International Journal of Computers and Informatics (IJCI)

Vol. (2), No. (7)

VISION FOR SCIENTIFIC RESEARCH AND PUBLISHING
IJCI

November 2023

المجلة الدولية للحاسبات والمعلوماتية

الإصدار (2)، العدد (7)

"Improved Rational Unified Process (RUP) MODEL using Agile Features"

Raed Zayed Gummah
Master of Information Systems, Faculty of Computers and Artificial Intelligence, Ain Shams University, Egypt
razayedkas@yahoo.com

Abstract:

Communication plays a crucial role during the software development process using various models. However, it is not sufficiently practiced in some effective models, which impairs its performance. Diverse customized models used in the software industry such eXtreme Programming (XP) and Rational Unified Process (RUP).

Each of these models has advantages that distinguish it from the other and make it unique. Agile models such as XP is distinguished from others as 'RUP' by its focus on effective practical activities, including communication practices for better customer satisfaction and engineering practices. But the shortcomings in document-driven approach as RUP limits its capabilities, makes it not adapt to changing requirements, and does not provide best practices for simplicity. Communication between the user and the owner of the product with the work team is a valuable aspect of the software development process and should not be neglected.

The purpose of this paper is to focus on using communication practices that take place in XP model as one of the agile methodologies and how they can enhance performance of RUP model. In this paper, a new enhanced RUB framework will be proposed for better communication practices through the software development process.

E-ISSN 2976-9361
https://doi.org/10.59992/IJCI.2023.v2n7p1

Keywords: Rational Unified Process (RUP), eXtreme Programming (XP), Software Development Models, Agile.

Introduction

In The Software Industry, most software development companies seek to satisfy their clients by producing high-quality software at the lowest cost in the shortest possible time. Management of the development process and proper planning by choosing the right methodology is crucial to achieve this goal. Several software development methodologies that propose different ways such as Rational Unified Process (RUP,and eXtreme Programming (XP).

Those methodologies establish the framework that structures planning, developing software systems and controlling. The Rational Unified Process (RUP) is an incremental and plan oriented architectural framework, focus on structured through step by step process methodology to promote qualitative object oriented software projects. RUP is a plan driven approach and conventional, which presents a clear formalized flow and structured for software development. Document driven approaches for software development provide a very structured and formalized way for stable requirements [1].

In some projects requirements are ambiguous and unclear in this situation agile methodologies is the best practice. Rational unified process methodology is considered a modern approach based on web-enabled program development and object-oriented. It works through splitting the workflow to four different parts. Those parts are namely inception, elaboration, construction and transition. Conventional software development methods have gradually been replaced by lightweight agile software development methods since the mid-1990s [2].

Agile is a popular and widely adopted software development technique. One of principles behind the Agile, at regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly [3].

Agile methodology is the most used and famous process model. It is a trend to promote efficiency of software products with high of customer satisfaction [4]. Practices that the teams have not employed, such as, pair programming and 143 user stories, and discuss the rationale for this from the teams' perspective. Agile technique may be characterized by a number of agile practices as well, such as pair programming in XP [5].

XP is lightweight and simple agile methodology for simple projects and small scale. XP is designed for the small team's work who need to work in a quick and fast software development environment where the requirements are changing exceptionally and frequently. XP functions by bringing the whole team to work together in the presence of clear simple practices, with continuous feedback to allow them to see position each other.

XP model has twelve core practices, which are as pair programming, planning game, collective ownership, on-site customer, 40 h's week, continuous integration, simple design, metaphor, refactoring, small releases, open workspace and just rules and tests. Feedback plays a very important role in all levels of XP model. Constant feedback and evaluation by the user is the major feature of XP [6]. The study shows that agile practices improve both informal and formal communication [7].

RELATED WORKS

Traditional software development methods differ from agile software development, thus that is reflected in its application during the software development process. Since the mid-nineties, the traditional software development methods have been replaced gradually by the lightweight agile software development methods.

Fundamentally, this phenomenon is due to there are shortcomings in the conventional methods including delays in schedule, over budget and not adequately adapting to changing work requirements.

In 2009, Juyun illustrated some characteristics strengths and weaknesses of both conventional and agile methods [2]. Various customized models are used in software industry to develop software projects such as Rational Unified Process (RUP), SCRUM and eXtreme Programming (XP). Scrum and XP are most practiced agile models which have a good performance for small projects. RUP model is more suitable for large projects. Salman and Jameel proposed a new integrated model to deal with small, median and large projects [1].

