Latest Transformations of XP Process Model: A Systematic Literature Review

Sadia Khan[1], Muhammad Abuzar Fahiem[2], Birra Bakhtawar[1], Shabib Aftab[1,3], Munir Ahmad[3], Nauman Aziz[4], Abdullah Almotilag[5], Nouh Sabri Elmitwally[5,6]

[1]Department of Computer Science, Virtual University of Pakistan, Lahore, Pakistan
[2]Department of Computer Science, Lahore College for Women University, Lahore, Pakistan
[3]School of Computer Science, National College of Business Administration & Economics, Lahore, Pakistan
[4]Department of Computer Science, Superior University, Lahore, Pakistan
[5]College of Computer and Information Sciences, Jouf University, Sakaka, 72341, Saudi Arabia
[6]Department of Computer Science, Faculty of Computers and Artificial Intelligence, Cairo University, 12613, Egypt.

Abstract — Process model is an integral part of software industry. Different process models are used now a days in the industry for different software projects. Process models need to be tailored to address some specific project needs. Agile models are considered as the most widely used process models nowadays. They have distinctive features and the ability to address the dynamic needs of today's software development. Extreme programming (XP) is one of the extensively used agile process model especially for small projects. Many researchers have tried to mold XP to overcome its shortcomings and for better working in specific scenarios. Therefore, many customized versions of XP process model are available today. In this paper, we are going to analyze the latest customizations of XP. For this purpose, a systematic literature review is conducted on studies published from 2012 till 2018 in renowned online search libraries. This comprehensive review highlights the purpose of customizations, along with the areas in which customizations are made, and phases & practices which are being customized. This work will serve the researchers to discover the modern versions of XP process model as well as will provide a baseline for future directions for customizations.

Keywords — Agile, Extreme Programming, XP, Modified XP, Customized XP, Systematic Literature Review, SLR.

1. INTRODUCTION

Conventional models gradually have been replaced by the agile models in software industry due to features that agile family provides [32], [33], [36], [37]. These features address the needs of modern software development. In agile models, researchers and software practitioners merged the top practices of software engineering and targeted to overcome the shortcomings of conventional software process models. Agile models provide fast development track to deliver high quality software [10]. Agile models have iterative and incremental nature and customer needs are satisfied through early and continuous delivery of partially working software [38], [39], [40], [41], [42]. Customer's changing requirements can also be tackled at any stage of development. Extreme Programming, Feature Driven Development, Scrum, and Dynamic System Development Method are some of the renowned agile models. Extreme programming (XP) is one of the most widely used agile process models in the software industry particularly for small scale projects. Uncertain and changing requirements are handled in an effective way to attain customer satisfaction.

XP has some advantages over the traditional models including:

- It can easily adapt to frequently changing requirements
- It produces a high quality software in less time and cost
- It helps small firms for the implementation of software process improvement strategy.
- It includes some of the best practices like pair programming, on-site customer, collective code ownership, continuous integration and continuous testing

The practices of XP such as pair programming, collective code ownership, on-site customer, continuous testing and continuous integration were new for the software industry, but their adequate outcomes forced the developers to adopt them in modern projects [14], [30], [31], [34], [35]. XP uses twelve best practices of software engineering. These practices include: planning game, small release, metaphor, simple design, continuous testing, refactoring, pair programming, collective ownership, continuous integration, 40-hour week, on-site customer, and coding standards [11]. Researchers have customized XP for various scenarios. They have tried to modify XP to make it appropriate in different situations by customizing its phases or by adding more practices for particular needs. Therefore, many customized versions of XP are available nowadays. To reflect the latest research, this study carries out a systematic literature review to investigate the latest

Manuscript received June 5, 2021
Manuscript revised June 20, 2021
https://doi.org/10.22937/IJCSNS.2021.21.6.19

customizations of XP published from the year 2012 till 2018. Related literature is considered in this SLR by applying inclusion and exclusion criteria. Literature extraction is done from renowned online search libraries.

