EXTREME PROGRAMMING VS SCRUM: A COMPARISON OF AGILE MODELS

Asma Akhtar, Birra Bakhtawar, Samia Akhtar

Department of Computer Science, Virtual University of Pakistan, Lahore, Pakistan

asmaakhtarjanjua@gmail.com, birra.bakhtawar@gmail.com, samiaakhtar9898@gmail.com

ABSTRACT

For past couple of years agile software methods have been quite popular among the researchers. Agile models are known as light weight in contrast with conventional software development methodologies, due to their casual, versatile, and adaptable style. Agile frameworks became heartily accepted by the software society in view of their concentration towards timely software conveyance, product quality and user satisfaction. For the fulfillment of requirements and needs of different software projects multiple agile frameworks are present to choose from. Out of these models Extreme Programming and Scrum are the most recognizable and generally utilized frameworks. This research contributes by investigating these two frameworks thoroughly. This paper conducts a comprehensive comparison between Scrum and Extreme programming to track down their commonalities, dissimilarities and investigate those attributes which complement each other.

Keywords*: Extreme Programming, XP, Scrum, XP vs Scrum*

1. INTRODUCTION

Software development methodologies are utilized for creating all kinds of projects whether small and easy or big and complex. These methodologies minimize the risks of project failure. They are very helpful in developing software in sectors like academic and industries [1]. There are different software development methods present like waterfall, spiral and agile [2][3][4][5]. Due to the benefits and the attributes presented by agile methodologies, agile frameworks have taken the place of traditional models in the software community [6][7][8][9]. These attributes tackle the necessities of present-day development of software. The scholars, analysts and software practitioners blended the best practices and features to conquer the

downfalls of traditional models [10]. To produce good quality product, agile frameworks deliver the software considerably fast [11]. Agile frameworks are iterative and incremental in nature hence user's requirements are fulfilled by constantly delivering fragmentary and partly done software [12][13][14]. Any change in client's requirements is also easily handled during any phase of development. These models have proven effective for handling the changing business environment [15][16][17][18]. Agile methodologies have been getting admired for quite some time now as they are simple, adaptable, and perfectly suitable for today's requirements of software development. They have proved to be quite resourceful in various fields like business and education system [18]. Feature driven development [19], Adaptive software development, Dynamic system development method, Lean software development, Scrum and Extreme programming [2] are the examples of agile models. These different agile methods also include a variety of techniques like test-driven development [20].

The most utilized agile frameworks are Scrum and Extreme Programming particularly for limited scope projects. Extreme programming has 6 phases in the development process. 12 best practices of Software engineering are used during these phases. Extreme programming utilizes these practices to achieve better quality software [21]. Change in requirements can be easily managed due to its adaptable, lightweight and iterative nature even in the last stages of development [22][13]. A great interest has been observed as well in customizing XP as it is simple and flexible [23]. An example of such customization is Scaled Extreme Programming [24] XP also assists the developers to consistently recognize and get work done on the software artifacts with higher priorities [15]. Scrum is an iterative project management model which gives a straightforward 'inspect and adapt' system [25]. In this model the delivery of the software is in increments known as Sprints. The scrum model is very useful for projects that are complex especially the ones with requirements and essential technology that is still immature [26] and out of all the agile models, Scrum is the most frequently used [27][5][28][29]. It is a very popular model and adopted all around the world. Scrum was also found to be widely used at Microsoft [30]. Like XP, Scrum can also be customized [10]. Many hybrid models of scrum have been developed within the agile family [6]. These two agile frameworks, XP, and Scrum have quite a few similarities and differences. This study is directed to investigate and analyze them thoroughly. This examination gives a profound understanding of these two approaches that will significantly be quite useful for developers and analysts.

A systematic research process is performed in this paper for the comparison study of XP and Scrum. To perform a successful systematic research process, an appropriate research method is required which can assist in accomplishing all the research objectives. Multiple

studies have been conducted which provide a guideline for the systematic research process [10][23][31]. A systematic search strategy was formed based on these studies which was adopted in this paper. Generally, Systematic research process consists of 3 fundamental steps; plan, conduct and document. The steps which are included in our research methodology are as follows; Research questions are defined, keywords are stated to form the query string, research space is specified to collect data, inclusion and exclusion criteria for articles and studies is defined, literature is extracted based on this criteria, quality of the study is assessed, required data is synthesized and lastly the results are documented [32].

The remaining paper is divided into following sections: Section 2 explains Scrum, Section 3 describes Extreme programming extensively and Section 4 presents the comparison of both models in detail.

2. THEORETICAL FRAMEWORK

In 1995 Sutherland and Ken Schwaber came up with an iteration-based software development framework. Later, it began to be known as Scrum when Ken Schwaber and Mike Beedle released a book called "Agile Development Software with Scrum" in year 2001. Scrum tackles loopholes of previous frameworks effectively and its every release is as per the changing customer demands. The overall process is completely transparent, well-inspected and easily adaptable. This strong check and balance system assures quick elimination of anomalies and timely results [33]. The product is released in phases called Sprints. The duration of sprints is 30 days or less. This model consists of 3 roles, 4 ceremonies and 3 artifacts.

2.1 *Scrum Phases*

There is Pregame, Game and Post game [34].

