

Agile Approaches For Effort Estimation In Software Development Process-A Review

Neha Gupta¹, Rajendra Prasad Mahapatra²

¹Research Scholar, SRM IST, Modinagar, Ghaziabad, India.

²Department of Computer Science and Engineering, SRM IST, Modinagar, Ghaziabad, India.

Abstract

According to the rapid changes in technologies that occurred in this world from the last few eras, it should be beneficial that update the software development approaches. Various organizations of software development are changed the working methodology so that they can compete in the market. According to the rapid increment in the demand for good software, companies are required to develop the software as early as possible. To make the profit it is clear that software should be complete in a short time and minimal usage of resources. To fulfill all these requirement companies, have a new methodology like Agile. This technology gives the new means of software development in case of easy development, short time, and less usage of resources. In this manuscript, some of the agile approaches are reviewed and give the advantages and challenges of these technologies with the help of comparison.

Keywords: Agile Modelling, SCRUM, Adaptive Culture, Object Oriented Approaches, XP

1. INTRODUCTION

Now according to today's era software industry grows every day according to the need of the users. As every field of today's life is connected with the software. No bare hardware is used without the software. We can say that software gives the actual meaning of the working of hardware [1]. The reason behind these changes is most important part of market and that called customer need. Entire software market is totally depending upon the customer need. As this need is grown and change rapidly so it makes difficult to update with software. If the developer uses the traditional approaches to resolve the needs of the customer, then it doesn't give the specified and fruitful results according to the needs [2]. Now a day's new approach is used called the agile approach. According to this approach the development of the software is easier and output of the software is really match up the needs of the customer and it is flexible in nature [3].

This manuscript is divided into some sections as section 1 give the introduction of the Software need and development process. In continuation previous work of software

development process is discussed. Section 2 describe the Agile Approaches. Section 3 gives the limitation of this approach and Section 4 describe the advantages of the same. At last in section 5 give the conclusion of this manuscript.

This type of approaches is used to keep track of various processes and step of software development and give the better output then the traditional approaches. When a developer start developing the software so it uses the better approach available in the market to divide the complete software development processes into various modules [4]. When a complete software development process is subdivided into various modules then it is easy to develop each and every module. This easiness is not only for the development process but it is also for the testing of that module, error finding, error correction.

In earlier days the software development approach gives not an efficient result in the output so software crises happen [5]. This particular problem arises in the 1970s. This is the ear when the customer need is drastically change according to the market so it need the changes in the software development process to fulfill the exact meaning of customer need. Developers use the latest approaches to overcome the problem and fragment the problem statement into various modules so that developers give complete and concentrated efforts to particular module in a single unit of time [6]. This approach gives some relaxation in the loss process. After using such upgraded approaches, the software problem is resolved and it provides the maximum output in terms of reducing the efforts and money to develop software.

Once in software development approach developer uses the structured methodology then it upgrades for the new approach that is object-oriented approach [7]. It is the combination of various unified process and rational unified process. According to this approach it will uses the structured approach best practice.

2. AGILE METHODOLOGY

When at early stages when discuss about the software development process then it seems that most of the user have the software needs with a normal requirements and if some upgradations required then these are not so much complex [8]. This means that to made such changes not so much technology and resources are required and it is not time consuming also [9]. Some of the software are used at the organization level and this software are complex in nature because of security point of view so it totally depends upon the organizations requirement. Some of the challenges are as below:

- Customer need: The most important input of any software development process is the customer need. At the starting customer needs are very basic and targeted to less number of clients but as the business grows the needs of the customer increased and then advancement in the software is required. Very few customers have the long term approach or say that they are not using the futuristic approach in all case of business. The actual situation is realise at the time of use the software for the real time application or environment. When the software developer develops the many software applications then he has the experience for such development and if any other software is required in the future then the development of that software is get easy just because of the experience [10].