The agile software development is achieving benefits in reducing the cost during implement and test software development process. Security is a major issue to adapt cloud in agile industries. The research dealt with a discussion of some limitations in the practices of agile models and their impact on the security of customer data (Butt, Jamal and Arshad, 2020) [4].

Kuppuswami, Vivekanandan, Ramaswamy & Rodrigues 2003 stated traditional software development methodologies are heavyweight, rigid and oriented to documentation. This make it hard to follow in the present E-Business. Extreme Programming (XP) has a few simple rules on less documentation [11]. The increasing demand for information systems with the development of technology creates a competitive environment and pushes companies to focus on customer needs to gain more competitive advantage.

Agile models tend to focus on improving the practical implementation of software development to meet user requirements. This causes a lot of big challenges during the software development implementation process (Sohaib, Solanki, Dhaliwa, Hussain& Asif, 2019) [6].

Using Rational Unified Process (RUP) to develop a software application system for android based that can help user to navigate indoor using mobiles devices. RUP is a complete methodology with an emphasis on accurate documentation [8].

Pressure is increasing in software industry field to development efforts that could help to adopt agile software development to respond rapidly for changing the requirements. Safety-critical system developments need usually a high regulated, that may causes constrain adopting the agile software development and require to select accuracy practices and methods [5].

Pikkarainen, Haikara, Salo, Abrahamsson & Stil, 2008 mentioned a little who is knowing about the effect of communication in software development process as a resulting to apply the agile practices. To face the challenges of business volatile environments eXtreme Programming (XP) and SCRUM increasingly have been adopted [7].

XP is an agile methodology that depends on face-to-face and in an informal communication. Layman a,Williams, Damian and Bures suggested that informal communication can use on a global software development systems in case the critical enabling factors are dealt and addressed [9].

Juha and Abrahamsson pointed out that Extreme programming (XP) is similar the other agile approaches in terms value the collaboration with customers on-site customer is one of XP practices that suggest customer should be available 100% during software development process. Role of XP customer is difficult, demanding and costly [10].

Refactoring helps to increase code quality and make it easier to modify, maintain and understand. By assess the effect of refactoring phase in a close-to industrial environment, results point out that refactoring improves productivity and increases aspects of software quality [12].

Author's (s) name	Publication year	Notes
Juyun Cho	2009	The research focused on the comparison between conventional and agile models in software development and show characteristics, strengths and weaknesses for each. Not mentioned the possibility of merging some features with each other.
M. Salman Bashir M. Rizwan Jameel Qureshi	2012	Proposing a new model is a combination of three models for software development (RUP, XP and Scrum) and did not explain the mechanism of work of the model.
Aziz Butt Tauseef Jamal Arshad Ali	2020	The authors discussed the components of agile methodology, its benefits and drawbacks but no practical solutions have been proposed to reduce the flaws and what threatens its effective.
S. Kuppuswami, K. Vivekanandan, Prakash Ramaswamy, Paul Rodrigues.	2003	Research focus on individual use of XP model practices but ignoring the advantages that traditional models can provide.
Sohaib, Solanki, Dhaliwa, Hussain & Muhammad Asif	2019	The researchers did not focus on what XP practices can be used to satisfy and meet the user's needs.

Asmaa H. Thannon;
Sahar Esmaiel Mahmood;
Falah Y.H. Ahmed
	2020	The researchers did not elaborate on how to deal with changing requirements during the software development process and what are the challenges that can be faced in the absence of an iterative approach.

Islam and Storer	2020	The study lacks further specification and address the importance of developing communication practices in software development models.

Author's (s) name	Publication year	Notes
M. Pikkarainen, J. Haikara, O. Salo, P. Abrahamsson & J. Stil	2008	The research did not address the use of traditional software development methods and their advantages during the development process.
Layman a, Williams, Daniela & Bures	2006	The research focus on applying practices of XP model and they didn't address the possibility of integrating those practices into a more specific and structure framework such as RUP. Didn't mention how a fine combination can work in order to achieve high quality software and enhance the team communication.
Juha Koskela and Pekka Abrahamsson	2004	The research focused on the practice of only one of the practices of XP 'on-site customer' and did not address the importance of the other XP practices that may support communication during the software development process.
Raimund Moser, Pekka Abrahamsson, Witold Pedrycz, Alberto Sillitti and Giancarlo Succi	2007	The study focused on refactoring phase and its impact on product quality during the software development process. The study did not address practices that can be applied at each phase to improve the product development process.