Pregame: Pregame defines the tentative vision based on customer expectations and market demand. This vision is continuously modified throughout the process later. The main purpose in this initial phase is to create a Product Backlog which has a list of functional and non-functional requirements [33]. Other form of data which needs to be reported is time and cost estimation plus the number of releases and the expected delivery date. For all these things, you need people, tools and funds [35]. So, the backlog contains this information as well. It has the names of members, the required development tools and the minimum number of finances needed to carry out each step of the plan. Fig. 1. Shows framework of scrum.

Game: Game happens in Sprints. Sprint is a one to four weeks period-based process in which you create, wrap, check and modify the product in question [34].

Post-game: In this phase, the final product is released, and the integration test is performed. Of course, this is after ascertaining that all the established requirements have been met. The user manual and training materials are also prepared to facilitate the user.

2.2 Sprint Cycle

The product backlog sets guidelines for the software [34]. The Scrum Master guides the team in its development phase and before the actual delivery of the product. The following steps are taken in the overall process

Sprint Planning and Daily Scrum: First, the scrum master and the product owner decide on the requirements. The priority tasks are filtered and guidelines for understanding the user needs are established. Next, the mission is handed to the team members after clearly communicating the requirements [36]. The self-driven, self-organizing and highly trained professionals, who despite having zero involvement from the outside, are expected to answer the following questions on almost a routinely basis [37].

- What is the progress so far?
- Are there any deviations from the established sprint goals?
- Are there any nuisances in the process?

Of course, the goal is to keep a tight check on the over-all performance without disrupting anybody's personal space for quick and efficient results.

3. LITERATURE REVIEW

Each sprint is developed and tested as per the guidelines and established priority, as communicated by the master. Now the sprints are reviewed by the boss and the owner. A detailed inspection is carried out to check whether the product is in line with the customer requirements. Next, a meeting is held where thoughts and opinions between the two parties are shared in detail. Based on this discussion, the next course of action is decided. The team members might have to completely re-think and design the sprint if the owner deems it necessary. For a 1-month sprint, the maximum duration of a sprint review is 4 hours [38]. Having established the basic plan, the next meeting is all about the improvements that could be made. The two parties check for any loopholes and decide on the remedial measures. Scrum is the team established for product development. It consists of one, the product owner, two, the scrum master and three, the team members [37]. The product owner is the master of all trades.

He creates the project schedules, manages finances, negotiates with the shareholders, and communicates their needs to the team. He has a fairly good idea of market's strengths and

weaknesses and understands his team [39]. Therefore, it is imperative that he has strong grip on both the management and engineering fields. He stays in touch with the changing market dynamics and keeps his team informed about all the rising opportunities that can be beneficial for the project [40]. The scrum master is the team's master. He establishes some ground rules and makes sure everybody follows them [41]. Also, he keeps in check the over-all working environment, protects the team from any outside intervention and takes care of the interests of his members. He holds scrum meetings daily where work-related issues are discussed, and practical solutions are offered [42].

The team has generally 3 to 9 members. Although the team size depends on the area of operations, but most studies suggest that a team size of 7 +/- 2 has more success rate [38]. Each member is highly qualified and trained for the task assigned [43]. While each member has the freedom to choose his own task, the team must ensure smooth execution of the process from initial phase to its final practical implementation in the market [44].

3.2. Research Model

In our research, we have developed the following model for the better understanding of the concept used in the research.

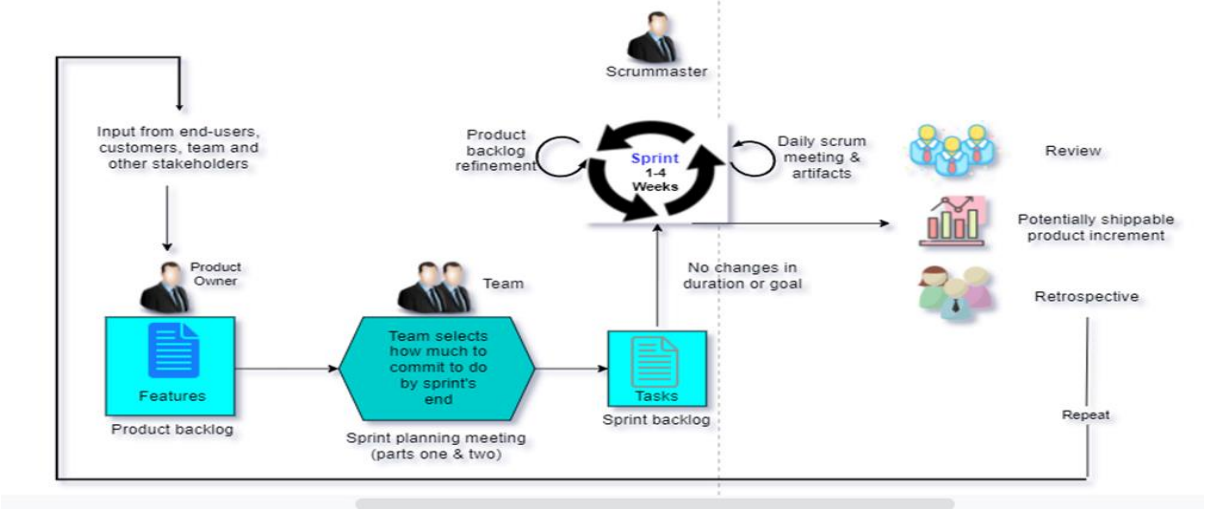

Figure 1: Scrum Framework

3.3. Variables Identification

Extreme Programming:

