete, many tests will be run before the software is put into the hands of the customer. The fundamental methodology involves carefully visualizing the finished product before the process begins and following that planned and imagined procedure until the finished product is ready. It limits the usefulness of the product and forbids modifications.

This method only works for stable and less complex types of projects. It involves planning all activities by taking into account all aspects before beginning the real process of software development and continuing with the same planning until the end of the software development process. For a complex project with shifting requirements, nothing can be planned or followed through to completion. The process of development could be destroyed.

Traditionally, personnel who work on projects are treated as production factors together with those who design products. There is no latitude given when assigning responsibilities to project workers.

Customer feedback and testing activities are typically maintained during the last phase of a project's lifespan. The timely delivery of the finished product and staying under the budgeted amount are the two major criteria that determine the success of a predictive heavyweight project.

Agile software development uses a nonlinear process to create software, and all types of people are given equal weight (customers, developers, stakeholders as well as end users).

Agile technique uses a self-organizing development team. Agile teams adapt changes in the following increment or iteration of software products or applications in a very favorable way. It places more weight on actual progress than meticulous planning. The technical team is given authority by management to make technical decisions without their consent regarding which requirements from the prioritized list of requirements should be implemented first. From the beginning of the software product development process until the client receives the finished product, the customer is involved.

#### V. SCRUM METHODOLOGY

The Scrum framework is built on systematic and structured communication between the Scrum Team, the Product

Owner, and the Scrum Master [10]. Scrum basically evolved

from activities that take place during a rugby match.

Scrum is an agile technique that places an emphasis on producing the most business value in the shortest amount of time. Software features are quickly developed, taking between 15 and 30 days to complete. so that anyone can view a working version early on in the development process and choose to use that version or improve it in subsequent revisions. Sprint refers to the individual iteration phase. Each sprint results in the creation of a deliverable product. Priorities are set by the company. Scrum teams self-manage to decide how to deliver the most important features in the most effective way.

With regard to visibility, process effectiveness, quick development, clearly defined responsibilities (as seen in Fig. 2), cooperation significance, and comprehension, Scrum has a strong qualitative approach [11]. Following is the discussion about Roles & responsibilities of Scrum as shown in Fig 2.

##### A. Product Backlog

It is a prioritized list of needs that offers the customer business values. Items may at any time be added. The Product Owner should keep track of it, prioritise it, and *share it with the scrum team*.

##### B. Scrum team

The average scrum team has 6 to 10 individuals, all of whom have specialized skill sets and enough size to comfortably complete the responsibilities. Developers, testers, database specialists, support workers, etc. are among the members of this group, which also includes the product owner and a scrum master. To carry out and create the planned features, all of these team members closely collaborate over a repetitive and set period of time [12].

##### C. Scrum meetings

These are short (15 minutes) meetings held daily by the scrum team. Meeting focuses on mainly three questions:



Fig. 2. Roles & Responsibilities of scrum

- What have you done since the last team meetings?
- What difficulty are you facing?
- What do you plan to achieve by the next team meeting?

##### D. Product owner

A key stakeholder among all those involved in the software development process is the product owner, who is either the owner or the direct user of the application system to be produced.

#### E. Scrum master

It is a team leader who oversees the Processes and serves as the scrum team facilitator. The scrum master serves as a liaison between the Product Owner and other team members and ensures that the scrum team formed is productive and forward-thinking.

#### F. Demos

Delivering software increment to the customer.

#### G. Burn down chart

A burn down graphic compares the estimated effort of the scrum jobs to the actual effort. Daily tasks are tracked to determine whether the stories are moving closer to completing the committed story points or not. It is used to track sprints.

Scrum is most typically used to manage the iterative and incremental process of developing complex software and products. Comparing Scrum to the old "waterfall" technique, productivity is noticeably higher and the time to benefits is shorter. Scrum procedures let organizations generate products that match changing business objectives and easily adapt to needs that change quickly. Scrum is a set of techniques that include sprints, sprint reviews, sprint planning, sprint retrospectives, daily scrums, a scrum master, a development team, a product owner, and a product backlog, as shown in Fig. 3.

**Sprint:** The scrum team works for a brief period of time to complete a specific job that has been determined [13].

**Sprint Planning:** Sprint planning, which involves the scrum master talking with the team on how much work has to be done in a sprint while taking the resources, time, and budget into consideration, is necessary for a sprint to be finished on schedule [13].

**Sprint Review:** After the sprint is over, the scrum master calls a sprint review meeting with the scrum team and stakeholders to go through the sprint's results, what was and wasn't completed, and the project forecast [13].

