Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:56 Issue:02:2023
DOI 10.17605/OSF.IO/BCGU9

PRESENTING AND EVALUATING SOFTWARE QUALITY ASSURANCE FOR AGILE METHODOLOGIES BY USING NOVEL SCRUMP

JAVED IQBAL

Department of Information Technology, The Islamia University of Bahwalpur.

DOST MUHAMMAD KHAN

Department of Information Technology, The Islamia University of Bahwalpur.
Corresponding Author Email: khan.dostkhan@iub.edu.pk

NADEEM IQBAL KAJLA

Department of Computer Science, Muhammad Nawaz Shareef University of Agriculture, Multan.
Corresponding Author Email: nadeem.iqbal@mnsuam.edu.pk

MALIK MUHAMMAD SAAD MISSEN

Department of Information Technology, The Islamia University of Bahwalpur.

NAJIA SAHER

Department of Information Technology, The Islamia University of Bahwalpur.

FAISAL SHEHZAD

Department of Information Technology, The Islamia University of Bahwalpur.

Abstract

Agile methods are playing an important role in developing good quality software among all software development methodologies. The basic purpose of the agile method is to fasten the software development cycle as well as its quality. These methodologies provide more benefits than traditional software methodologies with few drawbacks. Therefore, many projects are compiled with agile methods, because it requires fewer resources and generates output in a short period of time. Agile methods are more beneficial than other software development methodologies in the market. There are three agile methodologies including Scrum, Extreme programming, and dynamic system development method. These methods are used by various organizations for their medium and large-scale projects along with their well-organized teams. With the rapid development of the agile method, a hybrid approach is being used in this article to remove all the weaknesses in this method. Several software development organizations are using agile methods because of their low cost and high quality. But still, issues in software quality assurance are faced by the software development organization. So, this article is designed to improve the quality of the software with fewer resources. This novel SCPRUMP methodology is a combination of Extreme Programming (XP), and Scrum (RUM) which reduce software cost, save time and provide high-quality software.

Keywords: Agile Method, Extreme Programming, Scrum, Dynamic System Development Method, DXPRUM, Software Quality Assurance, Rational Unified Process (RUP)

1 INTRODUCTION

The agile software improvement approach has been a very much researched theme throughout recent many years. Its true capacity for empowering more lean and client-situated esteem creation processes makes it important for practically any association. Then again, it is realized that the achievement and expected effect of agile development

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:56 Issue:02:2023
DOI 10.17605/OSF.IO/BCGU9

improvement strategy is reliant upon the way things are placed practically speaking. While obviously simply applying specific practices miss the mark in procuring the worth of agile technique, implementing it in improper hierarchical settings or considering it as a silver shot creates more mischief than they help. To empower logically fitting receptions, one significant basis is recognizing the present status of an association. Context-oriented suitability itself is an element of culture, the intricacy of front and center concerns, type of the worth to be conveyed, and possibly numerous different ways. Subsequently, to assess an association's present status for the utilization of agile development advancement strategy, complex ways are expected to be thoroughly surveyed. However, the inquiries include yet are not restricted to how to construct such an appraisal model so it gives sufficient adaptability to be relevant to various hierarchical settings, which ways to consider in the assessment without settling from viable materialness, how to guarantee the objectivity of the outcomes, how to distinguish improvement regions and guide associations towards those improvement regions stay unanswered. However, there exist a large number of models and different procedures for accomplishing this objective, the current models for the most part need both fastidious logical and modern approval, reasonable materialness, and context-oriented fittingness [2] .

1.1. Agile Methodologies

The agile method is characterized by its versatility and adaptability while creating software items [3], having arisen at a time when software designing techniques demonstrated not adequately outfitted to manage the consistent change and flightiness of the software market [4]. As agile strategies have been broadly utilized these beyond twenty years, a few examinations have detailed their weaknesses, going from hardships in issue understanding and arrangement finding [5] and to increasing the value of absence of consideration regarding the plan and building issues [6]. As the software business keeps on advancing what does as well is expected to flourish in it, and "fixing" agile techniques can be a method for accomplishing legitimate advancement and stay away from the send-off of bothersome items [7]. The agile methodology is shown in figure 1.

Fig 1: Agile Methodology

Authors [8] recommends that agile techniques ought to be joined with different strategies (hereinafter additionally alluded to as "points of support") to conquer their shortcoming, with an important "strategy combo" being that of consolidating them with UCD and Lean Startup [9], as the previous empowers software engineers to see that the client's necessities are met and the last option presents try driven improvement, moderating

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:56 Issue:02:2023
DOI 10.17605/OSF.IO/BCGU9

gamble and directing the age of significant worth to business partners [10]. This consolidated methodology (hereinafter alluded to in that capacity) has been the subject of exploration for quite a while and its prosperity and enhancements upon ordinary agile techniques have been accounted for [11,12]. In any case, there actually is an absence of concentrates completely analyzing an exhibited example of the consolidated methodology. The agile family is shown in figure 2.

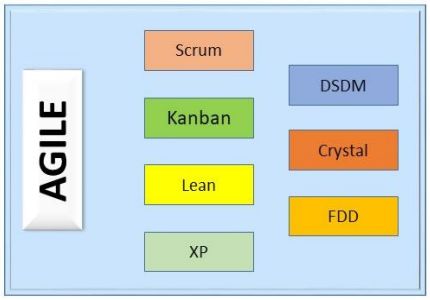

Fig 2: Agile Family

The agile family has the following methodologies:

1.1.1. Scrum

In this approach, the development cycle is split into short (2-3 weeks) with incremental interactions (sprints). The progress of a project is tracked constantly and meetings are organized with various customers and software developers on a regular basis.

1.1.2. Kanban

It is a mechanism in which visual representations of a project are being done in progress. The iteration of limited tasks is taken to see the iteration objectives. In this method, the development team must avoid overburdening.

1.1.3. Lean

The basic purpose of lean is to minimize the waste in software development and to help the team with efficient use of resources and simultaneous communication.

1.1.4. Extreme Programming

It is used to grow the traditional software development methodologies to the external level. It promotes the software to release in small pauses (1-3 weeks). A pair of programmers, one for writing code and the other for reviewing this written code.