The Proposed Framework

Two methodologies on which this proposed framework is based on, the enhanced RUP model and Extreme Programming (XP) practices. The main methodology is the enhanced RUP model which is integrated with XP communication practices to ensure that feedback is received without delay during most parts of the software development process.

Four major phases of the Rational Unified Process Inception phase, Elaboration phase, Construction phase and Transition phase [8]. Inception: defines the project scope and develops a business case. Elaboration: specify features, Plan project and baseline the architecture. Construction: Build the product. Transition final version of the product to its users. XP Approach is based on six phases and twelve practices [9].

See Figure (1) below, the four main phases of RUP have now been renamed, enhanced and restructured' into five more specific major phases to integrate with the most effective XP communication practices. These five phases are present as methodical and structured methods to software development. The five main phases of the enhanced RUP model are defining and analyzing phase' as the first phase, 'Design' coming as a second phase, 'implementation' is the third phase, Evaluation is the four phase and the final phase is 'Release' the product. Each phase but Define Analysis and Release is done in several iterations. During applied these five phases, most important communication XP practices methodology will be held to make a more efficiency process and to provide simplicity.

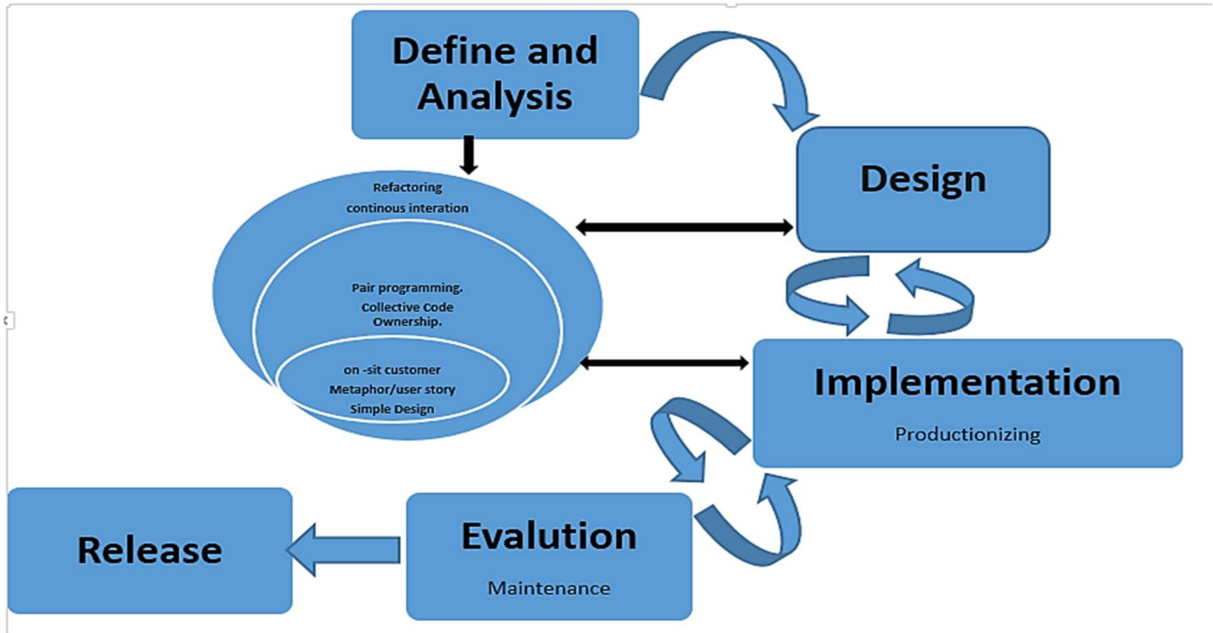

Figure (1): The Proposed Framework

Not all XP practices will be applied throughout all phases of this new hybrid model, but some practices that promote communication and allow to get valuable feedback will be identified. Moreover, not all XP communication practices will apply to all phases of proposed hybrid model 'enhanced RUP model' but each of them will be determined according to the nature of phase. Communication practices helps to a software provider & vendors to compete with others to achieve the market value.

