

ScrumSpiral: An Improved Hybrid Software Development Model

Tapu Biswas

Department of Computer Science, American International University-Bangladesh, Dhaka, 1229, Bangladesh
E-mail: tapubiswas731@gmail.com
ORCID iD: <https://orcid.org/0009-0005-5267-8424>

Farhan Sadik Ferdous

Department of Computer Science, American International University-Bangladesh, Dhaka, 1229, Bangladesh
E-mail: sadikferdous@yahoo.com
ORCID iD: <https://orcid.org/0009-0007-7686-164X>

Zinniya Taffannum Pritee

Department of Computer Science, American International University-Bangladesh, Dhaka, 1229, Bangladesh
E-mail: zinniyataffannumpritee@gmail.com
ORCID iD: <https://orcid.org/0009-0006-2755-9770>

Akinul Islam Jony*

Department of Computer Science, American International University-Bangladesh, Dhaka, 1229, Bangladesh
E-mail: akinul@aiub.edu
ORCID iD: <https://orcid.org/0000-0002-2942-6780>

*Corresponding Author

Received: 14 August 2023; Revised: 10 October 2023; Accepted: 21 November 2023; Published: 08 April 2024

Abstract: In the lightning-quick world of software development, it is essential to find the most effective and efficient development methodology. This thesis represents "Scrum Spiral" which is an improved hybrid software development model that combines the features of Scrum and Spiral approach to enhance the software development process. This thesis aims to identify the usefulness of "ScrumSpiral" methodology and compare it with other hybrid software development models to encourage its use in software development projects. To develop this hybrid model, we did extensive research on the software engineering domain and decided to create a hybrid model by using Scrum and Spiral, named "Scrum Spiral" which is suitable for complicated projects and also for those projects whose requirements are not fixed. Traditional software development models face numerous challenges in rapidly changing markets. By developing this kind of hybrid model, we want to overcome these kinds of limitations and present the software development community with a novel concept for better project results. Final outcome of this thesis was that we developed a model that should be able to complete the project according to the expected schedule, satisfy customer requirements, and obtain productivity through team coordination. The significance of the hybrid model "Scrum Spiral" is reflected in its ability to offer flexibility towards various size projects, proactive risk management to identify all risks before developing the system, and result in higher-quality outcomes for those projects whose requirements are not properly described initially in the project.

Index Terms: Hybrid Model, Software Development Lifecycle, Scrum, Spiral, Project Management.

1. Introduction

The first software development life cycle models (SDLCMs) were published in 1956 [1]. The purpose of software development life cycle models (SDLCMs) describes the process of completing or developing a software project through well-defined processes or stages. Since then, many software development life cycle models (SDLCMS) have been developed. Software development model faces numerous challenges in rapidly changing markets. To handle this problem industrial companies, develop specific development approaches by encompassing all the features of

well-structured methods and flexible agile models [2]. Companies tend to use a hybrid approach for software development [3]. The reasons for using a hybrid strategy are determined by the project's budget and criticality level. Another factor that can influence a company's decision to choose a hybrid approach is because of the company's experience with many IT projects and the work experience of its team members in the IT sector. This is the reason this research is motivated to build a hybrid model by combining the best features of both traditional and agile software development models and building a hybrid approach that can offer the flexibility and responsiveness of agile methods, as well as the structure and control of traditional methods.

There are numerous widely accepted SDLC models in use. Yet, none of them are absolutely flawless, and each and every model has advantages and also some disadvantages. Software development using Scrum, an agile methodology focused on incremental and iterative procedures, Agile process models are more flexible by their nature than others [4]. Although this model is very flexible, it has some drawbacks as well. They are high chances of project failure, restricted team size, challenging quality implementation, difficulty implementing the Scrum framework in large teams, and last of all, any team member leaving in the middle of a project could have a devastating effect on the project. On the other hand, one of the most significant models for the software development life cycle that supports risk handling is the Spiral model. The Spiral model is an incremental model that composes the elements of designing as well as prototyping in different stages of its development [5]. The Spiral model has some constraints as well. Especially in comparison to other SDLC models, this model is more complex and expensive; the risk analysis phase is vital to the project's success and works poorly for smaller projects.