1.1.5. DSDM

DSDM (dynamic system development method) is a procedure that ensures rapid software delivery organizing the valuable 80% of the system with 20% time.

1.1.6. Crystal

It balances the projects according to their size and critically applies various methodologies to software.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:56 Issue:02:2023
DOI 10.17605/OSF.IO/BCGU9

1.1.7. FDD

FDD (feature-driven development) is a method in which the overall software model is created, along with preparing a list of features.

To adapt to the advantages and disadvantages of various techniques, the software improvement associations are likewise involving cross models for software advancement [10]. These crossover models are mixes of at least two approaches and are utilized with the mission to combine their advantages in one spot so associations can have the best of them accessible for getting the greatest result [11]. As indicated by Livermore, Scrum is frequently joined with Extreme Programing (XP) rehearses [12]. Both Scrum and XP have various flavors. As per Pressman, to construct an effective software augmentation, DSDM might be joined with XP. This will really produce a combo procedure that characterizes a strong interaction model (the DSDM life cycle) by utilizing stray pieces rehearses (XP) [13]. A few different scientists join the agile method with outdated approaches.

Lina and Dan join Scrum with CMMI to get a superior structure for little and medium-measured associations [14]. Keeping this multitude of focuses in view, the analyst chose to propose another agile improvement model by joining three previously existing models (DSDM, XP, and Scrum) and naming it DXPRUM (D comes from Dynamic Frameworks Advancement Strategy, XP comes from Outrageous Programming and RUM comes from Scrum). The new model covers every one of the qualities of three previously existing models (DSDM, XP, and Scrum) by eliminating their shortcomings. The model additionally gives improved brings about terms of value affirmation when applied to medium-scale projects. The recommended model will likewise function admirably with constantly varying necessities. A concise presentation of these three Dexterous systems is as per the following and furthermore displayed in figure 3.

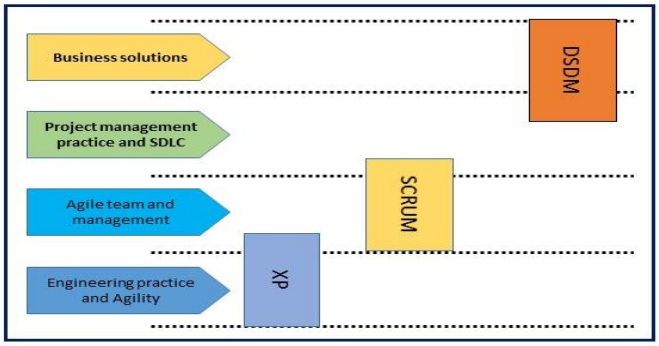

Fig 3: What agile method covers?

1.2. Extreme Programming

XP is the agile approach for software development that assists individuals to create good quality software and furthermore makes lives simple for the improvement group. So, XP is generally polished in little groups with up to 20 individuals and is truly group situated for example the item conveyance is the joint liability of all the engineers in the group

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:56 Issue:02:2023
DOI 10.17605/OSF.IO/BCGU9

instead of the manager of the organization. This approach also called outdated programming goes to exciting levels and is shown in the figure. 4. The essential motivation behind fostering this model is to design a lightweight process model. With regards to proper design practices in computer programming, XP is the most unambiguous among agile systems. It helps software engineers to make the product as indicated by the client's necessities. For instance, it is an agile method that may consist of several announcements in a short period of time. It aids in working on the quality of software with extra care. It advances bunch improvement of an undertaking in which developers and clients are involved broadly [32].

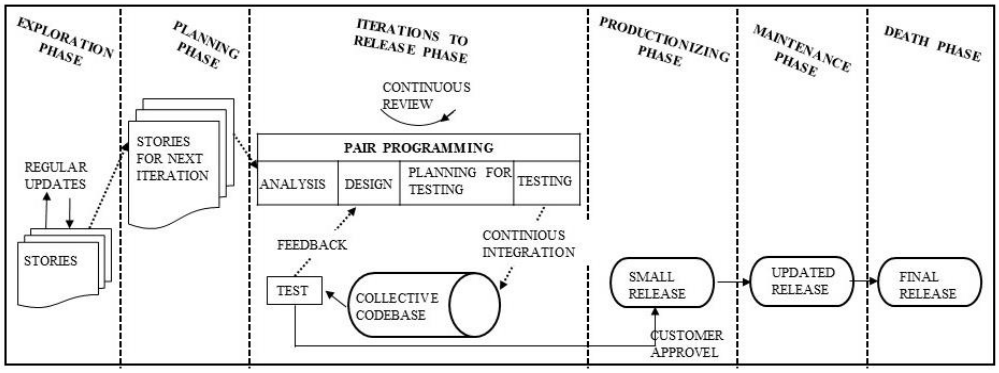

Fig 4: XP Process Model

As they are fostering the product client can recommend changes and updates according to the need, as the issue is better perceived by means of time. EX elevates areas close to one another, preferably in a solitary space for more powerful correspondence. Due to eavesdropping, the element of delay can likewise be dealt with. It additionally, promotes the close client association accordingly it advances that a delegate of the client ought to turn into a basic piece of the group and conveys successfully. Furthermost groups pick XP since it assists with lessening time misfortune while perusing documentation as up close and personal correspondence is there. Engineers can invest more energy carrying out their thoughts as opposed to planning plan thoughts and getting ready documentation. However long the groups are little XP rehearses pay off as it is quicker to share thoughts by talking than to make documentation.