2. RELATED WORK

In [12], a systematic literature review is carried out to investigate the method of requirements engineering in agile models. The study mainly focuses on different methods that are used to inquire stakeholders during requirements engineering and management procedure. In [13], a systematic literature review was conducted to find the outcome of agile release practices on software projects. They are utilized for low cost and fast development. In [14], a systematic literature review is conducted on researches done from 2013 to 2017 to evaluate the latest customized models of XP. In this review the reason of customization, customizations types, and the practices and phases being customized are identified. In [15], a systematic literature review is carried out to figure out the general themes in agile maturity model research. It is concluded that there is an increase in agile maturity model researches in the second half of the last decade. In [16], an SLR was conducted to figure out the challenges and success factors in agile method's conversion for large projects. Researchers have identified 35 challenges and 29 success factors on which the chosen papers are evaluated. In [17], researchers focused on the trend of merging agile practices with out-contracted software development. Synchronization and communication problems were being faced by this sort of development. The focus of this study is to find out useful collaboration practices. It also differentiates these practices from the traditional practices being used in non-distributed scenarios. In [18], researchers have chosen literature from 2002 – 2013 to gather data related to requirements engineering practices and disputes in agile models. In [19], researchers have given various aspects for the customizations of agile models. They have identified the basis used for selecting practices for modifications. The research published from 2002 to 2014 is used in this SLR to find out the usual basis used for customization. In [20], an SLR is carried out to identify the techniques used for effort estimation in agile development. According to the researchers, use case, expert judgment, and planning poker techniques are the most utilized methods in agile processes. An SLR is conducted in [21] to figure out the security related issues in development using extreme programming. The literature published from 2002 to 2012 is used by the authors. The authors come to a conclusion that the security based practices can be combined with XP to get optimal outcomes. In [22], literature related to merging user centered design (UCD) with agile models is studied. The authors conclude that this mergence is utilized for design and usability analysis, but there are less studies on integration of UCD and agile that give empirical proof. In [23], a meta-analysis is carried out to figure out the effect of pair programming on effort and quality. The results show that pair programming has bad effect on effort used in development, and it performs better in quality. To find out the outcomes of pair programming, another SLR is conducted in [24]. Such factors are identified that have an effect on the utility of pair programming. In [25], a systematic literature review is conducted to know the effect of user involvement in the success of project. 87 empirical studies are evaluated in and it is concluded that an affirmative function is played by user involvement in the success of a project.

3. RESEACH PROTOCOL

A detailed research methodology with step by step directions is needed for a high quality systematic literature review that may assist in getting the objectives of the research in true spirit. A systematic literature review is one which provides a compact information on the particular research topic for given time period. Various studies elaborate the guidelines for SLR including: [26], [27], [28], [29], [43], [44], [45]. In general, an SLR has three steps: plan review, conduct review and document review. The steps included in the research methodology are: 1) Defining research questions, 2) Keywords finding for the query string, 3) Identifying research space for getting the data, 4) Criteria setting for including or excluding papers, 5) Extracting the literature on the basis of criteria, 6) Assessing the quality of study, 7) Synthesizing the data, and lastly 8) documenting the outcomes and results (Fig. 1).

3.1. Defining Research Questions:

The objectives of a systematic literature review is represented by the research questions. The answers to these questions help in concluding the SLR. The research questions which will cover our research purposes are following.

RQ1: What customizations/modifications are proposed by the researchers?