There are numerous development models in the development industries that are used frequently during many projects. Here this research paper showcases Spiral and Scrum so that they can overcome each other's demerits and make a self-standing hybrid model that can further help with developing a time-efficient and fulfilling project. Scrum has some drawbacks that can be fulfilled by combining it with the Spiral model and furthermore, its modules and added features also help to develop the model more efficiently than Spiral and Scrum. This thesis paper will elaborate on the answer to these two research questions in the paper:

- Why select the Spiral and Scrum development model to build this Hybrid model?
- Is this Hybrid model better in efficiency than the Spiral and Scrum development model?

In the world of software development, every software project has its unique requirements to fulfill to successfully accomplish its implementation. Consequently, several projects cannot be accommodated by a single software process [6]. The Spiral was first introduced by Barry W. Boehm in 1988 [7] and it has advanced iterative-type software development. The Spiral model has some advantages like risk management, detail planning before development, customer evolution before the next development, flexibility to combine various approaches, good for a large and mission-critical project, and early development of a project. Starting in the early 1990s, the Scrum was originally designed for product management and development [8]. As well as Spiral, Scrum has also advantaged such as being ideal for rapidly changing requirements, being fast, quick, and adapting to changes easily, requirement prioritization, consisting of small sprint development, and flexible sprint according to the release date. These benefits influenced our decision to use Spiral and Scrum software development models to create this hybrid model. Software models play a significant role as specific software models are required in almost every industry as well as in every business. With the passing of time, it becomes more and more. Many companies use their own personalized methodology for developing their software [9]. ScrumSpiral hybrid model tries to obtain best qualities to create best-quality software as per client and market need. As in our hybrid model, this paper tries to make a better version of the software model by combining two software development models Spiral and Scrum. These models have drawbacks in addition to their benefits. The Spiral model is not suitable for a small project, with specific expertise for risk analysis, and complex, well-qualified developers. Furthermore, Scrum is highly dependent on the team, Lack of visibility disorganized documentation, fewer customer interactions with the team, is not suitable for a large project, and no risk analysis. By integrating these two models, this paper has attempted to address and overcome the disadvantages of each separately. This ScrumSpiral model can help to improve the effectiveness of software models in developing projects by eliminating their drawbacks and introducing new features.

2. Literature Review

This paper discusses several significant studies that compared the hybrid model described in this work to other models that were integrated. In the past decade, many software process models have been developed and introduced. Here this study has mainly included the hybrid models that had been introduced over the past years. In this section, the study provided the research gap between the existing work and the ScrumSpiral hybrid model.

Dhruv Doshi et al. [10] described the four steps of Spiral development, maintain user involvement, maintain budget along with the complexity of the project, and flexibility for changes. Iqbal H. Sarker et al. [11] explained that the Spiral development model maintains a high amount of risk analysis, mission-critical projects, and strong documentation control. On the other hand, despite its merit, it fails to maintain small projects and cost maintenance projects. Overall, it needs high expertise people to handle it and the project mainly depends on risk analysis. ScrumSpiral model maintains

small projects as well as big projects. The ScrumSpiral model also has an expert team member that handles the project efficiency as well as the complexity of the project. It made an estimated budget at the beginning of the project to maintain the project's resources and time.

M. Mahalakshmi et al. [12] explained that the Scrum development model is fast, quick, and can adapt to changes easily, control project schedules, ideal for rapidly changing and accumulating requirements. Wherever it also mentions that in Scrum documentation is very less, never changes the schedule or sprint, and adjusts the scope if needed to meet release dates. ScrumSpiral maintains a constant documentation process, it changes its schedule according to the project size, and customers can add additional requirements in the development phase. This model maintains customer involvement throughout the whole process.