**Sprint Retrospective:** Final action taken both after the sprint review and before the next sprint planning.

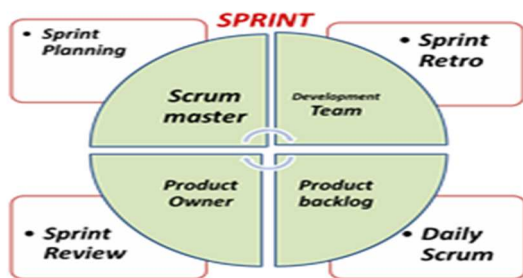


Fig. 3. Scrum Practices

The scrum team discusses the lessons learned and difficulties

encountered during the previous sprint during this session so that the processes can be adjusted as needed for the following sprint.

Adopting Scrum methodology for development of software provides following benefits to the organization [13]:

- Capable of quickly adjusting to changes
- Offers deliverables of higher caliber as a result.
- Provides better estimates while taking the least amount of time to produce them. [14].
- Meeting the deadline while producing a top-notch product.

## VI. EXTREME PROGRAMMING (XP)

One of the agile process models used for software product development is extreme programming. It is a fairly well-known methodology for delivering superior quality products while also delighting customers, and it was created in typically small teams with team members who value collaboration [15]. The foundation of this concept is a dynamic agile process model that executes tasks while incorporating ongoing conversation and new features and ideas. Instead of the owner or the manager of the firm, it is the collective obligation of all the engineers on the team to deliver the product [15].

The authoring and testing of code forms the basis of the entire XP concept. Object-oriented thinking is the chosen development methodology for XP [16]. Planning, Design, Coding, and Testing are the four framework activities that are defined. As shown in Fig. 4, it illustrates the actions and results of each phase

#### A. PLANNING:

It starts with the creation of a collection of stories (also known as user stories), each of which is written by the customer and placed on an index card. Next, the customer determines the priority of each story, after which the XP agile team evaluates each one and assigns a cost, groups the stories according to the committed release of an increment, and uses the project velocity calculation to determine the timing of subsequent delivery dates for other increments.



Fig. 4. Activities & Phases of Extreme Programming

#### B. DESIGN:

CRC cards were created on the "keep it simple" principle. For complex design issues, it provides "spike solutions"—a design prototype. It takes place both prior to and following the start of coding and promotes "refactoring," an iterative



improvement of the internal program design.

### C. CODING:

This phase celebrates the creation of a set of unit tests for each of the stories before any actual coding takes place. Promotes "pair programming," which involves two programmers working together at a single terminal. One is known as the "driver," who develops code, and the other is the "observer," who reviews. Both programmers frequently move between positions. It offers superior mechanisms for real-time quality control and problem solutions.

### D. TESTING:

By putting the Unit Testing framework in place, testing becomes automatic. Regression testing is thus encouraged. Daily integration and validation testing is possible. Acceptance tests, also known as customer tests, are derived from user stories and are provided by the customer and carried out to evaluate the customer's visible functionality.

In a team that uses the XP agile process, everyone collaborates throughout every stage of the project. Coder insists on developing straightforward code, which saves time and effort and adds value to the final result. Team members frequently release software updates, solicit user feedback, and make product improvements in response to these inputs and any new requirements. Coders are constantly prepared to adapt to changes and empirically assess their own work without offering justifications.

## VII. RESULT ANALYSIS

Extreme Programming (XP), Feature-Driven Development, and Agile Unified Process (FDD). The Scaled Agile Framework (SAFe) and hybrid (HYB) approaches, which have grown in favour as agile has been utilised in bigger projects and embraced corporate-wide, were highlighted in later findings [17]. Figure 5 depicts the results of a survey on the use of the software development technique by the number of organizations and project size.

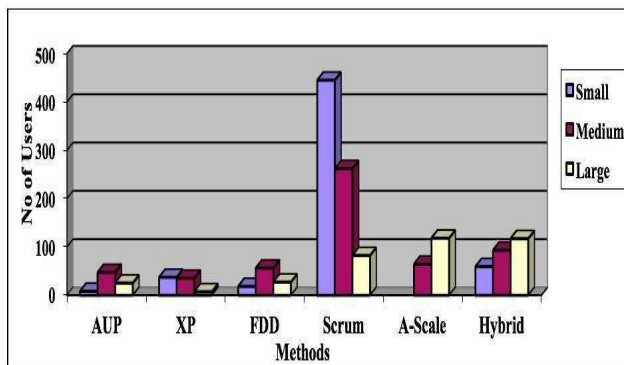


Fig. 5. Methodology Usage by No. of Organizations by Size of Project

#### Note

Small Project (567): project that delivers a product can be developed by a single agile team.

Medium Projects (581): agile project that uses 2 to 5 teams at same locations to develop products.

Large Projects (352): large project that uses 5 or more teams

often at different locations to develop products.

Agile Unified Process, Extreme Programming, Feature-Driven Development, Scrum and its offshoots, Agile at scale methodologies, and Hybrid, which is a blend of methodologies, are the methodologies taken into consideration in this study. According to the infographic, the majority of firms utilize the scrum approach for small, medium, and big projects, while only a tiny number of organizations use other methodologies. Some organizations also use the hybrid methodology, which combines more than one methodology.