Extreme programming, an agile model, was invented by Kent Beck in the year of 1996. He then introduced his work on Extreme programming in a much sophisticated and advanced form in the shape of a book known as "Extreme Programming Explained". It is quite simple, uncomplicated, and more adaptable methodology of development with the capacity to oversee unclear, ambiguous, or quickly varying requirements [26]. This model is well suited for teams of small or medium size [45]. Extreme Programming is a set of values, principles and practices which are implemented in an orderly fashion [46]. Such practices that proved useful in creating a high-quality software were taken to extreme hence the name Extreme programming. This agile model greatly emphasizes on user satisfaction. To tackle the defects and errors at early stages, frequent reports and partially done software releases are necessary. Lesser number of defects bring down the development cost and expenses and produce high quality product at such a low cost. Extreme programming overview is given in Fig. 2.

XP Phases:

There are six phases in Extreme Programming development process: Exploration, Planning, Iteration to release phase, Production, Maintenance and Death phase.

Exploration Phase:

This stage is the first step of Extreme Programming life cycle that manages product's architectural development and requirements of the product. This stage describes client's requirements, design and architecture, tools and software used. To schedule the release, a meetup is organized between client and the developer. The story cards are used by clients to compose user stories to present software requirements. The user story cards consist of story's priority, a brief name and one to two paragraphs of non-technical text [47]. The user stories are supposed to be comprehensive and very much in detail to assist the developer to better grasp and comprehend system requirements and help furthermore with estimations. The time that is needed to execute a story is known as time estimation [48]. In case a story takes more time to execute then the customer can break that story down into rather small fragments of stories. During architectural spike metaphors are formed for the purpose of modeling architecture [49]. A Metaphor does not fulfill all the criteria of a whole architecture rather it is just a structure containing objects and their interfaces. Exploration stage has a duration of several weeks to several months.

Planning Phase:

Planning stage begins right after the exploration stage has completed and this second phase looks for the responses of two inquiries fundamentally:

Which parts can be formed within the deadline that consist of a business value?

What would be the strategy for the upcoming iteration?

Planning stage requires merely one or two days to finish if the exploration stage went well. Utilizing the user stories tasks are drawn and composed on task cards during Planning. In Extreme programming planning is knowns as "Planning game" which is additionally carried out in two sections i.e., Iteration planning and Release planning [50].

Release Planning:

The main target of release planning is to discover the attributes required by the system and delivery plan of those attributes. In the meeting held for Release planning both the client and the developer take part. Release Planning is further sub divided into three stages: "exploration phase, commitment phase and steering phase" [47]. Story cards are written by the clients to find out the much-needed system attributes [51]. Then all these required attributes are arranged by their significance and for the latest release a smaller group of story cards is chosen. This is a continuous procedure that can be modified and changed by adding, erasing, combining, or breaking up a few stories [52].

Iteration Planning:

Every iteration usually begins with iteration planning. A plan is devised by developers of the tasks which implement required attributes of the latest release at this stage. During this phase the developer choses the activities that are to be carried out and evaluates the needed expense, time, and effort for the chosen activity [53]. These activities might be shared between the programmers to stabilize and adjust the load of the work.

Iteration to Release Phase:

This stage includes the fundamental development tasks like designing, coding, testing and integration. Every iteration may take 1-4 weeks. In the very first iteration those stories are chosen that design and consist of the architecture of the product. A pair of developers execute the activities chosen for the recent iteration. The developers choose an activity and then create a basic design to code it [54]. Functional testing is the next step after the completion of coding. In case the programmed code does not satisfy the needs then code refactoring is implied. Multiple iterations might be required for development to end. To monitor development process or to discuss any problems that may arise, standup meetings are held. The code is set for the next phase after the last iteration [55].

Production Phase:

Extreme programming releases small releases of the software as it an incremental process. Continuous release system in this model lets the system to be developed in iterations [56]. A cycle of release may contain many iterations and each iteration may be of one to four weeks of duration. This phase focuses on software delivery in small releases [57]. During production phase developers delay the speed of system evolution to avoid any risks.

Maintenance Phase:

Maintenance is a fundamental process for any software framework. In Extreme programming the software is updated and modified continuously for some time span. A new functionality is developed while the previous is still in use at this stage [58]. Those alterations that might hinder or cause issues with the production are terminated.

Death Phase:

Death phase is the last stage of Extreme programming. To arrive at death phase there are two possibilities. In first scenario the software is released when the built software contains all the required functionality, client satisfaction and zero stories. A record is kept in the shape of a short document of 5-10 pages in case it is needed in the future. In the second scenario "entropic death" of the software occurs and it would be wise to cease the development of the software [59].

4. DATA ANALYSIS

4.1 XP Practices

Extreme programming has 12 practices. These practices make Extreme programming unique and stand out as compared to other software frameworks. During the development of the software these twelve practices are applied considering the principles of Extreme programming [60].

Planning Game: For additional planning requirements of the system are gathered on a story card. During planning game roles and size of the team, time and whole plan and agenda is laid out [61]. This practice is carried out in 2 sections i.e., release planning and iteration planning.

Small Releases: In each delivery a bunch of requirements are created that have a little business value [47]. These small releases make the product accessible and in reach of the client for analyzation and inspection.