Table 1. Advantages and disadvantages of scrum and spiral model

Model	Advantages	Disadvantages
Spiral	<ol style="list-style-type: none"> 1. A significant amount of risk analysis. 2. Suitable for huge and mission-critical projects. 3. Beginning of the software life cycle software is developed. 4. Tight approval and documentation control. 5. At a later date additional functionality can be added. 	<ol style="list-style-type: none"> 1. It is very complex. 2. Can be an expensive model to use. 3. Risk analysis needs highly specific expert members. 4. The success of the project is critical. 5. Dependent on the risk analysis phase. 6. Ineffective for smaller projects
Scrum	<ol style="list-style-type: none"> 1. Accept rapid changes and be flexible to changes 2. Project cost determined during the project 3. Team adaptability and creativity are endless. 4. A short, fixed schedule per cycle with a flexible scope. 	<ol style="list-style-type: none"> 1. Documentation is much less. 2. Never change the schedule or sprint 3. If team members are cooperative, the project will fail. 4. There is no customer involvement in the middle of the sprint

Farrukh Musa and Muhammad Ali Tariq [13] implemented an agile methodology that operates through a series of sprints that involves breaking down the entire project into smaller pieces and then priority-based implementing them. The manager's good understanding of the client helps to satisfy the requirements of the client. The methodology first completes and tests each sprint sequentially. In this method, clients can make small changes to the budget or schedule, require very limited planning, give high priority to the client to approve the product in each step of development and maintain lower cost and quick response. ScrumSpiral development model gets user stories then proper planning is conducted, then requirements are prioritized. After prioritizing, risk analysis and development are done on the priorities where the customer can modify the project through testing. Then approving all the iterations from the customer, the team integrates and again tests the whole project through alpha and beta testing.

Ghulam Rasool et al. [14] proposed a hybrid model eXRUP that rarely involves the customer in providing a response when the iteration begins. After risk analysis, diagrams are used to communicate how the system will work. The production phase development and validation of modules are ensured to make the system bug-free. The released project and integration with the previously developed module have been maintained by the team along with integration testing of the software. In the released phase, chances of errors are minimized as well, and user manuals and documents are finalized. Alpha testing is performed by the potential customer. In ScrumSpiral risk analyses are done on the priorities before development. During the development phase after each iteration customer approves the iteration then the next iteration proceeds to be executed. Again, the development is flexible according to the project size and customer demand. The ScrumSpiral model splits the whole project into small iterations to make the development easier and faster.

Brijendra Singh and Shikha Gautam [15] implemented a Hybrid Spiral Model in which the model uses tacit knowledge for both input and output. User manuals and system analyst knowledge are used as inputs, and after gathering requirements, draft requirements are developed as an output. A rough architecture is constructed. The design document uses that draft architecture as input. A software code is created that the code developer runs and checks it using knowledge conversion, and that module is the result. Using knowledge conversion module is tested to find out errors or bugs and fix them. After correction, a draft of the product is produced to send to the user for any comments or ideas. The model uses a cyclic and iterative flow of knowledge and if it satisfies the draft product and its quality then the iteration will get a stop. The ScrumSpiral development model involves a sequential and Spiral flow of production. Hereafter gathering requirements, conducting planning, and estimation, the requirements are prioritized by the team with the approval of the customer. The team then performs risk analysis and implements an iteration. The whole project is divided into three iterations that will be implemented in the development phase sequentially after customer approval. Each iteration implementation time is flexible according to the project constraint, size, and requirements. The integrated project is tested by the team. Finally, proper documentation and maintenance are carried out before the project's final release.

Gul Ahmad et al. [16] presented an XSR hybrid model that has three phases of the whole process inception phase, construction phase, and transition phase. In inception, the team defines the model, settles with stakeholders according to the project plan, and requirements, builds the team, finalizes the project budget, and removes the risks, and defines technical strategy initially. In construction, this timeline is broken up to create iterations. Also, this phase focuses on changes in stakeholder requirements, quality improvement, and major risks. The transition phase focuses on implementation and delivering the project into the marketplace. For external systems, high-level testing is done before

release, which involves multiple alpha and beta tests. The ScrumSpiral model focuses on risk analysis and development on each part of the iteration. Again, approving each iteration from the customer will reduce the risk of the whole project as well as lessen the testing time later. Proper documentation will be conducted throughout the whole process.