## VIII. CONCLUSION

The two frameworks of the agile software development methodology—Scrum and Extreme Programming—are the main topics of this paper's discussion. By concentrating more on people and their interactions than on tools and processes, the agile methodology is successful in producing better quality functional software in a short amount of time, ultimately delighting consumers. Customers have a welcome opportunity to participate in the entire software development process throughout the project thanks to the agile methodology. Prioritizing features puts the user experience and commercial value of the software product in the forefront.

Employees generate significantly better value when self-organized, and Scrum has faith that they are self-motivated and seek to take on more responsibilities. It is a loosely managed approach that fully sympathizes with regular updates of the progress; as a result, project development steps are apparent in this method.

Although Scrum and XP are practically identical, there are minor variations between the two methodologies. In a team that uses the XP agile process, everyone collaborates throughout every stage of the project. Coder insists on developing straightforward code, which saves time and effort and adds value to the final result. Team members frequently release software updates, solicit user feedback, and make product improvements in response to these inputs and any new requirements. Coders are constantly prepared to adapt to changes and empirically assess their own work without offering justifications.

A product's development utilizing agile methods can be influenced by a number of factors, either directly or indirectly. The development team and the customer are both satisfied as a result of using this methodology due to its favorable effects on quality and productivity. From the review of few papers, it has been observed that many software developers and researchers are devoted to making appropriate Improvements on existing development models according to the characteristics of actual software projects for certain application fields so that any suitable methodology or combination of methodologies can be used for the development of projects a per the requirements of customers.

From the review of few papers, it has been observed that many software developers and researchers are devoted to making appropriate improvements on existing development models according to the characteristics of actual software projects for certain application fields so that any suitable methodology or

combination of methodologies can be used for the development of projects a per the requirements of customers.

#### REFERENCES

- [1] Isaac Sacolick, Agile Methodology & Modern software development [online], (6 April 2022), Available: <https://www.infoworld.com/article/3237508/what-is-agile-methodology-modern-software-development-explained.html>
- [2] K. Singh, "Agile Methodology for Product Development: A Conceptual Study", International Journal of Recent Technology and Engineering (IJRTE), Vol.10 Issue-1 ISSN: 2277-3878, May 2021.
- [3] G. Kumar, P.K. Bhatia, "Impact of Agile methodology on Software Development Process" International Journal of Computer Technology & Electronics Engineering (IJCTEE), Vol.2, Issue 4, August 2012.
- [4] Scheerer, A., "Coordination in Large-Scale Agile Software Development", Springer. 2017.
- [5] Dynamic System Development Method Consortium, "Dsdm Tour" [online]. Available: <Http://Www.Dsdm.Org>, Accessed 20/2/2022.
- [6] Ms. I.G.U., D. Rathnayaka, Dr. BTGS Kumara, "A Review of Software Development Methodologies in Software Engineering", IJARIE-ISSN, Vol-6 Issue-4, pp (O)-2395-4396 ,2020.
- [7] P.Saranya, V. Monica, J. Priyadharshini , "Comparative Study of Software Development Methodologies", International Research Journal of Engineering and Technology (IRJET), Vol.04 Issue: 05, e-Issn: 2395 -0056 May -2017.
- [8] Agile study [online]. Available: <https://Www.Infoq.Com/Articles/Reifer-Agile-Study-2017>. Accessed 16/4/2022.
- [9] Baruah, N. (2015) Requirement Management In Agile Software Environment. Pr Ocedia Computer Science, 62,81-83[online].Available: <https://Doi.Org/10.1016/J.Procs.2015.08.414>
- [10] A. Ali, M. Rehman, M. Anjum, "Framework for Applicability of Agile Scrum Methodology: A Perspective of Software Industry", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 8, No. 9, 2017.
- [11] N. Bohrweg, Applicability of Agile Scrum within IT infrastructure projects, 2015.
- [12] J. Cho, "Issues and Challenges of Agile Software Development with Scrum ", Colorado State University-Pueblo, Joey.Cho@Colostate-Pueblo.Edu, Issues in Information Systems, Vol Ix, No. 2, 2008.
- [13] Help with Agile & Scrum [online]. Available:<https://www.cprime.com/resources/what-is-agile-what-is-scrum/> Accessed on July 2022.
- [14] Software Testing Help[online]. Available:<https://www.softwaretestinghelp.com/agile-scrum-methodology-for-development-and-testing> Accessed 22/5/2022.
- [15] A. Shrivastava, I. Jaggi, N. Katoch , D. Gupta and S. Gupta, "A Systematic Review on Extreme Programming", IRMAS 2021 Journal of Physics: Conference Series, doi:10.1088/1742-6596/1969/1/012046
- [16] R. Fojtik, "Extreme Programming in development of specific software", Procedia Computer Science 3 1464–1468, ELSEVIER, 2011.
- [17] S. Stavru, "A critical examination of recent industrial surveys on agile method usage", Journal of Systems and Software, pp. 94, 87-97, 2014.