Metaphor: It is the structural and architectural plan of the system that depicts how system should function. It is a vital method for the understanding of the system for developers [62].

Simple Design: This practice serves to be an incredible way to develop fundamental required functionality of the framework and steer clear of less important information. It centers around latest required attributes.

Continuous Testing: Continuous testing gives a speedy input and response. Unit testing and acceptance testing are utilized by Extreme Programming persistently.

Refactoring: Refactoring is rebuilding the framework without altering its behavior [61]. To improve the code quality, developers use this activity consistently.

Pair Programming: In extreme programming two software engineers code at one same machine. To build best quality software and a cheaper cost [63], pair programming is utilized since many defects are caught and rectified within a short span of time by the second programmer.

Collective Ownership: The concept of collective ownership of the code is such that any developer can have the approach to any piece of code whenever they desire to modify it [64]. By granting this permission of review by multiple programmers boosts the software's quality.

Continuous Integration: By the time each task is finished the system integration is performed along with testing which lowers the chances of any integration issues and further enhances the quality [65].

40-Hour Week: It is a standard defined by Extreme programming that programmers are not allowed to work more than 40 hours a week. Extra working hours for programmers are not appreciated by this agile model as chances are exhausted and fatigued developers will make more errors.

On-Site Customer: A representative of the client is included in Extreme programming team and stays on site during all the process. This representative is a specialized professional who has the power to choose required attributes of the system, respond to the questions, and can lead the development process. Being on site helps to reduce the communication issues.

Coding Standards: There are a few coding standards that need to be followed in this agile model [66]. The ownership of the code is collective and can be approached and modified by any developer due to which coding standards need to be implied.

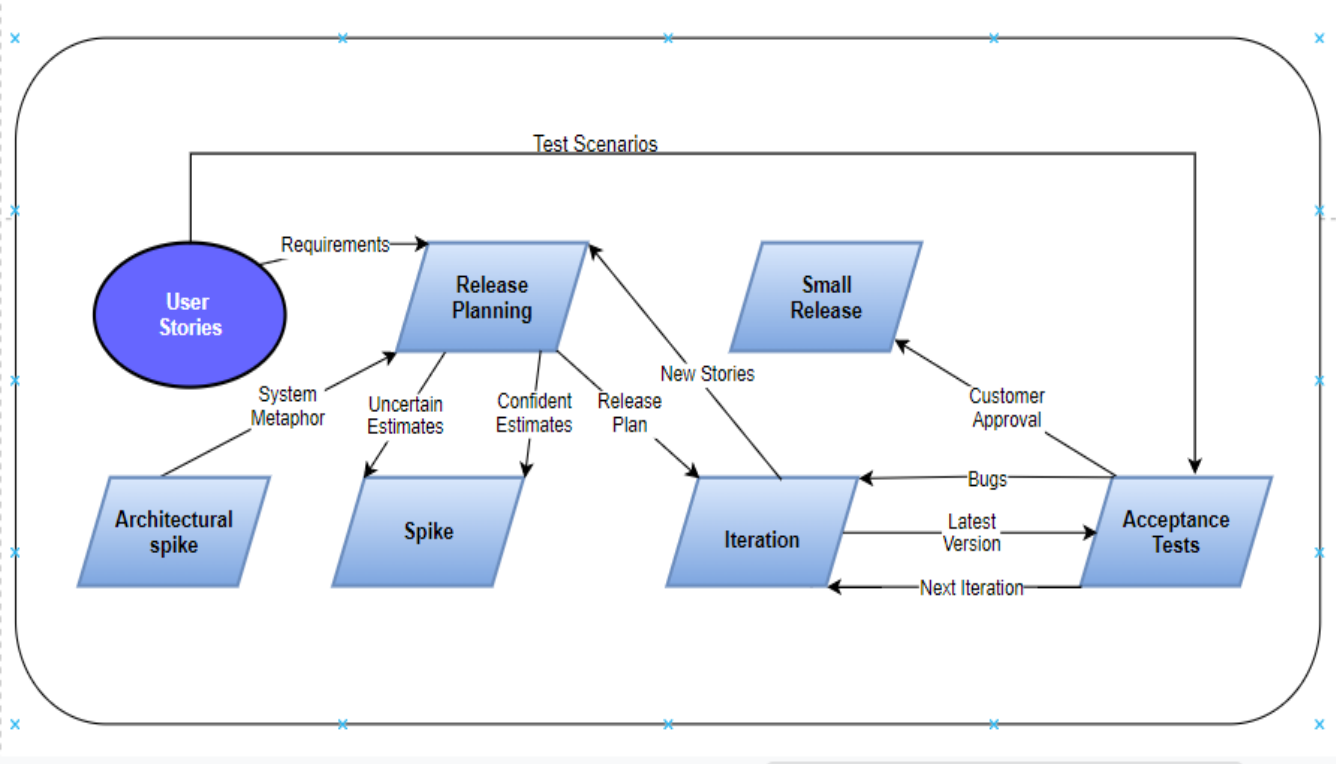

Figure 2: Extreme Programming overview

4.4 Comparison

Extreme programming and Scrum are very popular agile frameworks. To have a different view of these models a thorough comparison is done based on various factors and attributes. For this purpose [67] was considered and consulted. This comparison can be seen in 'Table 1'

Table 1: XP VS Scrum

Extreme Programming	Scrum
It is Iterative and incremental	It is also Iterative and incremental
It has a small project size	Project size: All
The team consists of 2 to 10 members.	More than one team with less than 10 members
It has following team activities: Planning game, Pair programming, Collective code ownership etc.	No Team activities.