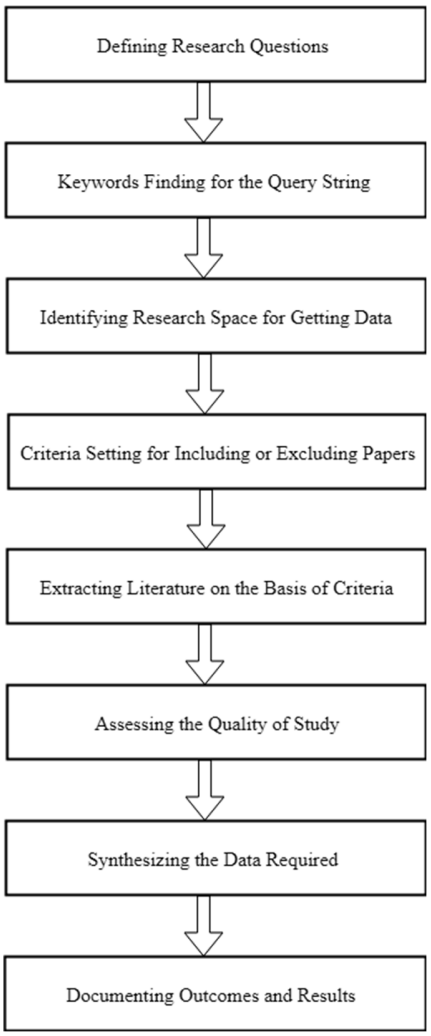

Fig. 1. Systematic Literature Review Steps

RQ2: Which phases, practices, or quality characteristics are added in XP?

RQ3: Which practices or phases are removed or modified in XP?

RQ4: What purposes are achieved by modifying XP?

RQ5: How are the modified versions evaluated?

3.2. Keywords Finding for the Query String

The step 2 of our research strategy is to make the query string. For that, we have extracted keywords from research questions, and then have assembled them in sequence to shape it into a query. Following keywords are extracted from research questions for query:

"Extreme Programming", "XP", "Agile", "Version", "Modified", "Changed", "Customized" "Tailored", "Improved", "Updated", "Enhanced", "Quality", "Process", "Model", "Lifecycle", "Method", "Procedure", "Practices".

Following is the query string:

(Modified OR Changed OR Tailored OR Customized OR Optimized OR Enhanced OR Improved OR Updated) AND (Agile OR (Extreme Programming OR XP) AND (Process OR Model OR Lifecycle OR Method OR Procedure).

3.3. Identifying Research Space for Getting the Data:

Search space signifies the collections and repositories from where we can collect the data. We have selected well known online libraries to find the papers from 2012 to 2018: IEEE, Elsevier, Science Direct, Springer, and Google Scholar. The data was extracted through the developed query string. Some adjustments were done in the query string to extract the appropriate literature as all these libraries had different options to extract the material. We searched the query multiple times with different combinations. Table 1 shows the search queries results.

TABLE 1 SEARCH SPACE AND EXTRACTED RESULTS

Sr. No	Digital Library	Total Results
1	Elsevier	1073
2	Science Direct	1658
3	IEEE	586
4	Springer	2883
5	Google Scholar	9660

3.4. Criteria Setting For Including or Excluding Papers

The inclusion and exclusion criteria defined here is used to select the related material. Inclusion criteria is represented by IC, and exclusion criteria is represented by EC.

3.4.1 Inclusion Criteria

The inclusion criteria has the following rules to retrieve the relevant material for the SLR:

IC1: Papers that are published from 2012 to 2018.

IC2: Papers that are available in journals, conferences, or proceedings of conferences.

IC3: Papers that have modified the phases of XP.

IC4: Papers that have customized the practices of XP.

IC5: Papers that introduce software quality improvement practices in XP.

IC6: Papers that have proposed modified version of XP with figure.

3.4.2. Exclusion Criteria

The exclusion criteria has the following rules to retrieve the relevant material for the SLR:

EC1: Papers that are not published from 2012 to 2018.

EC2: Papers that are not written in English language.

EC3: Text that is part of any book.

EC4: Papers that create a hybrid model by combining XP with some other process model.

EC5: Papers that do not give pictorial representation of the proposed model.

EC6: Papers which propose application of XP in the field other than software engineering.

EC7: Literature that is under review or is a part of thesis report.