XP's on-site customer practice is one of the XP practices suggests that the customer should be 100% available for the development team that may achieve values close collaboration with customers. Using this practice allows the possibility to customer needs/ requirements are reexamined all the time, answer questions and resolve problems. Questions are answered by a real person who will use the system. A real, live user on the team, available full-time to answer questions [10].

An Extreme Programming system metaphor is a practice in which XP developers replace the standard project architecture used in traditional software development methodologies. User stories are one of the essential development artifacts for Extreme Programming (XP) project teams. A user story is a very deep and high-level definition of a requirement, containing just enough input so the developers can produce a reasonable estimate of the effort to implement it.

Pair programming is an agile technique which coming of Extreme programming (XP) in which two developers of the team work together on one computer. Two team members work together to design, code and test user stories [11]. Refactoring is a technique to upgrade code without doing changes in functionality. Refactoring is an ongoing process of simplification that applies to testing, design, code and XP itself. In XP model the developers will be refactoring during the entire process of software development. [12].

Continuous Integration (CI) is an eXtreme Programming (XP) practice in which the members of a delivery team considerably integrate their work (e.g. least once daily or hourly). Each integration is proving through an automated build in which also performs testing to detect any integration errors automatically and quickly [13].

Define the requirements and system analysis is the main root phase for this proposed integrated framework RUP-XS. Deep and Intensive System Analysis has to conduct to define the desired requirements. This should be done by applying parallel with XP best practices. Using XP practices that create a fully-functional product. Customer’s involvement and interaction are the main parts of the production process. There is a focus on practical planning and create detailed task scheduling.

With the result of finding and analysis phase of the project the metaphor and simple design practices will make project justification, in this phase, the significant of the project is strengthening, where the scope of the user will be taken in consideration and what risk does the project might be facing, such as cost, time or hardware [14] . In this critical phase it is important that the developer knows what he will build.

First Phase:

Define and Analysis' begins through frequent meetings/ interviews between product owner or user and work team in which thorough comprehensive understanding of the requirements need to be achieved in this phase [15]. As a result of using XP communication practices as agile technique the continuous improvement, adapted planning, evolutionary development, early delivery and encourages make rapid and flexible response to change will be more available. Understanding the scope and objectives of the project in a detailed analytical way is the main goal at that stage [16].

After each meeting with project owners, team work and project manager are held other meeting to examine requirements in a more analytical way and the possibility extent of its application from a technical point of view. In this meeting effective XP practices can be use according to the nature of each project, such as brainstorming, metaphor and story cards. Define and Analysis phase is a crucial step in software development. The software development work team has to focus on this stage a lot, so they can get it right from the first time. During this crucial software development stage, developers spend time to learn project business processes, technical ecosystem, pain points, goals and challenges [17].

Once that information is gathered, validate goals to present the scope of work. The outcome of this phase will be documented, that is the general core of the project requirement, key features and main constraints. The first phase 'Define and Analysis' is the longest phase in this novel framework for the software development process. This phase ends when a clear and deep view of the product analysis and detailed requirements are being obtained. The deliverable is documented to describe the market and the reason behind the product/project.

Second Phase:

'Design' cannot start before the first phase is substantially completed Thus, software requirements can be re-evaluated further in the design phase. In this phase, the team focuses on establishing the system's architecture and requirements. It's time to map out workflows, determines what database will be used, and create a data model. There is able to see and try software to the last development stage is finished, which results in low project risks and predictable project results. In the design phase, the development team determines what software needs, how it will look, and what the timeline for development is going to be. Good design in software development is extremely important in SCLC [18]. Something that is beautifully designed, user friendly, and bug free, product owners need a tool that will actually produce a return on investment. Given that usability is one of the most important aspects of software quality [19]. This stage is done through some agile practices related to communication that help to provide a high-quality product in the least possible time such as pair programming, continuous integration and sprint meeting. Once we have all the requirements and designs documented and mapped out, it's time to start the code writing. This phase a particularly critical phase for the project where this phase signifies move from lower risk to high risk, where the actual coding will be taking place in the next phase [20]. The product architecture will be developed to establish the baseline for the architecture to produce an accurate schedule, mitigate essential cost and risk estimates to refine and detail high level project. The class diagram will be created in this phase as a deliverable that will describe the various system modules, allowing developer and stakeholder to understand the map dependencies between the modules and system services. The deliverable is the class diagram allowing map dependencies.