1 to 3 weeks is the sprint duration	Sprint duration is 4 weeks
Throughout the process there is stakeholder's involvement	Not Specified
Communication is done Orally, through standup meetings	Also Orally, through Scrum meeting
There is no project management	Project management is present
Co-located teams	Physical Environment is not specified
It focuses more on engineering factors	It focuses on management and productivity.
It has a Quick response towards change	Same as extreme programming
For requirement gathering User stories and on-site customer practices are utilized	Not specified
Difference between different requirements is not specified.	Not specified
Documentation is Less	Same as Extreme programming
There is no upfront design document	Not Specified
Development order is specified by the Customer	Development order is specified by the Scrum Team
The development style is adaptive	It is also adaptive for scrum
The Whole team has the access to code	Not specified
Alteration during iterations is allowed	Alteration during iterations is not allowed
Feedback can be given in minutes to months	Feedback can be delivered in over a month
Unit testing, integration testing and acceptance testing are performed	Not specified
There are no structured review meetings	Same as extreme programming
Functional Testing and Acceptance Testing are performed for validation	Not specified
Test first approach is carried out for QA	Not specified
Coding Standards are properly defined	Coding standards are not defined
Software configuration practices are not defined	Software configuration practices are not defined as well
There is no support for distributed projects	Not defined
No process management	No process management

5. DISCUSSION ON RESULTS

5.1 XP Values

When Extreme programming practices are applied, there are five Extreme programming values that are focused on and they are simplicity, respect, courage, communication, and feedback.

Simplicity: Things are kept simple by this agile model like simple design, simple code and simple plan. Simple solutions are preferred, and no additional functionality is added unless the customer specifies otherwise [68]. Plain and minor repetition of Extreme programming aides in avoiding the risk of project distraction.

Communication: Among the team members, Extreme programming uses continuous and active communication instead of documentation [69]. Meanwhile all the team members and customers continuously communicate on site to find more appropriate and budget-friendly solutions to the problem.

Feedback: The feedback that spans on different time scales is used by Extreme programming. Rapid feedback about the developing software is provided by unit testing and integration testing which are performed daily. The project is kept on the right path by the help of feedback and communication. A distinguishing feature of Extreme programming which is presence of a customer on site aids in getting fast feedback about the developing software.

Courage: Courage is required for Extreme programming practices. It is sometimes required to rewrite the design or code that is completed after substantial effort. It also refers to making decisions that have never been made for the system before.

Respect: Another major value of Extreme programming introduced in [47] is respect. Respect for other members and self-respect is important which makes it possible to execute Extreme programming. The developers can be compelled to do high quality work by showing respect for work [70].

5.2 XP Roles

Seven roles of the team members with their qualities and responsibilities are defined by Extreme programming which they must execute within the team.

Programmer: The most important role in the Extreme programming team is that of a programmer. The main activity in Extreme programming performed by the programmer is coding. All these tasks are to be performed by the programmer hence there is no designer, architect, or analyst in this agile model's team.

Customer: Another major member of the Extreme programming team who takes part in a dynamic role throughout the development process is the customer. He writes stories, derives functional tests, and henceforth verifies those tests.

Coach: A person acting as a coach should have management skills. His decision making and communication skills assist the team members to be on the right path.

Tracker: The tracker's duty is to collect metrics such as load factor and functional test scores regarding the project. After two to three days, the tracker gathers data from all the developers and records the time consumed on a task and the time still required to complete it. It is the tracker's duty to validate the iteration and commitment schedules that are realistic and can be met.

Tester: The tester's responsibility is to conduct and aid the customers in writing functional tests and verifying them [48]. A tester has very less to do as the unit testing is performed by the developers in the Extreme programming.

Consultant: Extreme programming team does not have a specialist however, in some cases when the team needs technical guidance from an expert, a consultant can be hired for a specific period. The solution of the problem is discussed with the consultant by two or more developers in a meeting.

Big Boss: He is a coordinator that has responsibilities of building the team, providing required resources, equipment, and tools of the project. The big boss needs to show determination while supporting the team's decision [49].

6. CONCLUSION AND RECOMMENDATIONS

Extreme Programming and Scrum are well known agile frameworks that are broadly utilized for small scale projects. Best practices of agile industry are utilized by these frameworks for software development. In this research a detailed description of various stages, practices and roles of these models is given. To have a better understanding of these frameworks a comparison is also carried out [53]. This detailed comparison can be extremely useful for all those researchers with an interest in such frameworks of agile. It is uncovered by this comparison that these models have many similarities and differences. A few dissimilar attributes of these models complement each other which can lead to experimentation with combining Scrum and Extreme programming for high quality software development.

REFERENCES

[1] D. de Gois Marques, T. L. D. de A. Dallegrave, L. E. L. Barbosa, C. M. de Oliveira Rodrigues, and W. B. Santos, "Industry-Academy Collaboration in Agile Methodology: a Systematic Literature Review," in *2022 17th Iberian Conference on Information Systems and Technologies (CISTI)*, 2022, pp. 1–6.