Zaigham Mushtaq and M. Rizwan Jameel Qureshi [17] implemented a Novel Hybrid Model that involves the planning phase, design phase, coding phase, and testing phase. The process of planning involves the sprint backlog that is planned for development. During this phase, the scrum team creates class and object diagrams. Additionally, they design test classes for testing purposes. During the coding process, continuous testing and refactoring are essential. Each team member is allowed to modify any part of the code at any time. After the written code is tested by unit tests and testing is started even before writing code to understand the problem to programmers. Each feature of the new code is integrated into the existing codebase as soon as it passes testing. Daily scrum meetings are conducted regularly. A working set of the product is released in a sprint review meeting. The ScrumSpiral development model uses an iterative approach, with the entire project divided into three iterations that are implemented in the development phase. Each iteration's implementation time is flexible based on project constraints, size, and requirements, and the team seeks approval from the customer before proceeding with the next iteration. After all, three iterations have been implemented, the team integrates and tests the entire project through alpha and beta testing. Proper documentation goes through the whole process and maintenance is done before and after the final release.

Sara Ashraf and Shabib Aftab [18] showcased IScrum, an improved Scrum process model where the product owner manages the requirements. A workshop is arranged that improves team collaboration. Customer active participation in projects is wished as it helps them to be in charge of the project. All the work that the team needs to do should be included in the product backlog to make the team's work visible. The QA engineer lags one sprint behind because they have to wait for the release before testing. The technical writer can be an integral player in the team conducting documentation from initialization to the end of the project. Finally, it is suggested to hold the daily scrum meeting at the end of the working day. The first stage of the ScrumSpiral model involves obtaining user stories from the client and conducting estimation, planning, and prioritization by the team with approval from the customer. While requirement gathering both the scrum master and product owner participate in it because sometimes the product owner can be confused about requirements. The second stage involves the development and risk analysis of the priorities. The team member ensures a clear understanding of risks and constraints to identify and mitigate potential risks. The development phase is divided into three iterations. Customer involvement throughout the development phase increases customer satisfaction. In the last integration and testing the entire project has been implemented through alpha and beta testing. This stage ensures that the project is functioning as intended and that any issues are identified and resolved before the final release. The final stage of the ScrumSpiral model involves final documentation, user manual, and maintenance before and after the final release.

3. Research Methodology

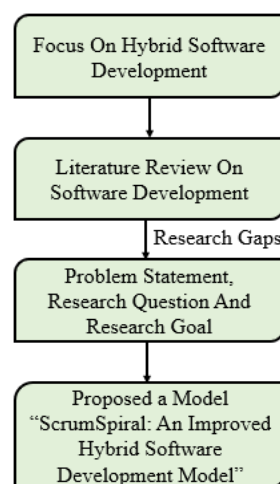


Fig.1. Working strategy

There is a persistent demand for new and improved software development approaches because the field of software engineering is rapidly increasing, so this paper concentrated on hybrid software development. This research showcased an extensive analysis of the literature review and tried to fill the research gap which combines the best principles of many methodologies. Firstly, software models like Scrum, XP, Spiral, Waterfall, and many others were analyzed to figure out the advantages and limitations of these models. After that Hybrid models were intensively analyzed to find out the existing work in this field and the limitations of these models to overcome them in our hybrid model. Scrum is a widely used model, but Spiral has not been introduced by many. These two models have the potential to overcome each other's limitations that why ScrumSpiral, a concept that blends Scrum and the Spiral model, grabbed our interest

specifically. Scrum is an agile process software model that manages complex product development, planning, and management [19]. On the other hand, the Spiral software development model is comparable to the incremental model and provides more emphasis on risk analysis [20]. Finally, this paperwork was conducted based on these two models since these models have received a lot of attention but have not yet been examined by extensive research. Then we find the problem statement, and some research questions and identify what the research goal will be. So, this research paper combined two different models, Scrum and Spiral, and proposed the idea of “ScrumSpiral: An Improved Hybrid Software Development Model”.

4. Proposed Model

The proposed ScrumSpiral model is an improved hybrid software development model that collaborates features of the Scrum and Spiral models. The hybrid model focuses on maintaining collaboration between the project team and the customer to ensure that the project meets the client's expected desired requirements. The ScrumSpiral model is divided into three major phases the following phases: requirement phase, development phase, and release phase.