3.5. Extracting the Literature

The most related literature is selected for the review through the selection criteria. We selected 9 articles after applying the inclusion and exclusion criteria. Fig. 2 shows the overall procedure of literature gathering.

3.6. Assessing the Quality of Study

For a good literature review, quality is very essential. We have completed every step under the umbrella of particular quality parameters including:

- Complete literature is chosen from reliable and well-known libraries.

- Only reputable journals publications are included in the SLR.

- There is no discrimination in collecting the literature.

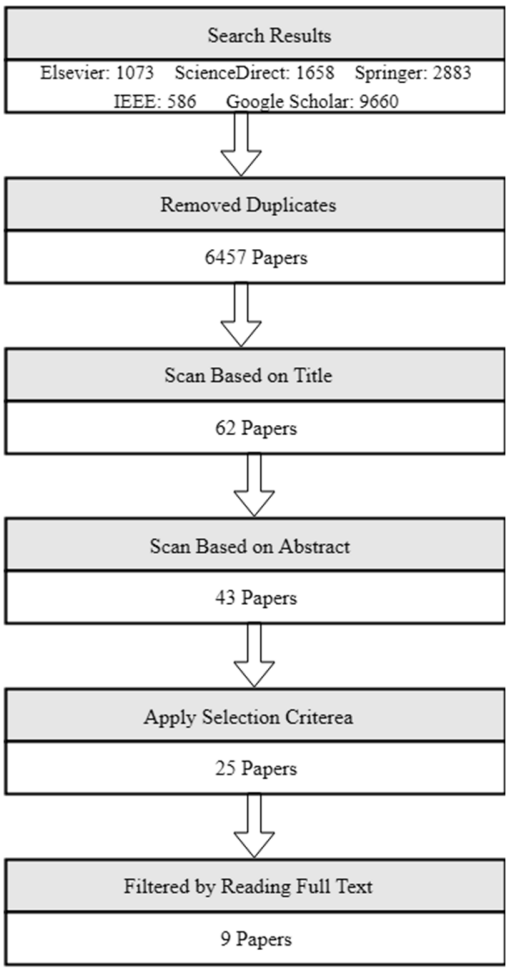

Fig. 2. Search Process

4. DATA ANALYSIS

Nine papers are shortlisted after applying the extraction criteria given in Section III. Following answers to the research questions are extracted after the comprehensive analysis and review of the selected papers.

RQ1: What customizations/modifications are proposed by the researchers?

In [1] the researchers improved the extreme practices of XP. Three most criticized and extreme practices: lightweight requirement, pair programming, and onsite customer are modified in this research. In [2] the XP framework is improved by adding user related security processes. The research done in [3] overcomes the problems of XP such as lack of documentation, and controversial practices like pair programming and on-site customer. In [4], the researchers state that in some situations the extreme practices of XP such as on-site

customer, continuous testing and continuous integration overburden the software development process. Especially in small scale projects these practices cause extra effort and delay. XP is modified to overcome the problems caused by these practices that can be a hurdle in timely completion of small projects. In [5] the researchers have proposed a quality model for extreme programming by adding quality attributes in the phases of XP model. It ensures that the product will contain quality attributes as desired by the stakeholders and existing in ISO/IEC (Quality standards, ISO 9126, ISO 25000). In [6], the researchers have modified XP by adding component based structure reusability. In [7], XP is modified to achieve better agility, documentation, and good architectural design. Agility of XP is affected because of sequential testing and refactoring during iterations, and it results in less time for documentation and a straight architectural design. This makes XP suitable for small projects only. So it modifies sequential testing and refactoring in XP. In [8], XP is modified to make it suitable for medium scale projects that can handle big development teams, as XP is suitable mostly for small teams. XP limitations like lack of design and less documentation are covered in the modified model, and project failure risk is also managed. In [9], XP is modified to make it suitable for medium and large scale projects. Limitations of XP like poor design, lack of documentation, lack of risk management, and poor architecture are covered that limit its suitability to small scale projects. In this research, XP is modified for security critical projects.