[2] S. P. [48] Kodmelwar, M. K., Futane, P. R., Pawar, S. D., Lokhande, S. A., & Dhanure, "A Comparative Study of Software Development Waterfall, Spiral and Agile Methodology," *J. Posit. Sch. Psychol.*, vol. 6, no. 3, pp. 7013–7017, 2022.

[3] S. Einsiedler, S. Mule, C. Rau, P. Hehenberger, and K. Roth, "A macro-level process model for integrating agile approaches in the design of product-service systems," *Int. J. Agil. Syst. Manag.*, vol. 15, no. 1, pp. 70–92, 2022.

[4] D. Y. Nakano, Y., Oura, F., Ichsan, M., Faturohman, T., Bounsaythip, A., Inthakason, B., ... & Wardhana, "A Proposal of Efficient Agile Implementation Process Model to Enterprise Organization by Involving Indirect Departments. RIBER," *RIBER*, vol. 11, no. 1, pp. 111–124, 2022.

[5] A. Hinderks, F. J. D. Mayo, J. Thomaschewski, and M. J. Escalona, "Approaches to manage the user experience process in agile software development: A systematic literature review," *Inf. Softw. Technol.*, p. 106957, 2022.

[6] S. Ashraf and S. Aftab, "IScrum: An Improved Scrum Process Model.," *Int. J. Mod. Educ. \& Comput. Sci.*, vol. 9, no. 8, 2017.

[7] Z. Nawaz, S. Aftab, and F. Anwer, "Simplified FDD Process Model.," *Int. J. Mod. Educ. \& Comput. Sci.*, vol. 9, no. 9, 2017.

[8] S. Ashraf and S. Aftab, "Pragmatic evaluation of IScrum \& scrum," *Int. J. Mod. Educ. Comput. Sci.*, vol. 11, no. 1, p. 24, 2018.

[9] S. Aftab, Z. Nawaz, M. Anwar, F. Anwer, M. S. Bashir, and M. Ahmad, "Comparative Analysis of FDD and SFDD," *Int. J. Comput. Sci. Netw. Secur.*, vol. 18, no. 1, pp. 63–70, 2018.

[10] S. Khan *et al.*, "Latest Transformations of XP Process Model: A Systematic Literature Review," *Int. J. Comput. Sci. \& Netw. Secur.*, vol. 21, no. 6, pp. 143–150, 2021.

[11] A. Rasheed *et al.*, "Requirement engineering challenges in agile software development," *Math. Probl. Eng.*, vol. 2021, 2021.

[12] F. Anwer, S. Aftab, M. S. Bashir, Z. Nawaz, M. Anwar, and M. Ahmad, "Empirical comparison of XP \& SXP," *IJCSNS*, vol. 18, no. 3, p. 161, 2018.

[13] G. S. Matharu, A. Mishra, H. Singh, and P. Upadhyay, "Empirical study of agile software development methodologies: A comparative analysis," *ACM SIGSOFT Softw. Eng. Notes*, vol. 40, no. 1, pp. 1–6, 2015.

[14] S. Aftab *et al.*, "Using FDD for small project: An empirical case study," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 3, 2019.

[15] G. Rasool, S. Aftab, S. Hussain, and D. Streitferdt, "eXRUP: a hybrid software development model for small to medium scale projects," 2013.

[16] A. Cockburn, "Agile software development joins the" would-be" crowd," *Cut. IT J.*, vol. 15, no. 1, pp. 6–12, 2002.

[17] K. Schwaber, "Scrum development process," in *Business object design and implementation*, Springer, 1997, pp. 117–134.

[18] J. A. Highsmith and J. Highsmith, *Agile software development ecosystems*. Addison-Wesley Professional, 2002.

[19] A. A. Janes and G. Succi, "The dark side of agile software development," in *Proceedings of the ACM international symposium on New ideas, new paradigms, and reflections on programming and software*, 2012, pp. 215–228.