- Team up with customer: All the analyst of the customer need taken the customer requirement and analysis these with lots of technique then give the blue print of these requirement to the developer. That means developer only develop the things received from the

customer requirement analyzer. If some mistake is happening at the end of analyzer then whole process get effected in terms of money and time. So customer involvement at each stage is necessary part of the software development process according to new techniques applied.

- **Cost Estimation and Time Requirement:** Before start with the software development process the cost estimation about the software development is carried out. This cost estimation is having the ideal cost and some cost in buffer for any unfair able condition. Time estimation is also the early process of the software development. AS the time is increased the cost estimation is also increased accordingly.

- **Wrong Requirement Analysis:** When customer requirement taken by the analyzer sometimes it may happen that the analyzer predicts the wrong requirement due to accent of the customer or similarities of the requirements [11]. If this happen then the further process of software development are carried out with this wrong input and that lead to losses in terms of money and time so it is very important to analyze the correct need and verify with the customer before start the development process.

With the presence of such issues, the OO programming improvement systems can't fulfill the destinations of programming advancement organizations. New improvement approaches must be applied so as to conquer these issues.

New methodologies of software development are totally different approach the previous traditional approaches in respect of complexity, need of resources at the initial level and platform heaviness. These traditional process are more likely to tends towards the documentation but in case of agile methodology it is different it concentrates on the software development process then documentation [12]. Agile approach provides more flexibility to the user and software up gradation according to the need of user time to time [13]. This platform provides the services to the user and provides the comfort ability to change according to the requirement in a short period of time [14]. After all changes according to the requirement user have the upgraded and required software.

2.1. Effort Estimation Techniques

This is the method by which developer can able to develop the software on time and fulfil all the needs of the customer within the deadlines [15]. This is happening just because of the agile effort estimation techniques. These techniques are as follows:

2.1.1. Poker estimation Technique

It is used for the small number of estimations in small team in circular position. In this techniques all the participant played with the numbered playing cards. For the estimation voting process between the members is happened at regular basis so that all players reached at the same estimation. The main theory of this technique is to estimate the relative small number in the batch of few players. This technique is applied between the small team only. This technique is used for estimation process again and again up to it reached to the consent. This technique is providing the better results of estimation because the estimation process is held between the various peoples so they all are involved in this process with full concentration and senses as shown in the figure 1. When various opinion of various expert person is come for the such estimation then it provides the better results.



Figure 1 : Steps of Planning Poker Process

According to agile estimation the average of individual estimation of the players is not provide results better than the average of expert’s estimation. In thus technique each and every player give estimation according to the expertise so at every level the expert estimation is provided [16]. At the last the final estimation is the average of all the estimation which is also better than the single player estimation. Figure 2 below provide the flow diagram of the planning poker process.

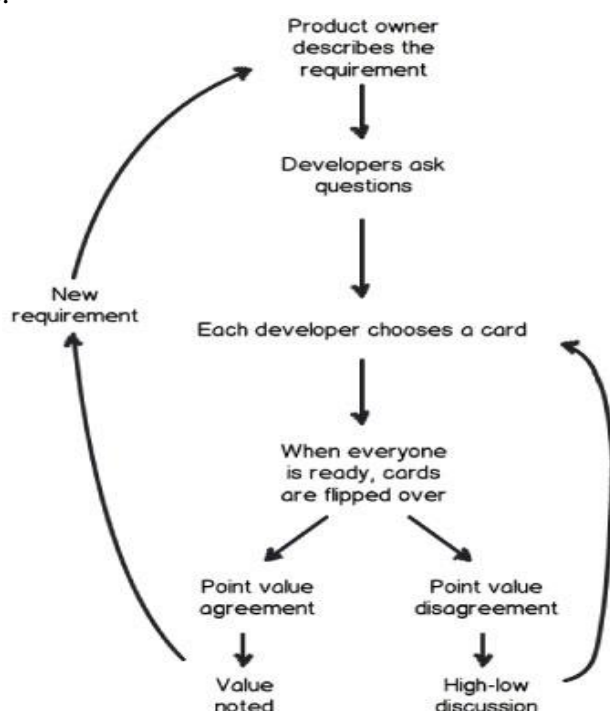


Figure 2: Step by Step process of Planning Poker

2.1.2. T-Shirt Sizing Technique

This provides the general and quick estimation idea of large number of items in a group format. This technique is also providing the better estimation in case of huge number of products. This