RQ2: Which phases, practices, or quality characteristics are added in XP?

In [1], the three most criticized and extreme practices of XP including lightweight requirement, pair programming, and onsite customer are modified in order to improve the life cycle. The study states that to improve the requirements in XP, use cases should be collected from scenario based requirement engineering and then stakeholder analysis should be done. To improve the pair programming problem personal development traits, Distributed Pair Programming and Collaborative Adversarial Pair Programming models are recommended. Alternatives like surrogate customers and multiple customer models are suggested to resolve the onsite customer problem in XP. In [2] the processes of authentication and authorization are used to meet the security goal of confidentiality, integrity, and authorization. First the user whose story is to be taken for incremental delivery of the product is asked for authentication. Then the user story is opted for advance development. The second security step of authorization is taken when the user evaluate the system. The modified version of XP proposed in [3] addresses architecture, design, and documentation issues. The proposed SXP model includes analysis and design step, and documentation is produced in each phase. In [5], quality attributes including availability, efficiency, usability, testability, flexibility, portability, and maintainability are added in XP. The framework proposed in [6] adds components of reusability practices in XP, and performs refinement and refactoring. For this purpose, it includes the steps of component search and retrieval, identify components to extend and refine, generate target components, and repository management. The model proposed in [7] adds parallel improvement iterations along with the development iterations in XP. In [8], analysis and risk management phase is added in XP to defeat project failure risk. The model proposed in [9] adds analysis and risk management phase in XP, and also includes security checks in all the phases in XP.

RQ3: Which practices or phases are removed or modified in XP?

In [1] the practice of onsite customer is removed from XP and its alternatives are suggested. Instead of the pair programming practice, distributed pair programming and collaborative adversarial pair programming models are suggested. The research done in [2] modifies the steps of taking user stories, and also changes the steps in which the developed software is validated by user. It adds security processes in these steps. The modified version of XP proposed in [3] eliminates the practices of pair programming and on-site customer to avoid the troubles caused due to those practices. In [4], the extreme practices of XP including on-site customer, continuous integration and continuous testing are removed from XP. In [5], nothing is removed from XP. Nothing is removed from XP in [6] also. Sequential quality activities i.e. testing, and refactoring are removed from XP in [7], and parallel quality improvement is introduced instead. In [8], nothing is removed from XP. The model proposed in [9] merges the “design” and “development” phases of existing XP model.

RQ4: What purposes are achieved by modifying XP?

The main purpose of the study conducted in [1] is to improve the extreme practices of XP, as they make the software development process more complex. In [2] a user is validated with the help of password so that he cannot deny a card raised by him. So, as the software is created with the agreement of the user and the developer, and the user authenticates it, the possibility of not accepting the solution after it has been created is reduced. It addresses the problem of authorization and authentication, and also helps the organization in faster and reliable delivery of the software. The purpose of the modified version of XP proposed in [3] is to overcome the problems of XP such as lack of documentation, poor architecture and design, and controversial practices like pair programming and on-site customer to make it a good method for medium scale

projects. The purpose of the model proposed in [4] is to modify the Extreme practices of XP that cause unnecessary effort and delay for small scale projects. The purpose to modify XP in [5] is to ensure the product quality. It aims to make sure that the product will provide quality characteristics as described by the stakeholders, and mentioned in ISO/IEC (Quality standards, ISO 9126, ISO 25000). The purpose to modify XP in [6] is to introduce reusability of components architecture in XP to reduce development time, effort, and cost to produce quality software. The main hindrance in developing reusable components in XP is the short term development time, so a new functionality is developed from scratch. It modifies XP to introduce reusability architecture in it. The purpose of modifying XP in [7] is to achieve better agility, documentation, and architectural design. Some of XP practices like sequential testing and refactoring effect its agility and result in less time for documentation. It makes XP suitable for small projects only. Such practices of XP are modified so that it may exhibit better documentation, design, and agility. In [8], the purpose of modifying XP is to make it suitable for medium scale projects also that have big development teams. It covers the limitations of XP like lack of design and less documentation which make it suitable for small projects only. It also introduces analysis and risk management in XP to overcome the project failure risk. The purpose to modify XP in [9] is to make it suitable for medium and large scale projects. It aims to provide risk responsiveness, improved documentation, stable requirements, and strong architecture. It also aims to provide a better model for security critical projects.