[20] K. Beck, *Test-driven development: by example*. Addison-Wesley Professional, 2003.
[21] F. Anwer, S. Aftab, and I. Ali, "Proposal of Tailored Extreme Programming Model for Small Projects," *Int. J. Comput. Appl.*, vol. 171, no. 7, pp. 23–27, 2017.
[22] M. R. J. Qureshi and J. S. Ikram, "Proposal of enhanced extreme programming model," *Int. J. Inf. Eng. Electron. Bus.*, vol. 7, no. 1, p. 37, 2015.
[23] F. Anwer and S. Aftab, "Latest customizations of XP: a systematic literature review," *Int. J. Mod. Educ. Comput. Sci.*, vol. 9, no. 12, p. 26, 2017.
[24] M. Ibrahim *et al.*, "Presenting and Evaluating Scaled Extreme Programming Process Model," *Int. J. Adv. Comput. Sci. Appl.*, vol. 11, no. 11, 2020.
[25] E. Hossain, M. A. Babar, and H. Paik, "Using scrum in global software development: a systematic literature review," in *2009 Fourth IEEE International Conference on Global Software Engineering*, 2009, pp. 175–184.
[26] J. Newkirk, "Introduction to agile processes and extreme programming," in *Proceedings of the 24th International Conference on Software Engineering. ICSE 2002*, 2002, pp. 695–696.
[27] H. F. Cervone, "Understanding agile project management methods using Scrum," *OCLC Syst. \& Serv. Int. Digit. Libr. Perspect.*, 2011.
[28] P. Zhang, H. Wang, and Z. Nie, "Agile Innovation Process Model Based on Computer-Aided Patent Knowledge Mining and Functional Analogy," 2022.
[29] G. Edwards, K., Cooper, R. G., Vedsmand, T., & Nardelli, "Evaluating the agile-stage-gate hybrid model: Experiences from three SME manufacturing firms," *Emerg. Issues Trends Innov. Technol. Manag.*, vol. 1, no. 1, pp. 229–263, 2022.
[30] B. Murphy, C. Bird, T. Zimmermann, L. Williams, N. Nagappan, and A. Begel, "Have agile techniques been the silver bullet for software development at microsoft?," in *2013 ACM/IEEE international symposium on empirical software engineering and measurement*, 2013, pp. 75–84.
[31] K. Beck, "Embracing change with extreme programming," *Computer (Long. Beach. Calif).*, vol. 32, no. 10, pp. 70–77, 1999.
[32] S. Hamadneh, O. Pedersen, M. Alshurideh, B. A. Kurdi, and H. M. Alzoubi, "An Investigation Of The Role Of Supply Chain Visibility Into The Scottish Blood Supply Chain," *J. Leg. Ethical Regul. Issues*, vol. 24, no. 1, pp. 1–12, 2021.
[33] K. Schwaber and M. Beedle, *Agile software development with scrum. Series in agile software development*, vol. 1. Prentice Hall Upper Saddle River, 2002.
[34] D. Cohen, M. Lindvall, and P. Costa, "An introduction to agile methods.," *Adv. Comput.*, vol. 62, no. 03, pp. 1–66, 2004.
[35] J. R. Hanaysha, M. E. Al-Shaikh, S. Joghee, and H. M. Alzoubi, "Impact of innovation capabilities on business sustainability in small and medium enterprises," *FIIB Bus. Rev.*, vol. 11, no. 1, pp. 67–78, 2022.
[36] H. M. Alzoubi and R. Aziz, "Does emotional intelligence contribute to quality of strategic decisions? The mediating role of open innovation," *J. Open Innov. Technol. Mark. Complex.*, vol. 7, no. 2, p. 130, 2021.
[37] J. Sutherland and K. Schwaber, "The Scrum guide. the definitive guide to Scrum: The rules of the game," *ScrumGuides. com*, 2013.
[38] J. Sutherland, *Scrum: a revolutionary approach to building teams, beating deadlines, and boosting productivity*. Random House, 2014.
[39] A. Akhtar, S. Akhtar, B. Bakhtawar, A. A. Kashif, N. Aziz, and M. S. Javeid, "COVID-19 detection from CBC using machine learning techniques," *Int. J. Technol. Innov. Manag.*, vol. 1, no. 2, pp. 65–78, 2021.
[40] S. Joghee, H. M. Alzoubi, and A. R. Dubey, "Decisions effectiveness of FDI investment biases at real estate industry: Empirical evidence from Dubai smart city projects," *Int.*

J. Sci. \& Technol. Res., vol. 9, no. 3, pp. 3499–3503, 2020.
[41] A. Al Ali, "The Impact of Information Sharing and Quality Assurance on Customer Service at UAE Banking Sector," *Int. J. Technol. Innov. Manag.*, vol. 1, no. 1, pp. 1–17, 2021.
[42] H. Alzoubi and G. Ahmed, "Do TQM practices improve organisational success? A case study of electronics industry in the UAE," *Int. J. Econ. Bus. Res.*, vol. 17, no. 4, pp. 459–472, 2019.
[43] N. Radwan and M. Farouk, "The Growth of Internet of Things (IoT) In The Management of Healthcare Issues and Healthcare Policy Development," *Int. J. Technol. Innov. Manag.*, vol. 1, no. 1, pp. 69–84, 2021.
[44] N. Ali *et al.*, "Modelling supply chain information collaboration empowered with Machine Learning Technique," *Intell. Autom. Soft Comput.*, vol. 29, no. 3, pp. 243–257, Jul. 2021.
[45] E. Mnkandla and B. Dwolatzky, "A survey of agile methodologies," *Trans. South African Inst. Electr. Eng.*, vol. 95, no. 4, pp. 236–247, 2004.
[46] E. R. Mahajan and E. P. Kaur, "Extreme programming: Newly acclaimed agile system development process," *Int. J. Inf. Technol.*, vol. 3, no. 2, pp. 699–705, 2010.
[47] K. Beck, *Extreme programming explained: embrace change*. addison-wesley professional, 2000.
[48] H. M. Alzoubi, M. Vij, A. Vij, and J. R. Hanaysha, "What Leads Guests to Satisfaction and Loyalty in UAE Five-Star Hotels? AHP Analysis to Service Quality Dimensions.," *ENLIGHTENING Tour. A PATHMAKING J.*, vol. 11, no. 1, pp. 102–135, 2021.
[49] M. Alshurideh, A. Gasaymeh, G. Ahmed, H. Alzoubi, and B. Kurd, "Loyalty program effectiveness: Theoretical reviews and practical proofs," *Uncertain Supply Chain Manag.*, vol. 8, no. 3, pp. 599–612, 2020.
[50] H. M. Alzoubi and R. Yanamandra, "Investigating the mediating role of information sharing strategy on agile supply chain," *Uncertain Supply Chain Manag.*, vol. 8, no. 2, pp. 273–284, 2020, doi: 10.5267/j.uscm.2019.12.004.
[51] H. Alzoubi, G. Ahmed, A. Al-Gasaymeh, and B. Kurdi, "Empirical study on sustainable supply chain strategies and its impact on competitive priorities: The mediating role of supply chain collaboration," *Manag. Sci. Lett.*, vol. 10, no. 3, pp. 703–708, 2020.
[52] A. Alzoubi, "The impact of Process Quality and Quality Control on Organizational Competitiveness at 5-star hotels in Dubai," *Int. J. Technol. Innov. Manag.*, vol. 1, no. 1, pp. 54–68, 2021.
[53] T. Mehmood, H. M. Alzoubi, M. Alshurideh, A. Al-Gasaymeh, and G. Ahmed, "Schumpeterian entrepreneurship theory: Evolution and relevance," *Acad. Entrep. J.*, vol. 25, no. 4, pp. 1–10, 2019.
[54] A. Q. M. Alhamad, I. Akour, M. Alshurideh, B. A. Kurdi, and H. M. Alzoubi, "Predicting the intention to use google glass: A comparative approach using machine learning models and PLS-SEM," *Int. J. Data Netw. Sci.*, vol. 5, no. 3, pp. 311–320, 2021.
[55] M. Alnuaimi, H. M. Alzoubi, D. Ajelat, and A. A. Alzoubi, "Towards intelligent organisations: An empirical investigation of learning orientation's role in technical innovation," *Int. J. Innov. Learn.*, vol. 29, no. 2, pp. 207–221, 2021, doi: 10.1504/IJIL.2021.112996.
[56] N. Guergov, S., & Radwan, "Blockchain Convergence: Analysis of Issues Affecting IoT, AI and Blockchain," *Inf. Manuf.*, vol. 1, no. 1, pp. 1–17, 2021.
[57] J. R. Hanaysha, M. E. Al Shaikh, and H. M. Alzoubi, "Importance of marketing mix elements in determining consumer purchase decision in the retail market," *Int. J. Serv. Sci. Manag. Eng. Technol.*, vol. 12, no. 6, pp. 56–72, 2021.
[58] N. N. Alnazer, M. A. Alnuaimi, and H. M. Alzoubi, "Analysing the appropriate