1.2.1. Attributes of Extreme Programming (XP)

In XP, least going with measures are required which implies the requirement for it isn't expected to make documentation and undertaking prerequisites. It is likewise truly group situated, which implies that it is a joint liability regarding every one of the designers to effectively finish the task and not just of the proprietor or manager of the group. XP is greatest rehearsed while the group dimensions are little e.g. normally 12-14 individuals. It additionally advances the contribution of consumer and client at the beginning phase, by this the time squandered by correspondence hole can be protected. It is likewise socially arranged.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:56 Issue:02:2023
DOI 10.17605/OSF.IO/BCGU9

1.3. Hybrid DXPRUM Methodology

This hybrid DXPRUM model is a mixture of three other models generally utilized Dexterous models named Dynamic System Development Model (DSDM), Extreme Programming (XP), and Scrum. The basic purpose of this model is to consolidate the weaknesses of these three models and improve the quality of the overall model along with these advancements as compared to previous software development models. It joins the venture of the board rehearses of Scrum with a business-centered methodology of DSDM. It also covers the entire SDLC below the umbrella of designing practices of XP. It creates DXPRUM more impressive model as compared to previous models for medium-scale projects. The best elements of the DXPRUM model incorporate the board of software projects with proper time requirements and function admirably with evolving necessities. The imperfection degree of DXPRUM is very little while it is analyzing three agile models. These outcomes are excellent software items. The DXRUM has consolidated elements of DSDM, XP, and Scrum. It contains pre-undertaking and post-project periods of DSDM alongside some SDLC includes too. The features of DXPRUM, sprint backlog, and structure backlogs also exist in Scrum. There are various stages of the entire SDLC covered in this proposed designing practice of XP.

These practices incorporate coding guidelines, match programming, refactoring, and aggregate responsibility for also, test-driven improvement. Mix every one of these to make

DXPRUM is an intriguing model for programming improvement associations. The DXPRUM model is shown in figure 5. It begins with the pre-project stage. At this stage, an achievability investigation of the product is to be constructed. Furthermore, this attainability study guarantees whether the product item might be finished inside specific time imperatives and spending plans. While the pre-project stage is directed before the task is formally begun.

On other hand, the principal significant period of software advancement life cycle starts and it is called the useful model stage of software development. In this stage, introductory plans of the items are designed. The contribution to this stage is the SRS report. Then, at that point, the backlog is made. At this point, the necessities order is finished according to primacies. The entire cycle is finished with the conference of the partners being done with stakeholders. These include DXPRUM master, item proprietor, and DXPRUM creation group. Then the item is partitioned into various sprints as given in the proposed model. First of all, the high need necessities ought to be finished and then the principal periods of SDLC are begun. These incorporate a comprehensive plan of the item, advancement, and testing stages. These stages are led below the umbrella of XP rehearses and also incorporate coding principles, match programming, refactoring aggregate responsibility, and test-driven advancement.

When a sprint is finished, a sprint assessment meeting is done. This gathering guarantees whether all the prerequisites of that sprint are executed or not. On the off chance that there are any leftover or new prerequisites investigated in that meeting, these become a

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:56 Issue:02:2023
DOI 10.17605/OSF.IO/BCGU9

piece of the sprint evaluation feedback (SEF) archive which is the contribution to framework overabundance for execution in the next sprint. Although if the whole prerequisites are applied and the item vendor is pleased with the given output sprint, at that time a sprint is released as DXPRUM increment. The post-project stage is considered the final stage of our proposed model. It is one of the phases where the support of the project is done in the future. If any error will be occurred in the future, then it would be managed in the maintenance stage.

The total course of DXPRUM means protecting the overall parts of SDLC. These play a vital role in the creation of software with good quality, minimal expense, and low surrenders. Every sprint cycle in DXPRUM is almost 7-10 days long. While the medium size projects contain 4-8 of such sprints. Subsequently, the period of the undertaking differs from 1.5 to 90 days long. The sprint sorting out gathers in DXPRUM is of 2-4 long stretches of the period. The day-to-day DXPRUM conference is held for at least 30-45 minutes of the sprint. As soon as, again the sprint assessment meeting is held 2-4 hours of term. The various modes of the DXPRUM model are DXPRUM master, item proprietor, and DXPRUM creation group. The DXPRUM jobs, occasions, antiques, and stages are discussed the in sense of exhaustively in the next segments.

2 LITERATURE REVIEW

The point of this exploration work is to concentrate on how Extreme programming could be joined with ISO 9000:2000 guidelines to further develop Nigerian software advancement processes. This examination paper will zero in on XP ìs lightweight, change-situated, and client-arranged way to deal with software improvement, which is the most broadly utilized agile approach. Extreme programming additionally highlights the significance of on-location client, oral correspondence, item quality, short criticism from client and end-clients, straightforwardness, negligible documentation, and escape of extra time. XP centers around nonstop testing, mixing and the need to continuously anticipate change from clients, additionally it accentuation less on documentation. XP is fruitful today since it centers on consumer loyalty and quickly answers changing necessities, particularly in a climate where prerequisites are unbalanced.

It makes conceivable by including clients being developed interaction, enables designers to consistently answer client needs and furthermore accentuates cooperation. Even though XP has a few shortcomings for instance it is a cycle for a little group, it isn't versatile, it needed curios and documentation, and so forth. A relative examination was completed on literary works discoveries, interviews and questionnaire reaction/results from Nigerian software organizations, association programming projects qualities and programming process rehearses were assessed. Then key-related rehearses in ISO and XP models be there recognized, e.g., shortcomings and viable qualities of the two models were additionally distinguished and XP rehearses were planned to ISO statements after exhaustive double-dealing of the elements or attributes of these models [13].

XP is a unique programming model, which is a broadly utilized software process model for the advancement of limited scope projects from the agile family. XP is broadly

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:56 Issue:02:2023
DOI 10.17605/OSF.IO/BCGU9

acknowledged by the software industry because of different highlights it gives, for example, taking care of incessant evolving prerequisites, consumer loyalty, fast criticism, iterative design, group coordinated effort, and little deliveries. Then again, XP additionally holds a few downsides, including less documentation, less spotlight on the plan, and unfortunate design. Because of these constraints, XP is just appropriate for limited scope projects and doesn't function admirably for medium and huge scope projects. To determine this matter numerous specialists have proposed its modified renditions, especially for medium and huge-scope projects. The main problem emerges when XP is chosen for the advancement of limited scope and okay undertaking however steadily because of prerequisite change, the extent of the venture changes from limited scope to medium or huge scope project. In this phase, its construction and practices which functions admirably for little undertaking can't deal with the prolonged possibility. To determine this problem, this paper contributes by proposing a scaled form of the XP process model called SXP. The proposed model can actually deal with such circumstances and can be utilized for little as well concerning medium and huge scope projects with the same proficiency. Besides, the current article additionally assesses the proposed model exactly to mirror its adequacy and effectiveness. A comprehensive and exact investigation is performed and it is seen that the proposed SXP performed well practically in each significant-quality boundary. Be that as it may, to additionally assess the proposed