RQ5: How are the modified versions evaluated?

The modifications made in [1] are not evaluated. The modified model proposed in [2] is not validated. The model proposed in [3] is not verified. The modified version of XP proposed in [4] is not evaluated. The model presented in [5] is validated by performing simulations in iThink technology which make sure that the quality of product meets the depicted criteria. The framework presented in [6] is not evaluated. The model proposed in [7] is not verified. In [8], validation of the proposed model is done via two industrial case studies. One of them is for small scale projects and the other for medium scale. The extended model proposed in [9] is evaluated using three independent case studies for small, medium and large scale projects.

5. DISCUSSION AND CONCLUSION

A systematic literature review is conducted in this paper to investigate the current customizations of XP process model. We have defined a research methodology with inclusion and exclusion basis to gather the related data. Nine appropriate papers are chosen based on the criteria for the review. It has been concluded that even after much time of invention of XP, researchers are still modifying it to make it more appropriate in different scenarios. In this research, we have provided a comprehensive systematic literature review of researches from 2012 till 2018 on customizations of XP. We identified research questions, and then follow the systematic research process to extract the most related research articles from well-known digital search libraries. Answers to the specified questions were figured out by critically studying the selected articles. The results have depicted that the phases and practices of XP have been customized by the researchers to get better results in various projects. In the customizations, some researchers have added more practices into XP to make it more applicable in specific scenarios. Such practices of XP that make it inappropriate to use in large or critical natured projects have been removed. The main purpose of these customizations is to make XP appropriate for different size and type of projects with maintaining the agility of XP. Development of good quality software with reduced cost, effort, and time is a good thing being provided, but there is lack of empirical proof in the proposed models which makes it difficult to access their applicability. Empirical validation is recommended to prove the effectiveness of the customized proposed models.

REFERENCES

[1] Kunwar, S. (2018). Extreme Programming (XP) Simplified. *European Journal of Advances in Engineering and Technology, 5*(3), 198-206.

[2] Singh, A. (2018, March 29-31). *Integrating the Extreme Programming Model with Secure Process for Requirement Selection.* Paper presented at International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India.

[3] Anwer, F., & Aftab, S. (2017). SXP: Simplified Extreme Programming Process Model. *International Journal of Modern Education and Computer Science (IJMECS)*, *9(*6), 25-31.

[4] Anwer, F., Aftab, S., & Ali, I. (2017). Proposal of Tailored Extreme Programming Model for Small Projects. *International Journal of Computer Applications*, *171*(7), 24-27.

[5] Tabassum, A., Bhatti, S. N., Asghar, A. R., Manzoor, I., & Alam, I. (2017). Optimized Quality Model for Agile Development: Extreme Programming (XP) as a Case Scenario. *International Journal of Advanced Computer Science and Applications (IJACSA)*, *8*(4), 393-400.

[6] Nagalambika, S., Majunath, R., & Praveen, K. S. (2016). Component Based Software Architecture Refinement and Refactoring Method in Extreme Programming. *International Journal of Advanced Research in Computer and Communication Engineering*, *5*(12), 398-401.

[7] Qureshi. M. R. J., & Ikram, J. S. (2015). Proposal of Enhanced Extreme Programming Model. *International Journal of Information Engineering and Electronic Business*, *7*(1), 37- 42.