Third Phase:

Third phase is implementation where construction complete of the product interface and develop used aids by implementing work plans and design. This phase is when the coding and implementation of all application features will take place. Using tools and the software to achieve features and coding during the software development process and create user interface while taking consideration a simple design that we have planned for at the previous phases. During the implementation phase available and existing technological capabilities is analyzing to meet user requirement, to find the best possible and as much as possible less risk way to build the product. Project construction and coding is the most important phase in the software development process [21]. The communication practice of the XP model will help reduce the potential risks. Pair programming is one of the agile technique radix of Extreme programming (XP), where two developers of team work working together on one computer. This phase is where integration processes with the other services or the existing software should occur as well. The two people work together to design and code even to test user stories. Ideally, two team members each would be have equal time at the keyboard and equally and equality skilled. Knowing existing technology to build the system to meet user requirements an important aspect to meet user requirements. The majority of the actual coding takes place, implementing and organizing all code coming into layers that make up the whole of the system. The end of this phase is measured by the completion of the Initial Operational Capability Milestone, which is based on some criteria: Is this product release stable and mature enough to be deployed in the user community? Are the stakeholders ready to the transition into the user community?

Evaluation Phase, The Fourth Phase:

where testing of all kinds takes place of the workflow. The evaluation and testing process starts once the coding is complete and the modules are ready to release. In this phase, the developed software is tested thoroughly and any defects found are assigned for development to get them fixed. Regression testing and retesting are done to the point at which the software is as per the customer’s expectation and meet the requirements. At the end of this phase tester indicate a document to make sure that the software is as per customer’s standard requirements. UAT 'User Acceptance testing' is done based on customer expectation. Continuous integration as one of XP practices that can used in this phase [22].

Last Phase:

The last phase in this novel proposed framework is release the product, the deployment workflow constitutes the entire delivery and release process, ensuring that the software reaches the customer as expected. Refactoring practice as one of XP model practices can be effective tool to be used in this phase, it can also be used in the evaluation or testing phase. Once the product has been tested, it is ready to deploy in the production environment. In the case of user Acceptance testing a replica of the system/product environment is created the developers along with the customer does the testing. If users find the system as expected so the sign off is provided by user to go live. This phase is when the finished product is final released and delivered to customers. However, the release phase is more than just the process of deployment but it has to handle all post release support, patches, and bug fixes and so forth as well [23]. This phase is based on a few simple questions: Are planned resource expenditures versus the actual expenditures still acceptable? Is the product owners/ users satisfied? Maintenance of the product i.e. if any enhanced or any issue comes up and needs to fix is to be done is taken care by the developers.

Conclusion

Various models in the software industry are using such as Rational Unified Process (RUP) and eXtreme Programming (XP). RUP is structured model coming in specific phases. RUP is document-driven approach for the software development process provides a structured process for the software development through its deep planning and analysis. RUP does not adapt to changing requirements and is more suitable for fixed requirements during the software development process. It doesn't give the best engineering practices in order to achieve simplicity, reliability and quick adaptation to changing requirements. Agile process models follow an iterative approach, focus on customer and team collaboration through small loops/iterations. Agile methodologies feature strong practices that meet customer satisfaction. This is due to the practice of various activities, especially communication activities that make it able to interact well with changing requirements. XP and are widely practiced agile models. XP model provides best engineering practices which that improve communication practices but there is a lack of formalized and structures nature. Engineering practices such as pair programming, test driven development and user stories. XP provides practical techniques to track and manage the software development but there is a lack of specific structured phases and mechanism of communication practice. This paper focuses on highlighting some of the strengths and characteristics of XP model to use it in RUP model. A modified RUP model is required to improve communication and thus reduce the proportion of changing requirements. RUP-XS is a modified of the rational unified process model integrated with XP practices features. New framework is proposed to enhance communication during the software development process to achieve high quality software and enhanced the team productivity. This communication improves performance and reduces the possibility of frequently changing requirements and meet customer expectations.

Future Work

This proposed framework 'RUP-XS' need to be applied on a practical experience as case study to see the extent of its application and effectiveness in the software development process.

References

[1] Bashir, M. S., & Qureshi, M. R. J. (2012). Hybrid software development approach for small to medium scale projects: RUP, XP & Scrum. *Cell*, *966*, 536474921.