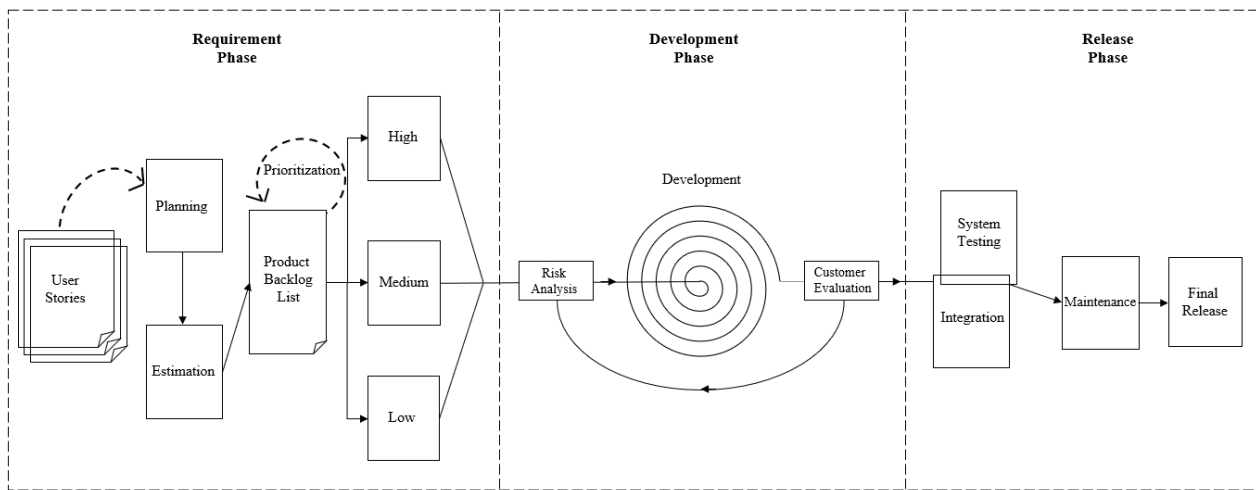


Fig.2. ScrumSpiral model

4.1. Requirement Phase

The process requirement phase involves 4 activities such as collecting user stories, planning, estimation, and product backlog list. At first, the customers write out the story cards that they wish to be included in the project [14]. This way the team can understand the customer's given requirements better as they elaborate them as a storytelling system without feeling any hesitation. The customer elaborates on all the features and requirements that they want to include in the project. Then planning involves customers and team members negotiating between them so that requirements can be sorted out from the user stories. Functional and non-functional requirements are further categorized from the negotiated requirements [14]. The estimation section accurately ensures the amount of effort, such as time and money, to determine the efficiency of the project. The total project time and each iteration duration depend on the project size. This section also includes the identification of tools, and technologies, and the definition of the project team for the system. Budget estimation is considered after the time and resource estimation. A Product Backlog list contains all the requirements that are taken from the planning phase [12]. At the end of the requirement phase, the requirements are prioritized which defines the workflow of the project [12].

4.2. Development Phase

The development phase starts after getting the prioritized product backlog list from the requirement phase. Firstly, the high-prioritized requirements from the product backlog go into the development phase then the medium, and lastly the low-prioritized requirements. This phase begins with risk analysis that identifies all risks and possible solutions then evaluates them [10] and selects the best possible strategy after risk analysis is completed, it starts developing the requirements according to prioritization in each spiral, each spiral has a maximum of 25 days for high prioritized requirements, 18 days for medium prioritized requirements, and 10 days for low prioritized requirements from the product backlog. After completing each spiral, the developed part of the software is shown to the customer, if the customer approves the developed software, it pursues the release phase, or if it is not approved by the customer it takes notes from the customer it goes back to the risk analysis phase until the software is approved by the customer. After completing each prioritized requirement from the product backlog list, further it continues with the release phase.

4.3. Release Phase

The release phase starts after each feature of the product is designed, implemented, and approved individually within the development phase. All the iteration is integrated into existing code as soon as it is passed through the development phase [8]. Integrating the whole process supports the code's structural infrastructure. This process continues until the software is completed. After integration, in the testing section alpha testing is performed among only a small group of people, such as the product's developers and testers. During alpha Testing functionality is tested as well as usability. The beta testing includes a substantial number of people like the general public who give feedback after testing the program. Throughout the beta testing usability, functionality, security, and reliability are being evaluated in the user's environment. Alpha and beta testing ensure product quality before the final release. In the maintenance phase, the team has to manage the whole released project which will be to the customer and maintain the integration with the previously developed component of the product. This is the last phase of the process and starts working after the software is complete and without errors. In this phase, the finalized version of the software project is imparted to the customer. A team member will assist in maintaining a throughout procedural document of the whole process [18]. It will help to keep detailed documentation of the whole system. User manuals and documents are completed and finalized in this phase so that the new users can use them for training [18]. Users can get help from the manuals at any stage in the future or during a problematic situation. When a new team member is appointed to maintain the software then these manuals will help them to understand how the whole system works.