technique is applied when various teams are worked on the same product and provide estimation for the same single product then this particular product is sub classified in the form of T-Shirt size. T-Shirt size of classified in the form of S, M, L, XL, XXL and etc. This means that people working on the same quality product but with different sizes. With the help of this same product can be classified into various sub groups [17]. This division of sub group is totally depending upon the certain discussion happened in the group of persons. The technique used in this section is not the formal and professional so it only gives the rough estimation for the number of sizes of T-shirt. The estimation is given in the two reference like time and cost. According to below figure estimation of the product can be performed according to their sizes. The estimation with this technique is in the reference of time and cost as shown in the Figure 3.

Size	Calendar time	People	Cost ('000)
XS	1-3 weeks	1	5-15
S	1-2 months	1-5	20-200
M	3-6 months	5-10	300-1,200
L	7-18 months	10-20	1,400-7,200
XL	18+ months	20+	7,200+

Figure 3: provide the estimation according to cost and time

2.1.3. Dot Voting Estimation Technique

This is used the way of ranking the product in the high to low or low to high order. This technique is used with low number of product for estimation. This is able to give better estimation only with relative less number of products. This estimation technique is generally utilizing in case of decision making but sometimes it is used for estimation process also. In this method the product is in a limited amount with every person and each person give proper estimation for each and every product. This technique only work well in case of less number of products. it also provides estimation with large number of item set but in case of relative estimation it provides better results with less number of products. If the dots in the estimation is large in number than it shows that it is for the large number of products item set. When each person vote for each product then collect all the details and segregate the data [18]. After that check for the votes and number of dots at every estimation as shown in the figure 4. If the number of dots are more this means that the size of or number of the product is large. After all estimation and votes the priority is to be set from high to low and low to high.

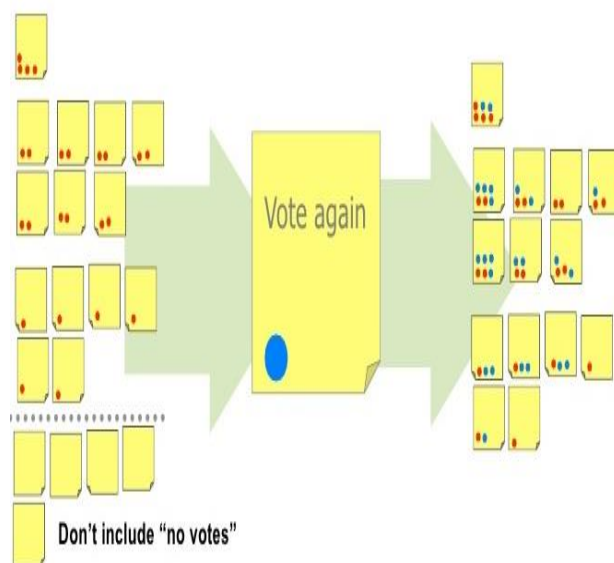


Figure 4: Dot Estimation Techniques

2.1.4. Bucket System Estimation Technique

In this technique bucket are used for the product segregation according to the estimation and it is used when large number of data available. This technique is used when the large number of products are available for the estimation. This technique provides the better results than the poker planning technique. Such type of methods is so useful for large number of products and it also include the persons large in number so that the estimation of each product can be easily and efficiently completed. Estimation can be performed in the groups and these group are created with the help of buckets. Each bucket identified the different kind of product and properties so that everyone can identified that particular bucket provide significance in which way so that the estimation of each bucket should be correct and provide better results [19]. This bucket also helpful in reduce the unnecessary estimation of product which are already processed for the estimation process. This particular technique has some advantages like. It is so fast because lots of product can be estimated in a very less amount of time.