[8] Qureshi, M. (2014). Estimation of the new agile XP process model for medium-scale projects using industrial case studies. *International Journal of Machine Learning and Computing, 3*(5), 393-395.
[9] Qureshi M. R. J. (2012). Agile software development methodology for medium and large projects. *IET software*, *6*(4), 358-363.
[10] Rasheed, A., Zafar, B., Shehryar, T., Aslam, N. A., Sajid, M., Ali, N., ... & Khalid, S. (2021). Requirement Engineering Challenges in Agile Software Development. Mathematical Problems in Engineering, 2021.
[11] Anwer, F., Aftab, S., Shah, S. S. M., & Waheed, U. (2017). Comparative Analysis of Two Popular Agile Process Models: Extreme Programming and Scrum. *International Journal of Computer Science and Telecommunications*, *8*(2), 1-7.
[12] E.M. Schön., J. Thomaschewski and M.J. Escalona, "Agile requirements engineering: a systematic literature review," *Computer Standards & Interfaces*, vol. 49, pp.79-91, 2017.
[13] T. Karvonen, W. Behutiye, M. Oivo and P. Kuvaja, "Systematic literature review on the impacts of agile release engineering practices," *Information and Software Technology*, 2017.
[14] Anwer, F., & Aftab, S. (2017). Latest Customizations of XP: A Systematic Literature Review. *International Journal of Modern Education and Computer Science(IJMECS), 9*(12), 26-37.
[15] Henriques, V., & Tanner, M. (2017). A Systematic Literature Review of Agile and Maturity Model Research. *Interdisciplinary Journal of Information, Knowledge, and Management, 12*, 53-73.
[16] K. Dikert, M. Paasivaara and C. Lassenius, "Challenges and success factors for large-scale agile transformations: A systematic literature review," *Journal of Systems and Software*, vol. 119, pp.87-108, 2016.
[17] T. Dreesen, R. Linden, C. Meures, N. Schmidt and C. Rosenkranz, "Beyond the Border: A comparative literature review on communication practices for agile global outsourced software development projects," *System Sciences (HICSS), 49th Hawaii International Conference* pp. 4932-4941, IEEE, 2016.
[18] I. Inayat, S.S. Salim, S. Marczak, M. Daneva and S. Shamshirband, "A systematic literature review on agile requirements engineering practices and challenges," *Computers in human behavior*, vol. 51, pp.915-929, 2015.
[19] A.S. Campanelli and F.S. Parreiras, "Agile methods tailoring–A systematic literature review," *Journal of Systems and Software*, vol. 110, pp.85-100, 2015.
[20] M. Usman, E. Mendes, F. Weidt, and R. Britto, "Effort estimation in agile software development: a systematic literature review," in *Proceedings of the 10th International Conference on Predictive Models in Software Engineering*, pp. 82-91, ACM, 2014.
[21] I. Ghani and I. Yasin, "Software Security Engineering in Extreme Programming Methodology: A Systematic Literature Review," *Science International*, vol. 25, no. 2, 2013.
[22] T.S. Da Silva, A. Martin, F. Maurer and M. Silveira, "User-centered design and agile methods: a systematic review," in *Agile Conference (AGILE),* pp. 77-86, IEEE, 2011.
[23] J.E. Hannay, T. Dybå, E. Arisholm and D.I. Sjøberg, "The effectiveness of pair programming: A meta-analysis," *Information and Software Technology*, vol. 51, no. 7, pp.1110-1122, 2009.
[24] N. Salleh, "A systematic review of pair programming research-initial results," in *Proc. New Zealand Computer Science Research Student Conference (NZCSRSC08), Christchurch*, 2008.
[25] M Bano. and D. Zowghi, "User involvement in software development and system success: a systematic literature review," in *Proceedings of the 17th International Conference on Evaluation and Assessment in Software Engineering*, pp. 125-130, ACM, 2013.
[26] P. Brereton, B.A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," *Journal of systems and software*, vol. 80 no. 4, pp. 571-583, 2007.
[27] B. A. Kitchenham, S. L. Pfleeger, L. M. Pickard, P. W. Jones, D. C. Hoaglin, K. El Emam, and J. Rosenberg, "Preliminary guidelines for empirical research in software engineering," IEEE Transactions on software engineering, vol. 28, no. 8, pp.721 734, 2002.
[28] B. A. Kitchenham and S. Charters, " Procedures for Performing Systematic Literature Review in Software Engineering," EBSE Technical Report version 2.3, EBSE-2007-01, Software Eng. Group.
[29] B. Kitchenham, O.P. Brereton, D. Budgen, M. Turner, J. Bailey and S. Linkman, "Systematic literature reviews in software engineering–a systematic literature review," Information and software technology, vol. 51 no. 1, pp.7-15, 2009.
[30] Rasool, G., Aftab, S., Hussain, S., & Streitferdt, D. (2013). eXRUP: A Hybrid Software Development Model for Small to Medium Scale Projects.
[31] Anwer, F., Aftab, S., Waheed, U., & 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), 1-10.
[32] Ashraf, S., & Aftab, S. (2017). IScrum: An Improved Scrum Process Model. International Journal of Modern Education & Computer Science, 9(8).
[33] Nawaz, Z., Aftab, S., & Anwer, F. (2017). Simplified FDD Process Model. International Journal of Modern Education & Computer Science, 9(9).
[34] Ashraf, S., & Aftab, S. (2017). Scrum with the Spices of Agile Family: A Systematic Mapping. Modern education and computer science (MECS), 9(11).
[35] Ashraf, S., & Aftab, S. (2017). Latest Transformations in Scrum: A State of the Art Review. International Journal of Modern Education & Computer Science, 9(7).
[36] Ashraf, S., & Aftab, S. (2018). Pragmatic Evaluation of IScrum & Scrum. International Journal of Modern Education and Computer Science, 12(1), 24.
[37] Aftab, S., Nawaz, Z., Anwar, M., Anwer, F., Bashir, M. S., & Ahmad, M. (2018). Comparative Analysis of FDD and SFDD. International Journal of Computer Science and Network Security, 18(1), 63-70.
[38] Anwer, F., Aftab, S., Bashir, M. S., Nawaz, Z., Anwar, M., & Ahmad, M. (2018). Empirical comparison of XP & SXP. IJCSNS, 18(3), 161.