5. Result and Discussion

The discussion section is where the article interprets the results to reach its major conclusions. This is also where the author's opinion enters the picture. The discussion is where the argument is made. Common features of the discussion.

Table 2. Comparison of different software development models

Model/Features	Scrum	Spiral	Agile Methodology: Hybrid Approach Scrum and XP	eXRUP: A Hybrid Software Development Model for Small to Medium Scale Projects	Hybrid Spiral Model to Improve Software Quality Using Knowledge Management	XSR: Novel Hybrid Software Development Model (Integrating XP, Scrum & RUP)	Novel Hybrid Model: Integrating Scrum and XP	IScrum: An Improved Scrum Process Model	ScrumSpiral: An Improved Hybrid Software Development Model Introduction
Requirement Prioritization	Medium	Medium	Low	Medium	High	High	High	Low	High
Team Meetings	High	High	Medium	Low	Low	High	High	High	High
Customer Involvement	Medium	High	High	Low	Medium	High	High	High	High
Cost Analysis	Medium	High	Low	High	Medium	Low	Low	Medium	High
Time Analysis	Medium	Low	Low	High	Medium	High	High	Medium	High
Project Size	High	Low	Medium	Medium	Low	Low	Low	Medium	High
Budget Estimation	High	Low	Medium	High	Low	High	Low	Low	High
Risk Analysis	Low	High	Low	High	Low	High	Low	Low	High
Flexibility	High	High	High	Low	Low	Low	Low	High	High
Iterative Approach	High	Low	High	Medium	Medium	Medium	Low	High	High
Testing	Medium	Medium	Medium	Medium	High	High	High	High	High
Documentation	Low	Medium	Low	Medium	High	Low	Medium	High	High
Maintenance	Medium	Medium	Low	Medium	Low	Low	Medium	High	High
Team Management	Low	Medium	Medium	Medium	Low	Low	High	High	High

This proposal introduced the idea of user stories following that chosen requirement from those stories. Prioritizing the requirements according to the customer's desires as well as categorizing them into high, medium, and low. After that, the requirements are passed through the development phase by passing the risk analysis, development, and customer evaluation section. Furthermore, the model integrates the whole system and after going through the testing phase (Alpha and Beta testing) it will be released to the customer. The maintenance will continue till the process ends as well as during the duration the customer wants to go with the system.

Here the hybrid model has further categorized the requirements after the product backlog list so that the development can be fast-forwarded. Risk analysis is done so that if the risk occurs in the future that can be handled efficiently. The development is flexible as its time span can be modified according to the project size and also can adapt to the new requirement given by the customer. The integration of the whole system helps to maintain the system

through development and customer evaluation. Two types of testing in system testing help to understand the developer's view along with general people who will use the system in the future. Maintenance helps to maintain the system during the whole procedure and after it is released to the customer. Here a team member assists in maintaining a document throughout the whole process. Finally, documentation and user manual are conserved to make it easier for new employees in the further future.

The proposed model needs a proper evaluative environment to improve and maintain proper management. The environment can be various industrial sectors as well as software companies that use numerous development models or environments to develop a project. By evaluating it in the professional environment its flaws and shortcomings can be identified that hope to be improved by using progressive ideas. The proposed model would take time, testing, and improvement to make itself a stand-alone model in the industry. After being tested this model's importance can be figured out in the development of a project. In the future, this paper wishes to add more features to improve its practical approach in the industry and expand its research to face difficulties in different situations. Various software development companies as well as professionals are working to improve the practices of Scrum and Spiral to make efficient use of their advantages to make a well-organized and structured project. In the upcoming time, it is wished to promote this advanced hybrid model. Furthermore, implementation practices in various projects can help to identify their limitations [21].