With the help of this technique every item can equally take participate in the estimation process as shown in the Figure 5. So it is collaborative also. This technique not provide the exact results of any estimation but it provides the relative results of every estimation. The estimation provided with the help of this system is not identified by the single person so it is also called group estimation. This technique is group technique so provide better estimation than other techniques.

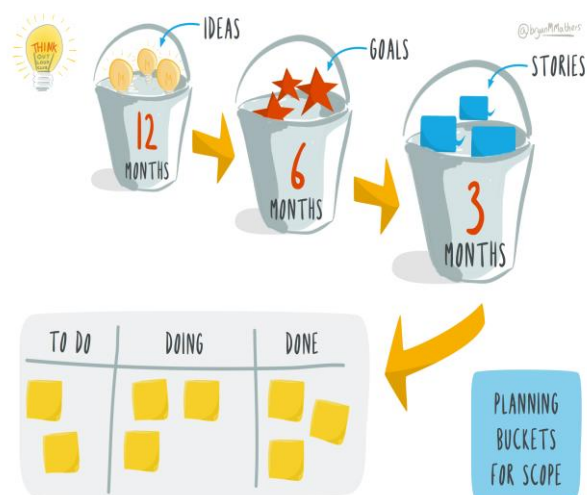


Figure 5: Bucket system effort estimation technique

2.1.5. Affinity Mapping Estimation Technique

This technique is generally used for finding the same properties between the products used for the estimation process. Person's responsible for the estimation grouped the product used. The process of estimation is used in this technique is visual and after that group them all in the ascending order of small group to large group as shown in figure 6. This technique is providing the better results with less number of persons and with respect of this less number of products. Estimation count provided by the persons is allocate to different group of products [20].

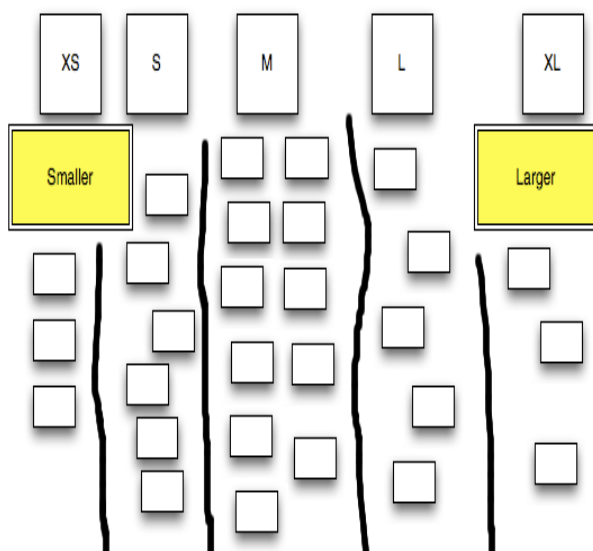


Figure 6: Affinity Estimation Process

2.2. Types of the Agile Methodology

- Adaptive Software Development
- Crystal methodologies family
- Agile Modelling
- SCRUM

- Feature driven Development
- Extreme Programming

2.2.1. SCRUM

This approach is invented in 1995 by the Ken Swaber. At the start it is used as normal approach but according to upgradation happens in this approaches it is consider in the agile methodology. This approach is work on the concept of simple design of documentation and updates required at the time of user requirement changes with time [21]. After consider in the agile methodology this approach also gives the abstract services to customer so that customer can be easily use these services and appreciate the design and software development with the help of this approaches.