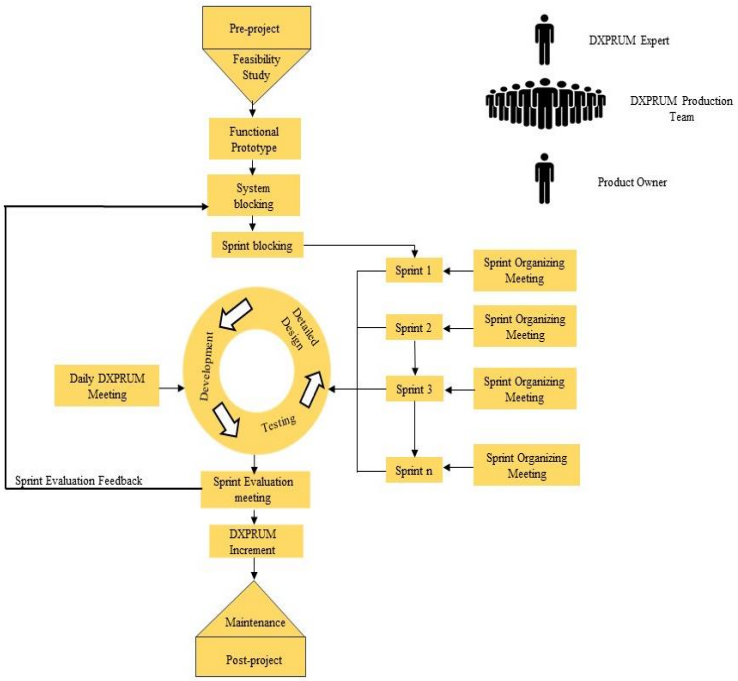

Fig 5: DXPRUM Model

Model, medium, or enormous complex undertaking ought to be picked for improvement [14].

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:56 Issue:02:2023
DOI 10.17605/OSF.IO/BCGU9

In this review, the agile strategies and a risk the executive's approach remained introduced through a specific model. Despite the fact that utilizing this technique brought about risk the board improvement, that model must be utilized in a few explicit activities. That model comprises three fundamental stages including item vision arranging, item guide discharge arranging, and execution every one of which comprises its specific risk of the executive's components [15]. The writing audit additionally shows that a mix of similar strategies utilized for modern activities can be used to risk the executives in software projects. For instance, PMBOK comprises a few information regions to guarantee project achievement [16], [17]. Experimental investigations propose that risk the board isn't viewed as an authority approach in numerous product organizations. Moreover, in some product systems, gambles are not authoritatively checked and in some others, casual techniques are utilized to control risk. In certain investigations, Scrum is likewise presented as one of the risks the board approaches [18]. In an exploration study, a risk of the executive's component was added to Scrum to accomplish ability development model mix (CMMI) targets. The consequences of this study showed that the introduced system had more than 70% progress in distinguishing risks and controlling them [19].

The agile approaches have transformed into an enticing choice for associations endeavoring to upgrade their routine. They deal with a different cycle advancement biological system exemplifying an agile platform. Scrum, Test-driven improvement (TDD), Feature Driven Development (FDD), Extreme Programming (XP), Dynamic System Development Method (DSDM) and Gem strategies are agile families. Specialists accept that an excellent item must be designed by following a great interaction. Jan and Javed attempted to further develop the product improvement process by mixing CMMI level 2 and 3's practices into Scrum and named this work SCXTREME. This mixture method combines the 8 Cycle Regions PAs (out of 22) of the CMMI, Scrum, and XP model. The determination of the Scrum model with other agile methods was a direct result of its wide use in the Pakistani industry. SCXTREME depends on dexterous for example Scrum and XP rehearses joined with the CMMI Explicit Practices SPs.

Design the executives, track and control changes, layout respectability, and hazard the board are for the most part disregarded regions in SCRUM/XP, so SPs of CMMI are added in regards to this region into the planned model. SCXTREME is not difficult to carry out under restricted assets and spending plans. The planned model should be approved in little to average scale associations. To work on the group's efficiency, code quality, and support process, Sultana et al. [S9] proposed a mix of practices from XP, Scrum, and DSDM. The model was an ideal combination of task the board, item designing, and setup of the executives rehearses. Project commencement exercises were explained exhaustively. Every one of the services, stages and relics were portrayed obviously. The job of a specialized author was another expansion for documentation-related exercises. This review will contribute a most recent coarse-grained outline that thusly may direct analysts for future examination tries [20].

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:56 Issue:02:2023
DOI 10.17605/OSF.IO/BCGU9

The agile interaction models supplanted the traditional and customary software advancement techniques due to powerful elements which were not accessible in traditional models [21]. These elements comprise accentuation on client fulfillment, group joint effort, and overseeing evolving prerequisites [22-24]. The agile models survey an iterative furthermore, gradual method of improvement which conveys high-quality software [25-27]. Numerous agile interaction models are utilized by the product business nowadays, for example, Extreme Programming (XP), Scrum, Feature Driven Development (FDD), and Dynamic System Development Method (DSDM) [28, 29]. Extreme programming (XP) is one of the famous agile process models for the advancement of little-scale projects as well as generally utilized by the product business [30, 31]. XP is thought of as one of the generally utilized agile interaction models by the product business for the improvement of little-scale projects.

On other hand the highlighted rehearses, XP has a significant disadvantage also and that is its capacity to deal with just little activities. To determine this problem numerous analysts have presented its redone adaptations explicitly to deal with medium and huge scope projects. Anyway, a genuine issue emerges when the ordinary XP process is chosen for a limited scale and generally safe venture yet with the steady section of time, every now and again evolving prerequisites because of present-day business change haul the degree of task from limited scope to medium or even enormous scope. During this period, a few qualities of traditional XP don't let its life cycle deal with medium or huge activities. These attributes include poor building structure, absence of documentation, less spotlight on the plan, and nonappearance of legitimate change in the board system.