6. Conclusions

Our main goal of this research is to prepare a model that should be able to complete the project according to the expected schedule, satisfy customer requirements, and obtain productivity through team coordination. For this, there is a Hybrid software model that unifies the merits of Scrum and Spiral methodologies while trying to overcome their weaknesses [21]. There are a lot of software development models on the market: Waterfall, V-model, Scrum, Extreme Programming (XP), Rational Unified Process (RUP), Kanban, and so on. This research used those models for analysis of our model structure by comparing the advantages and disadvantages. As all of those models have benefits and drawbacks, according to the specific requirements it is very tough to select the finest model among all the other models. At first, by conducting extensive research in the software engineering field, a decision was made to make a hybrid model by using Scrum and Spiral, which is named "ScrumSpiral". Our designed hybrid model, "ScrumSpiral: An Improved Hybrid Software Development Model" is suitable for complicated projects and also for those projects whose requirements are not fixed initially. This can be said that this model is able to carry out all the requirements, goals, and objectives of the customer and finally, design a project that can satisfy the needs of clients.

During the implementation, some limitations are faced in designing this hybrid model. As this research combined two different models, Scrum and Spiral, they have different principles and working procedures. So, it was very difficult to combine them and make a new hybrid model in a limited timeframe. Due to our limited time and resources in our research, this model was not able to conduct the testing process in the industry to examine the usefulness and performance of this hybrid model. As well in first if the customer is not clear about their requirement and the team also does not confront that problem and makes a perspicuous requirement list then the whole project can undergo a complex and never-ending process.

In the future, if any other opportunity occurs in future, this hybrid model will be further enhanced and used in real-world software development projects to validate this "Scrum Spiral" model. It is the goal to enhance the model by testing it through different practical settings and development environments. Furthermore, the research and expansion of the model will be added to overcome other problem areas of the Scrum and spiral model [18].

References

- [1] Ralf Kneuper, "Sixty Years of Software Development Life Cycle Models", IEEE Annals of the History of Computing, Vol.39, No.3, pp.41-54, 2017. DOI:10.1109/MAHC.2017.3481346
- [2] Marco Kuhrmann, Philipp Diebold, Jürgen Münch, Paolo Tell, Vahid Garousi, Michael Felderer, Kitija Trektare, Fergal McCaffery, Oliver Linssen, Eckhart Hanser, Christian R. Prause, "Hybrid software and system development in practice: waterfall, scrum, and beyond", Proceedings of the 2017 international conference on software and system process, pp.30-39, 2017. DOI:10.1145/3084100.3084104
- [3] Shyam Gawade, Vidyavati Ramteke, "Assessment of the Impact of the Hybrid Software Development Approach", Journal of Physics: Conference Series, Vol. 1964, No. 4, 2021. OI:10.1088/1742-6596/1964/4/042009
- [4] Farrukh Musa, Muhammad Ali Tariq, "Agile Methodology: Hybrid Approach Scrum and XP", International Journal of Scientific & Engineering Research, Vol.08, No.4, pp.1405-1409, 2017.
- [5] Akanksha Mathur, Archana Acharya, "A Comparative Study on Utilization of Scrum and Spiral Software Development Methodologies: A Review" International Journal of Engineering Research & Technology (IJERT), Vol.4, No.11, 2015. DOI:10.17577/IJERTV4IS110154
- [6] Vanshika Rastogi, "Software development life cycle models-comparison, consequences", International Journal of Computer Science and Information Technologies, Vol.6, No.1, pp.168-172, 2015.
- [7] Predrag Matković, Pere Tumbas, "A comparative overview of the evolution of software development models", International Journal of Industrial Engineering and Management, Vol.1, No 4, pp.163-172, 2010. DOI:10.24867/IJEM-2010-4-019
- [8] Ken Schwaber, Jeff Sutherland, "The scrum guide", Scrum Alliance, Vol. 21, No. 1, pp.1-38, 2011.