As various methodologies are in the market under the agile approaches. For better results and fulfill the practical approach of the market it is clear that it would not be possible without sharing. This approach also gives the benefits of sharing data with other approaches [22]. SCRUM approach uses the project management approach as the original part of it own. Once the all rule and regulations are deciding for any approach this provide the proper way to develop the effective software.

SCRUM has some of the different services then the other approaches of the agile methodology like service of Backlog. According to this service the developer has the space in which he maintains the pending works of the development and these all works are keep in record according to their size and it is depending upon the complexity of the work. This backlog contains the how much work is complete regarding the service and how much is left? What are the resources are required to complete this pending work and what are the resources system already acquired? The pending work is simply described in the backlog.

According to the Figure 7 three main work of the SCRUM could be explained. First part out of these is played by the owner of the product by which the business is named in the market. Next is the middle part of the SCRUM architecture which is the combination of various actors like analyzer, developer and tester and etc. This middle layer is responsible for the complete requirement phase and development phase of the software [23]. Last work is of SCRUM Master which is responsible for the organized completion of the work whenever and according to the need of the customer changes.

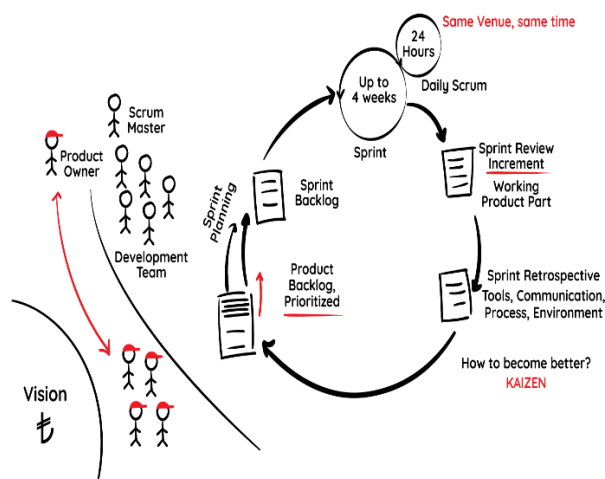


Figure 7. SCRUM Phases and Key Roles

According to the approach of SCRUM the entire work of the software development is divide into the phases and the entire team of the system is complete the work according to the sequence of the phase. Every phase contains the various step. Without completing every phase control is not shift to the new phase. Once the work of the particular phase is completed and testing of that work is done then it shifts to next phase [24]. When entire phase is over then each phase of the software is combined and final testing of the complete software is carried out once this testing give the positive results this means that software development work is over otherwise resolved the errors.

At last the SCRUM approach, give the various view according to the working approach. This approach is interactive and easy path of the development. This particular approach is also used for the program management.

2.2.2. Extreme Programming (XP)

This is the agile approach which is invented in year of 2000 by Kent Beck. This approach is very famous one because of its various working options, flexible environment for the software development and provide the flexibility to user also to made the updations easy according to the need of the market use. This is the combination of various services and ideas. But it just provides the different kind of services at the single platform and this give the better results in case of software development [25]. This is the approach generally used for the non-critical issues. But now a day it is already used in big software companies in the development of software because it made all the updations according to the market need.

Extreme Programming is very specific about the approach it uses for the development of the software. It follows the exact path it made to achieve the success in development and these steps are very simple and efficient as shown in figure 8. These ethics of this approach is not change at any stage of the software development like planning, designing, development and at last testing. The main aim of this approach is to satisfy the customer requirement first. This approach is very flexible for the moderation process at any phase. To complete its work on time and fulfill all the customer requirement it uses the team work fundamental.