This exploration is being proposed by a scaled rendition of the XP process model which can deal with such circumstances actually. Besides, the recommended model can be similarly powerful for little, medium, and huge scope projects. In this model, more spotlight is given to planning, testing, and especially changing the executive’s strategy. Because of these highlights, SXP can deal with any augmentation in the extent of the venture. An exact assessment is additionally acted to examine the adequacy of the proposed SXP. For this reason, a contextual investigation is directed in which an ongoing client arranged project is created. The exact consequences of software quality measurements are gathered during the turn of events and afterward contrasted and another distributed contextual investigation in which XP was utilized for the improvement of a client situated project. An itemized exact examination is performed and it is seen that the proposed SXP performed well practically in each significant-quality boundary. Anyway, to additionally assess the proposed model, medium or huge complex undertaking ought to be picked for improvement. Correspondingly, Bashir and Qureshi presented a hybrid approach in 2012 for all kinds of medium-scale projects after joining the RUP, XP, and Scrum [33]. Rasool et al. presented another hybrid eXRUP approach in 2013 to develop medium-scale projects [34].

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:56 Issue:02:2023
DOI 10.17605/OSF.IO/BCGU9

3 MATERIALS AND METHODS

There are various agile methodologies that help software developers to design good quality software with fewer resources and in a short period of time. The most popular agile methodologies are Extreme programming and Scrum which are suitable to minimize these time, cost, and resource constraints. So we are choosing both these methodologies to overcome these drawbacks. On other hand, hybrid agile methodologies are gaining fame due to their better results and good quality. The most promising hybrid agile method is DXPRUM, which we have already discussed above in detail. Many scholars have presented their work on agile methodologies and proposed hybrid approaches to remove these drawbacks. In our proposed model a novel Scrump (combination of Scrum and Extreme Programming) is presented to eliminate these drawbacks. The difference between Extreme programming and Scrum is shown below in Table 1.

Table 1: Comparison between XP and Scrum

Quality Parameter	XP	SCRUM
Industrial rehearses	Yes	No
Task managing rehearses	No	Yes
During iteration accept changes	Yes	No
Prerequisite	Yes	No
Prioritizing		
Refactoring	Yes	No
Pair software design	Yes	No
Task dimension	Small to medium	Medium to high
Test-driven progress	Yes	No
Self-association	No	Yes
Entity analysis	Yes	No
Plan methodology	Code centered	Design centered
Certification level	Fewer	Extra
Group size	Less than 10	Less than 10 with several groups
Program style	Unpolluted and modest	Not definite
Technology Surroundings	Fast feedback	Not definite
Bodily Surroundings	Co-located and partial delivery	Not definite
Professional Principles	Collective and obliging	Not definite
Task size	Small to med	Med to high

These are a few limitations that express Scrum and Extreme programming. It is important to combine both agile methodologies to remove all the weaknesses that occurred in software development.

3.1. Novel SCRUMP Methodology

This novel SCRUMP methodology is an improved version of previous agile software methodologies by whole project organization and manufacturing training. The key element of this proposed architecture is to provide a comprehensive product improvement

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:56 Issue:02:2023
DOI 10.17605/OSF.IO/BCGU9

cycle deprived of a Scrum background. All manufacturing practices of Extreme Programming occur due to the sprint cycle of Scrump Methodology as shown in figure 6. In this proposed novel Scrump methodology, sprint zero is recycled earlier than the start of the scrum improvement procedure. It is fundamentally a pre-scrum action in the scrum procedure, but no clear steps are well-defined. The scrum process always starts from the manufactured goods backlog. While this proposed Scrump methodology delivers whole stages of sprint zero. This methodology starts through the formation of manufactured goods characteristics and the final result is a product backlog. This novel Scrump process always starts with goods characteristics and is related to the user stories via customers. These goods characteristics comprise outstanding geographic of new goods such as mandatory through the goods principal. Every piece of goods' characteristics protects definite goals according to the client's requirements. These characteristics contain complete information on customer needs and permit its improvement group to create a sensible approximation of struggle throughout execution. Moreover, the characteristics of the goods become the part of product backlog after completing the approximation process and listing. The characteristics that are selected on approximate always need the struggle to form and apply these characteristics. According to user requirements, these characteristics should be motivated to require GUI plans. All this strategy emphasizes existing necessities and the arrangement of the proposal that is surveyed. There are two types of diagrams developed in these designing stages called class diagrams and object diagrams. The class diagram helps to design the front end (interfaces) while the object diagram builds the backend (databases). Similarly, test classes are developed, which helps the actual improvement of goods throughout this stage. In the end, the coding process needs coding principles, code proprietorship, pair programming, and, nonstop integration.

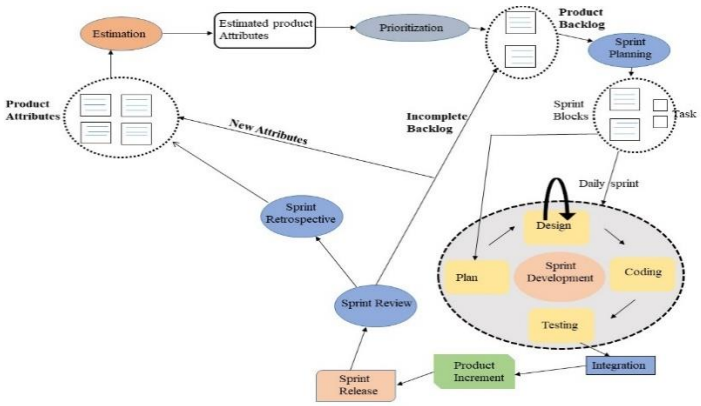

Fig 6: Novel Scrump Methodology

Coding method requires nonstop analysis and refactoring. Code is analyzed normally via unit tests. Every characteristic of goods is arranged, executed, and verified independently along with the sprint improvement cycle. When a fresh code permits via testing, it is then combined with the organization. This practice remains constant to minimize the execution risks. While nonstop integration makes sure working units are vacant to use new features.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:56 Issue:02:2023
DOI 10.17605/OSF.IO/BCGU9

The Scrum conference is held before starting daily routine work. The meeting duration is almost 15 minutes. The key members of this conference consist of the scrum principal, product holder, and improvement group. When sprint iteration is finished then the employed set of goods is unconfined. This portion of the product is accessible at the sprint evaluation conference. All users are requested to sprint evaluation conference.