- [9] M. A. Awad, "A comparison between agile and traditional software development methodologies", University of Western Australia, pp.1-69,2005.
- [10] Dhruv Doshi, Labdhi Jain, Kunj Gala, "Review of the spiral model and its applications", International Journal of Engineering Applied Sciences and Technology, Vol. 5, No. 2455-2143, pp.311-316, 2021. DOI:10.33564/IJEAST.2021.v05i12.053
- [11] Iqbal H. Sarker, Faisal Faruque, Ujjal Hossen, Atikur Rahman, "A survey of software development process models in software engineering", International Journal of Software Engineering and Its Applications, Vol.9, No.11, pp.55-70,2015. DOI:10.14257/ijseia.2015.9.11.05
- [12] M. Mahalakshmi, Mukund Sundararajan, "Traditional SDLC vs scrum methodology–a comparative study", International Journal of Emerging Technology and Advanced Engineering, Vol.3, No.6, pp.192-196, 2013.
- [13] Farrukh Musa, Muhammad Ali Tariq, "Agile Methodology: Hybrid Approach Scrum and XP", International Journal of Scientific & Engineering Research, Vol.8, No.4, pp.1405-1409, 2017.
- [14] Ghulam Rasool, Shabib Aftab, Shafiq Hussain, Detlef Streitferdt, "eXRUP: a hybrid software development model for small to medium scale projects", Journal of Software Engineering and Applications, Vol.6, No.9, 2013. DOI:10.4236/jsea.2013.69055
- [15] Brijendra Singh, Shikha Gautam, "Hybrid spiral model to improve software quality using knowledge management", International Journal of Performability Engineering, Vol.12, No.4, pp. 341-352, 2016. DOI:10.1109/CICN.2016.137
- [16] Gul Ahmad, Tariq Rahim Soomro, Mohammad Nawaz Brohi, "XSR: novel hybrid software development model (integrating XP, scrum & RUP)", International Journal of Soft Computing and Engineering (IJSCE), Vol.2, No.3, pp.126-130, 2014.
- [17] Zaigham Mushtaq, M. Rizwan Jameel Qureshi, "Novel Hybrid Model: Integrating Scrum and XP", International Journal of Information Technology and Computer Science, vol.4, no.6, pp.39-44, 2012.
- [18] Sara Ashraf, Shabib Aftab, "IScrum: An Improved Scrum Process Model", International Journal of Modern Education and Computer Science, Vol.9, No.8, pp.16-24, 2017.
- [19] Jeff Sutherland, Ken Schwaber, "The Scrum guide. the definitive guide to Scrum: The rules of the game", ScrumGuides.com, 2013.
- [20] Nabil Mohammed Ali Munassar, A. Govardhan, "A comparison between five models of software engineering", International Journal of Computer Science Issues, Vol.7, No.5, pp.94-101, 2010.
- [21] M. Salman Bashir, M. Rizwan Jameel Qureshi, "Hybrid software development approach for small to medium scale projects: RUP, XP & Scrum", Science international (Lahore), Vol.24, No.4, pp.381-384, 2012.

Authors' Profiles



Tapu Biswas is currently studying for her BSc In Computer Science & Engineering-Major in Information Systems at American International University-Bangladesh AIUB in Computer Science and Engineering. Her research interests are in software engineering, machine learning, data science, and data mining.



Farhan Sadik Ferdous is currently studying for his BSc In Computer Science & Engineering-Major in Information Systems at American International University-Bangladesh AIUB in Computer Science and Engineering. His research interests are in software engineering, machine learning, data science, and data mining.



Zinniya Taffannum Pritee is currently studying for her BSc In Computer Science & Engineering-Major in Software Engineering at American International University-Bangladesh AIUB in Computer Science and Engineering. Her research interest is in software testing and quality assurance, cybersecurity, and robotics.



Dr. Akinul Islam Jony is currently working as an Associate Professor & Head (UG) of Computer Science at American International University-Bangladesh (AIUB). His research interests include AI, e-Learning, machine learning, Cybersecurity, software engineering, and issues in data science.

How to cite this paper: Tapu Biswas, Farhan Sadik Ferdous, Zinniya Taffannum Pritee, Akinul Islam Jony, "ScrumSpiral: An Improved Hybrid Software Development Model", International Journal of Information Technology and Computer Science(IJITCS), Vol.16, No.2, pp.57-65, 2024. DOI:10.5815/ijitcs.2024.02.05