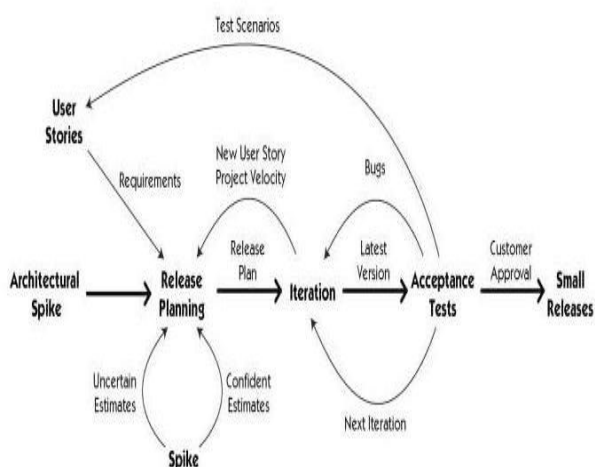


Figure 8. Phases of Programming Improvement

According to this approach of software development it is different from various other approaches by four ways. According to this approach the first one is most important factor communication between the customer and developer at the satisfactory level at every phase of software development. In the next point the architecture of this approach is simpler than the other approaches. In next point it provides the most flexible environment for the updations according to the customer need changes with time. In next point it gives the maximum assurance for fulfilment of the customer requirement according to the market day to day changes. These points of the extreme programming are also called the values of this approach.

According to the various case studies and out of these one case study by Kuppuswami et al. (2003) described that if developer improve the strategies at the XP internal level than overall complexity of the software development is get reduced. Extreme Programming is the versatile approach and it also uses some hit and trial methods to improve the development process with team. The main task of extreme programming is to find out the exact requirement of the customer, customize it into simpler phases and then applied the approach for the fulfillment of the customer requirement.

2.2.3. Agile Modelling

When talking about the software development process than modelling is the very important approach in this. This approach mainly focuses on the programming but this also enable the software programmer to keep track of all the complex issues in consideration. This particular approach is designed in 2002 by Scott Ambler. This approach is efficient with the combination of various parameters like values, ethics and resources for software development. This approach can be designed to resolve the problem in software development phases. It is used along with the other agile approaches by aiming that the customer needs always fulfil according to the changes of the required technology in future.

This AM approach is benefitted at some areas in software development in against of the various factors like feedback, humility, communication and simplicity. According to this approach sometime it is possible that attention can be provide to those peoples which are know some important part of the software development and as this part is the essential part so benefit is share with these peoples for that.

This approach is very much familiar with the extreme programming in reference of various factors like feedback and etc. AM approach uses the various also include the requirement of the modelling in the software development process and its utilization in the system. The aim of this approach is to give multiple effective models and continuously communicate with the customer and focus on the completion of the quality work.

Agilehas various version for model driven software development like Model Driven Development (AMDD) is the example. When working with the AMDD and plan for its implantation then complter system would be created at the initial level of the software development process. Only performing of the medling is left at the time of launching. Agile modelling is basically applied with different techniques like XP and Test Driven Development (TDD).

3. LIMITATION OF THIS APPROACH

The fulfillment of the customer need is main aim of any agile software development approach with the help of easy development of strategies. Developers which are uses the agile approach for the software development said that codes are the important factor for development and the others said that this code is supported the memory loss. Only the documentation part of this approach is not enough to provide the best services to the customer.

Implementation of the Agile Methodology is possible only with the few limitations. These limitations import some problems in the complete implementation process of this approach. As the documentation of this technique is not so much but still it is not support the green engineering and its maintenance. In this agile methodology one main point is that at every level of work and phases the participation of the customer is important and success of the software is depending upon the proper communication of the customer and the requirement analyzer representative. Agile technique is dependent of the developer knowledge and managements skills how to manage the process of the development of software. For a single module all the work is completed by single developer. Reusability is not supported by the agile methodology that reduces the utilization of the software number of times. As the agile approach is designed for the particular problem that particular solution is not necessarily solve every problem of the software development. This technique is providing the better solution with the team of small in size if the size of the team will increase than the results get worst with time.