Finally, when the effective accomplishment of all products is completed then the complete product is propelled with all landscapes. At the last of every sprint, a sprint evaluation conference is held which provides better results to the users. The determination of the sprint evaluation conference guarantees that vital objectives are accomplished or not. The authorization of product augmentation hangs on the degree of client approval. The product holder is shared during the sprint improvement cycle of the novel Scrump methodology to obtain a high level of user gratification. The judgment of subsequent sprint backlog ordering is occupied during this conference. The functioning of product augmentation is announced by examining all the users throughout the sprint evaluation conference. Any modification or fresh recommendation increases through sprint evaluation conference converts the portion of sprint reviewing. The objects of sprint reviewing developed as a portion of the following sprint. Sprint reviewers support group to become more popular to fulfill next sprint. These substances are measured in the backlog that never interrupts the regular working of the product increment. While this method bounds the functioning of product augmentation but permits the chance to convey an effective product augmentation at the termination of every sprint.

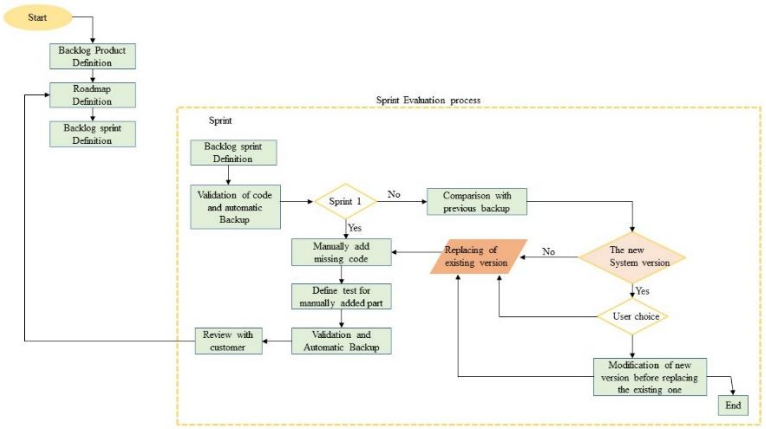

Fig 7: Flowchart for the sprint evaluation process

The sprint evaluation process is shown in above Figure 7. In the first step product backlog is defined according to the system requirements, then the backlog sprint is defined with its roadmap. While in the evaluation process, first of all, validation of the proposed code is presented, and its automatic backup is set here. Furthermore, the first sprint is evaluated and it contains two probabilities e.g. yes/no as shown in Figure 7.

1. If it goes to yes, then we add manually the missing part of this proposed code. After adding this missing part we perform a test on this code for further implementation of our proposed system. As per user requirements, we need to

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:56 Issue:02:2023
DOI 10.17605/OSF.IO/BCGU9

validate again through automatic backup and send the complete version of the software to the client for revision.

2. If it goes to no, then we do a comparison of this sprint with the previous backup and then produce a new version of the system. This new version goes further in two dimensions. If it goes to no, then we replace it with the existing version. If it's foes to yes, then we modify this version according to the client's requirements before replacing it with an existing version. As a result, we deliver it to the user at the end-stage.

3.2 Authentication of Novel Scrump Methodology

This novel Scrum methodology is authenticated by showing a case study to improve payroll application of the University of Central Punjab Lahore, Pakistan. The manager's justification recruits demand the improvement of this payroll application. The team selected for this purpose consist of 8 members and the total period of this project is approximately 4 weeks. The detailed explanation of this case study is shown in Table 2. A total of 6 iterations are accomplished to complete this project. For this purpose, one-week training is held to train the team about this novel Scrump methodology. There are following Scrum practices are presented to the team e.g. sprint zero, product backlog, sprint backlog, sprint arrangement conferences, regular scrum conferences, sprint evaluation conferences, and sprint reviews. The key elements that are introduced to train a team comprise simple plan, cooperative, pair programming, coding ethics, nonstop analysis, and nonstop integration. Following tools are available to fulfill this case study e.g. Rational Rose, Net Beans, My SQL, J-Unit, and IReport.

Table 2: Explanation of Case Study

Case Study Explanation	
Features	**Explanation**
Manufactured goods type	Payroll Application
Size	Medium scale
Project Type	Average
Type of Case Study	Exploratory
Task Period	6 weeks
Repetitions	4
Group scope	8 followers
Software design method	Object-Oriented
Comment	Daily base comments
Semantic	Java
Progress Situation	Net beans 12.0
Booklets	Ms Office XP
Additional apparatuses	Rational Rose
Analysis	J-Unit
Reports	IReport
Web Server	Apache tomcat

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:56 Issue:02:2023
DOI 10.17605/OSF.IO/BCGU9

3.3. Experimental Investigation of the Case Study

For experimental analysis, four sprint releases collect the data. The overall description of the dataset is shown in Table 3. In this case study first, two releases are completed in 2 weeks and the remaining 2 releases in 1 week. While term sprint release is used to represent the actual client test for this proposed Scrump model. The number of components is made through the improvement method and these components are characterized in every sprint release. The overall jobs along these components are characterized in row 3. Each column represents the detail of the total number of tasks defined for this case study. While the overall struggle remains constant for the duration of complete releases. On another hand, the nonstop period of time is 353 h in the first release, while in the remaining releases it is reduced to 170 h, 150 h, and 120 h separately.

The percentage of job struggle is minimized at every release (85% to 60%) in row 6. Here the total number of edges is 48, which are achieved during the overall development process and these edges are shown in row 7. The line of code (LOC) for edges during each sprint release is 16820. While 71 sessions are used to build this development process. The total amount of logical lines of code is produced by the team in a release. The overall group efficiency is most important to design this case study. Here test average is calculated as test LOC/total LOC. Assessment analysis is shown in row 14. The test report fluctuates at every sprint cycle.

While in above Table 3 total assessment analysis is 51.81% which is fairly reasonable for the overall development process. Software Configuration Management (SCM) is described by the integration of data from the project. Moreover, the primary purpose of SCM is to control and track the changes occurred in software. Product quality matters in the overall development process, so it is necessary to make sure the client requirements and client satisfaction. These results show that imperfection concentration is minimum and post-release fault per KLOC is 0.431. The defect density is estimated which is reasonable and provides a sign of a quality product.