[39] Aftab, S., Nawaz, Z., Anwer, F., Salman, M., Ahmad, M., & Anwar, M. (2018). Empirical evaluation of modified agile models. Int. J. Adv. Comput. Sci. Appl, 9(6), 284-290.
[40] Aftab, S., Nawaz, Z., Anwer, F., Ahmad, M., Iqbal, A., Jan, A. A., & Bashir, M. S. (2019). Using FDD for small project: An empirical case study. International Journal of Advanced Computer Science and Applications, 10(3), 151-158.
[41] Ibrahim, M., Aftab, S., Bakhtawar, B., Ahmad, M., Iqbal, A., Aziz, N., & Ihnaini, B. N. S. (2020). Exploring the Agile Family: A Survey. IJCSNS, 20(10), 163.
[42] Ibrahim, M., Shabib Aftab, M. A., Iqbal, A., Khan, B. S., Iqbal, M., Ihnaini, B. N. S., & Elmitwally, N. S. Presenting and Evaluating Scaled Extreme Programming Process Model.
[43] Shabib Aftab, M. A., Hameed, N., Bashir, M. S., Ali, I., & Nawaz, Z. Rainfall Prediction using Data Mining Techniques: A Systematic Literature Review.
[44] Ahmad, M., Aftab, S., Bashir, M. S., & Hameed, N. Sentiment Analysis using SVM: A Systematic Literature Review.
[45] Matloob, F., Aftab, S., Ahmad, M., Khan, M. A., Fatima, A., Iqbal, M, & Elmitwally, N. S. Software Defect Prediction Using Supervised Machine Learning Techniques: A Systematic Literature Review.