cognitive styles and its effect on strategic innovation in Jordanian universities," *Int. J. Bus. Excell.*, vol. 13, no. 1, pp. 127–140, 2017.

[59] T. Mehmood, "Does Information Technology Competencies and Fleet Management Practices lead to Effective Service Delivery? Empirical Evidence from E-Commerce Industry," *Int. J. Technol. Innov. Manag.*, vol. 1, no. 2, pp. 14–41, 2021.

[60] N. Alsharari, "Integrating blockchain technology with internet of things to efficiency," *Int. J. Technol. Innov. Manag.*, vol. 1, no. 2, pp. 1–13, 2021.

[61] O. Kobayashi, M. Kawabata, M. Sakai, and E. Parkinson, "Analysis of the interaction between practices for introducing XP effectively," in *Proceedings of the 28th international conference on Software engineering*, 2006, pp. 544–550.

[62] N. Aziz and S. Aftab, "Data Mining Framework for Nutrition Ranking: Methodology: SPSS Modeller," *Int. J. Technol. Innov. Manag.*, vol. 1, no. 1, pp. 85–95, 2021.

[63] A. A. Kashif, B. Bakhtawar, A. Akhtar, S. Akhtar, N. Aziz, and M. S. Javeid, "Treatment Response Prediction in Hepatitis C Patients using Machine Learning Techniques," *Int. J. Technol. Innov. Manag.*, vol. 1, no. 2, pp. 79–89, 2021.

[64] C. Lee and G. Ahmed, "Improving IoT Privacy, Data Protection and Security Concerns," *Int. J. Technol. Innov. Manag.*, vol. 1, no. 1, pp. 18–33, 2021.

[65] M. A. Khan, "Challenges Facing the Application of IoT in Medicine and Healthcare," *Int. J. Comput. Inf. Manuf.*, vol. 1, no. 1, pp. 39–55, 2021.

[66] E. P. Mondol, "The Impact of Block Chain and Smart Inventory System on Supply Chain Performance at Retail Industry.," *Int. J. Comput. Inf. Manuf.*, vol. 1, no. 1, pp. 56–76, 2021.

[67] F. Anwer, S. Aftab, S. S. M. Shah, and U. Waheed, "Comparative analysis of two popular agile process models: extreme programming and scrum," *Int. J. Comput. Sci. Telecommun.*, vol. 8, no. 2, pp. 1–7, 2017.

[68] A. Cruz, "Convergence between Blockchain and the Internet of Things," *Int. J. Technol. Innov. Manag.*, vol. 1, no. 1, pp. 34–53, 2021.

[69] T. Eli, "StudentsPerspectives on the Use of Innovative and Interactive Teaching Methods at the University of Nouakchott Al Aasriya, Mauritania: English Department as a Case Study," *Int. J. Technol. Innov. Manag.*, vol. 1, no. 2, pp. 90–104, 2021.

[70] D. Miller, "The Best Practice of Teach Computer Science Students to Use Paper Prototyping," *nternational J. Technol. Innov. Manag.*, vol. 1, no. 2, pp. 42–63, 2021.