There are 17 improvement proposals as presented in row 18. Many proposals are elevated from the first two releases and pair programming is widely applied in product development. The pattern of pair programming is 80% during the complete product development. To accomplish this case study, all clients have shared one office along with the development group. The whole customer participation is almost 25% on regular basis. While client fulfillment is a key point in the complete software development process. Total client participation for this case study is almost 85%, which leads the software team to develop good quality software that meets the user requirements. The client fulfillment rate is different at every sprint release.

The overall representation of the case study shows that we achieve better performance in experimental evaluation. The percentage of client participation and pair programming shows that this case study is valid for small and medium-scale projects. But if we move to large-scale projects it needs more power, resources, and time to fulfill client

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:56 Issue:02:2023
DOI 10.17605/OSF.IO/BCGU9

requirements. Therefore, this case study is valid for medium and small projects. Client fulfillment is the most important part of these agile methodologies. Therefore, after the completion of the whole process, one can easily deliver it to the user.

Table 3: Investigation of case study

SR#	Detail	Release 1	Release 2	Release 3	Release 4	Aggregate
1	Time (weeks)	2	2	1	1	6
2	Items Sprint backlog	8	8	5	3	24
3	Overall Jobs	50	12	14	16	82
4	Overall struggle	300	300	200	200	1000
5	Definite assigned period	353	150	170	120	793
6	Definite jobs assigned	88	87	85	60	80
7	Edges	20	15	7	6	48
8	Sessions	31	30	7	3	71
9	Analysis on sessions	10	8	5	7	30
10	Analysis session LOC	8098	5200	2296	4182	19776
11	Overall LOC	12700	12094	4365	7963	37122
12	Overall KLOC	21.036	3.758	4.365	7.963	37.122
13	Group Efficiency	59.59	25.05	25.68	66.36	46.81
14	Assessment Analysis (%)	54.75	47.37	52.60	52.52	51.81
15	Number of Integration	40	22	30	20	112
16	Post-release faults	7	3	3	3	16
17	Post-release faults /KLOC	0.333	0.798	0.687	0.377	0.431
18	Sprint Retrospective	7	5	4	1	17
19	Pair programming %	80%	80%	80%	80%	80%
20	Client participation	30%	28%	20%	22%	24.5%
21	Client Fulfilment	80%	80%	90%	90%	85%

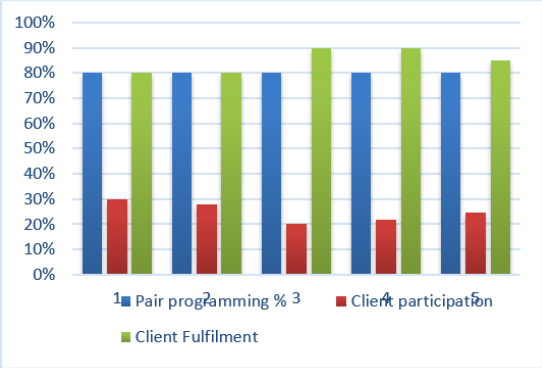

Figure 8: Graphical Representation of Pair programming, Client participation, and Client fulfillment

The graphical representation of Pair programming, Client participation, and Client fulfillment is shown above in Figure 8.

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:56 Issue:02:2023
DOI 10.17605/OSF.IO/BCGU9

4 CONCLUSION

With the rapid development of the agile method, a hybrid approach is being used in this article to remove all the weaknesses in this method. Several software development organizations are using agile methods because of their low cost and high quality. But still, issues in software quality assurance are faced by the software development organization. So, this article is designed to improve the quality of the software with fewer resources. This novel SCPRUMP methodology is a combination of Extreme Programming (XP), and Scrum (RUM) which reduce software cost, save time and provide high-quality software. The proposed Scrump model is the best source to design small and medium-scale software projects in less time. This case study helps the software engineers to design high-quality software and fulfill the user requirements. So, one can say that the current case study proved to be helpful to eliminate the drawbacks of extreme programming and scrum presenting better results.

5 FUTURE WORK

The current case study is presented to evaluate a medium size project. The obtained results show better performance, but these are not suitable for large-scale projects

According to the latest research on agile methodologies, it is hard to expect the future of these developments because:

- It needs a huge experience to apply agile methodologies for small-scale projects, but it is difficult to apply these methodologies for large-scale projects.
- The large companies and organizations need more skills to apply these methodologies for large-scale and tough projects with different circumstances.

Funding Declaration

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Conflict of Interest

Authors declare there is no conflict of interest regarding the publication of this paper.

Authors Contribution

All authors contributed equally to this article. Author Javed Iqbal did survery, collected data and did analysis on that data. Dost Muhammad khan and Nadeem Iqbal Kajla supervises the all research work and did the writeup. Malik Muhammad Saad Missen and Najia saher contributed in writeup and supervision of the research task and experimentation with Javed Iqbal. Faisal shehzad contributed in writeup and figures and tables of the article.

References

1. Bhavsar, K., Shah, V. and Gopalan, S., 2020. Scrum: An agile process reengineering in software engineering. International Journal of Innovative Technology and Exploring Engineering (IJITEE), 9(3), pp.840-848.
2. Tuncel, D., Körner, C. and Plösch, R., 2021, June. Setting the Scope for a New Agile Assessment

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:56 Issue:02:2023
DOI 10.17605/OSF.IO/BCGU9

Model: Results of an Empirical Study. In International Conference on Agile Software Development (pp. 55-70). Springer, Cham.

3. Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R. and Kern, J., 2001. The agile manifesto.
4. Barry, E.J., Mukhopadhyay, T. and Slaughter, S.A., 2002. Software project duration and effort: an empirical study. Information Technology and Management, 3(1), pp.113-136.
5. Lindberg, T., Meinel, C. and Wagner, R., 2011. Design thinking: A fruitful concept for it development? In Design thinking (pp. 3-18). Springer, Berlin, Heidelberg.
6. Gregory, P., Barroca, L., Sharp, H., Deshpande, A. and Taylor, K., 2016. The challenges that challenge: Engaging with agile practitioners' concerns. Information and Software Technology, 77, pp.92-104.
7. Verganti, R., 1997. Leveraging on systemic learning to manage the early phases of product innovation projects. R&D Management, 27(4), pp.377-392.
8. Vilkki, K., 2010, October. When agile is not enough. In International Conference on Lean Enterprise Software and Systems (pp. 44-47). Springer, Berlin, Heidelberg.
9. Grossman-Kahn, B. and Rosensweig, R., 2012. Skip the silver bullet: driving innovation through small bets and diverse practices. Leading through Design, 18, p.815.
10. Vargas, B.P., Signoretti, I., Zorzetti, M., Marczak, S. and Bastos, R., 2020. On the understanding of experimentation usage in light of lean startup in software development context. In Proceedings of the Evaluation and Assessment in Software Engineering (pp. 330-335).
11. Ximenes, B.H., Alves, I.N. and Araújo, C.C., 2015. Software project management combining agile, lean startup and design thinking. In Design, user experience, and usability: Design discourse (pp. 356-367). Springer, Cham.
12. Signoretti, I., Marczak, S., Salerno, L., de Lara, A. and Bastos, R., 2019, September. Boosting agile by using user-centered design and lean startup: a case study of the adoption of the combined approach in software development. In 2019 ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM) (pp. 1-6). IEEE.
13. Erharuyi, E., 2007. Combining eXtreme Programming with ISO 9000: 2000 to Improve Nigerian Software Development Processes.
14. Ibrahim, M., Shabib Aftab, M.A., Iqbal, A., Khan, B.S., Iqbal, M., Ihnaini, B.N.S. and Elmitwally, N.S., 2020. Presenting and Evaluating Scaled Extreme Programming Process Model. Int. J. Adv. Comput. Sci. Appl, 11(11), pp.163-171.
15. Nyfjord, J. and Kajko-Mattsson, M., 2008, September. Outlining a model integrating risk management and Agile software development. In 2008 34th Euromicro Conference Software Engineering and Advanced Applications (pp. 476-483). IEEE.
16. Albadarneh, A., Albadarneh, I. and Qusef, A., 2015, November. Risk management in agile software development: A comparative study. In 2015 IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT) (pp. 1-6). IEEE.
17. Godse, M. and Rajiv, B., 2021. Improvement of Project Management Knowledge Areas Using Scrum Technique. In Optimization Methods in Engineering (pp. 133-149). Springer, Singapore.
18. Afshari, M. and Gandomani, T.J., 2022. A novel risk management model in the Scrum and extreme programming hybrid methodology. International Journal of Electrical and Computer Engineering, 12(3), p.2911.
19. Natarajan, M. and Govindarajan, Y., 2014. Performance comparison of single and multiple

Tianjin Daxue Xuebao (Ziran Kexue yu Gongcheng Jishu Ban)/
Journal of Tianjin University Science and Technology
ISSN (Online): 0493-2137
E-Publication: Online Open Access
Vol:56 Issue:02:2023
DOI 10.17605/OSF.IO/BCGU9

watermarking techniques. International Journal of Computer Network and Information Security, 6(7), pp.28-34.

20. Anwer, F., Aftab, S., Waheed, U. and Muhammad, S.S., 2017. Agile software development models tdd, fdd, dsdm, and crystal methods: A survey. International journal of multidisciplinary sciences and engineering, 8(2), pp.1-10.
21. Ashraf, S. and Aftab, S., 2017. IScrum: An Improved Scrum Process Model. International Journal of Modern Education & Computer Science, 9(8).
22. Alshehri, S. and Benedicenti, L., 2013, May. Prioritizing CRC cards as a simple design tool in extreme programming. In 2013 26th IEEE Canadian Conference on Electrical and Computer Engineering (CCECE) (pp. 1-4). IEEE.
23. Qureshi, M. and Hussain, S.A., 2012. An Improved XP Software Development Model. arXiv preprint arXiv:1202.2501.
24. Qureshi, M.R.J. and Bajaber, F., 2016. COMPARISON OF AGILE PROCESS MODELS TO CONCLUDE THE EFFECTIVENESS FOR INDUSTRIAL SOFTWARE PROJECTS. Cell, 966, p.536474921.
25. Williams, L., 2010. Agile software development methodologies and practices. In Advances in computers (Vol. 80, pp. 1-44). Elsevier.
26. Williams, L., 2010. Agile software development methodologies and practices. In Advances in computers (Vol. 80, pp. 1-44). Elsevier.
27. Anwer, F., Aftab, S., Waheed, U. and Muhammad, S.S., 2017. Agile software development models tdd, fdd, dsdm, and crystal methods: A survey. International journal of multidisciplinary sciences and engineering, 8(2), pp.1-10.
28. Ramírez-Bedoya, D.L., Branch-Bedoya, J.W. and Jiménez-Builes, J.A., 2019. Metodología de desarrollo de software para plataformas educativas robóticas usando ROS-XP. Revista Politécnica, 15(30), pp.55-69.
29. Anwer, F. and Aftab, S., 2017. Latest customizations of XP: a systematic literature review. International Journal of Modern Education and Computer Science, 9(12), p.26.
30. Kazi, S., Bashir, M.S., Iqbal, M.M., Saleem, Y., Jameel Qureshi, M.R. and Raza Bashir, S., 2014. REQUIREMENT CHANGE MANAGEMENT IN AGILE OFFSHORE DEVELOPMENT (RCMAOD). Science International, 26(1).
31. Qureshi, M., 2017. Evaluating the Quality of Proposed Agile XScrum Model. International Journal of Modern Education & Computer Science, 9(11).
32. Shrivastava, A., Jaggi, I., Katoch, N., Gupta, D. and Gupta, S., 2021, July. A Systematic Review on Extreme Programming. In Journal of Physics: Conference Series (Vol. 1969, No. 1, p. 012046). IOP Publishing.
33. Qureshi, M., 2012. Empirical evaluation of the proposed exscrum model: Results of a case study. arXiv preprint arXiv:1202.2513.
34. Rasool, G., Aftab, S., Hussain, S. and Streitferdt, D., 2013. eXRUP: a hybrid software development model for small to medium scale projects.