[2] Cho, J. (2009). A hybrid software development method for large-scale projects: rational unified process with scrum. *Issues in Information Systems*, *10*(2), 340-348.

[3] Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., ... & Thomas, D. (2001). Manifesto for agile software development.

[4] Gochhait, S., Butt, S. A., Jamal, T., & Ali, A. (2020). Cloud enhances agile software development. In *Cloud Computing Applications and Techniques for E-Commerce* (pp. 28-49). IGI Global.

[5] Islam, G., & Storer, T. (2020). A case study of agile software development for safety-Critical systems projects. *Reliability Engineering & System Safety*, *200*, 106954.

[6] Sohaib, O., Solanki, H., Dhaliwa, N., Hussain, W., & Asif, M. (2019). Integrating design thinking into extreme programming. *Journal of Ambient Intelligence and Humanized Computing*, *10*(6), 2485-2492.

[7] Pikkarainen, M., Haikara, J., Salo, O., Abrahamsson, P., & Still, J. (2008). The impact of agile practices on communication in software development. *Empirical Software Engineering*, *13*(3), 303-337.

[8] Thannon, A. H., Mahmood, S. E., & Ahmed, F. Y. (2020, April). Using Rational Unified Process in Indoor Navigation application for Android Smart Phones. In *2020 IEEE 10th Symposium on Computer Applications & Industrial Electronics (ISCAIE)* (pp. 245-250). IEEE.

[9] Layman, L., Williams, L., Damian, D., & Bures, H. (2006). Essential communication practices for Extreme Programming in a global software development team. *Information and software technology*, *48*(9), 781-794.

[10] Koskela, J., & Abrahamsson, P. (2004, November). On-site customer in an XP project: Empirical results from a case study. In *European Conference on Software Process Improvement* (pp. 1-11). Springer, Berlin, Heidelberg.

[11] Kuppuswami, S., Vivekanandan, K., Ramaswamy, P., & Rodrigues, P. (2003). The effects of individual XP practices on software development effort. *ACM SIGSOFT Software Engineering Notes*, *28*(6), 6-6.

[12] Moser, R., Abrahamsson, P., Pedrycz, W., Sillitti, A., & Succi, G. (2007, October). A case study on the impact of refactoring on quality and productivity in an agile team. In *IFIP Central and East European Conference on Software Engineering Techniques* (pp. 252-266). Springer, Berlin, Heidelberg.

[13] Hummel, M., Rosenkranz, C., & Holten, R. (2013). The role of communication in agile systems development. *Business & Information Systems Engineering*, *5*(5), 343-355.

[14] Kruchten, P. (2004). *The rational unified process: an introduction*. Addison-Wesley Professional.

[15] Ullah, H., Hussain, T., Khan, G. F., & Alnafjan, K. (2012). A comparative analysis of XP, RUP and DSDM Techniques. *Science Series Data Report*, *4*(5).

[16] Kroll, P., & Kruchten, P. (2003). *The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP: A Practitioner's Guide to the RUP*. Addison-Wesley Professional.

[17] Rasool, G., Aftab, S., Hussain, S., & Streitferdt, D. (2013). eXRUP: A Hybrid Software Development Model for Small to Medium Scale Projects.

[18] Ahmad, G., Soomro, T. R., & Brohi, M. N. (2014). XSR: Novel Hybrid Software Development Model (Integrating XP, Scrum & RUP). International Journal of Soft Computing and Engineering (IJSCE), 2(3), 126-130.

[19] Paz, F., & Pow-Sang, J. A. (2016). A systematic mapping review of usability evaluation methods for software development process. International Journal of Software Engineering and Its Applications, 10(1), 165-178.

[20] Anwar, A. (2014). A review of rup (rational unified process). International Journal of Software Engineering (IJSE), 5(2), 12-19.

[21] Hanssen, G. K., Westerheim, H., & Bjørnson, F. O. (2005, November). Using rational unified process in an SME–a case study. In European Conference on Software Process Improvement (pp. 142-150). Springer, Berlin, Heidelberg.

[22] Shylesh, S. (2017). A study of software development life cycle process models. In National Conference on Reinventing Opportunities in Management, IT, and Social Sciences (pp. 534-541).

[23] Ruparelia, N. B. (2010). Software development lifecycle models. ACM SIGSOFT Software Engineering Notes, 35(3), 8-13.