To apply the agile methodology in the successful manner lots of assumption is carried out. Example for the success of the agile model at any system the customer relationship should be very good in every sense so that analyzer completely satisfied with its requirement and developer develop the system accordingly. But practically it is just an impossible task to complete all the session of customer requirement in a smooth manner. Some of other examples are also available in the market. Without applying these assumptions with this agile methodology this is not provide the better solution of the problem of software development than the other methodologies.

4. ADVANTAGES OF AGILE METHODOLOGY

- Agile methodology is the best approach for the software development process because it applies all the futuristic and customer oriented factors in this approach.
- This methodology is ready to accept the challenges come in front of developer according to the change in the need of the customer.
- This approach provides the updated solution of the customer requirement in a rapid way so that customer can able to fulfil the requirement and it increase the reply of customer towards the organization and developer.
- If sometime a situation arises for major changes or updation then with the help of this approach this work is completed in a simpler way with all the documentation in the computerized format so that it is available at every stage of software development.

This approach provides so many ways to improve the organizational work in various respect. It provides this improvement which gives:

- Results of unrivalled quality;
- More straightforwardness;
- adaptability;

- Diminished danger of missed objectives;
- Greater profitability;
- More noteworthy contribution and fulfilment of partners.

Table1: Comparison between the Agile methodologies

Approach	Main Feature	Key Points	Lacking point
XP	Customer driven Development, small team, daily builds	Refactoring-the ongoing redesign of the system to improve its performance and responsiveness to change	While individual practice is suitable for many situations, overall view & management practices are given less attention.
Agile Modeling	Adaptive Culture, Collaboration, Mission-driven component based iterative development	Organizations are seen as adaptive system. Creating an emergent order out of a web of interconnected individuals.	Agile Modeling is more about concepts and culture than the software practice.
SCRUM	Independent, small, self-organizing development teams, 30-days release cycles.	Enforce a paradigm shift from the “defined and repeatable” to the “new product development view of SCRUM”	While SCRUM details in specific how to manage the 30-days release cycle, the integration and acceptance tests are not detailed.

5. CONCLUSION

When the digitalization of data is start then the requirement of the software is increased because without this everything is with manual work with hardware. As software development start for ease of developers, scientist and other persons. This is the time period of 10970s when this process is come into picture. But the problem with this software development approaches is that these are so complex, time taking and lots of man power is required. Agile approach gives the new way to the software development organization to develop the software with very easy way. According to this approach the total success is depend upon the first communication between the developer and the customer to exactly identified the customer need and with the help of this the efforts are reducing to develop such software. This communication is not stop it is happening at regular intervals so that whenever developer confuse at some point of development with this communication that confusion get resolved and the progress of software

development cannot be stop. This approach is ready to deliver what needed by customer and at the time of demand.

According to the study in the manuscript three agile approaches are used in which SCRUM approach works on the concept of managing the activities of the software development so that development process gets easier. In case of XP approach, it works with the concept of development of the new code and test that particular for it is to be created. If it works properly means this approach works well while in case of Agile Modelling approach is focused on the documentation work, data and the methods for modelling approach.

Agile approaches are not used every time for the development of the software and use of the approaches in software development is depend upon the requirement and the condition in which it is going to apply. Agile approaches are not the best option for the developers if requirement of the customer is not well defined and accurately identified by the developers. This approach is not the worthy option for the setups in which new developers are involved reason behind that this approach is only give noble results when the customer needed are well identified and with the new developers they are not so much focus on proper customer need. Agile approaches are the ideal approaches for software development when customer need is properly identified and time line of the software development is countable. Agile systems are not most appropriate for all tasks.

REFERENCES

- [1] T. Schweighofer, A. Kline, L. Pavlic and H. Marjan, "How is effort estimated in agile software development projects?", Proc. 5th Workshop Softw. Qual. Anal. Monitor. Improvement Appl. (SQAMIA), pp. 73-80, Aug. 2016.
- [2] S. Bilgaiyan, S. Sagnika, S. Mishra and M. Das, "A systematic review on software cost estimation in agile software development", J. Eng. Sci. Technol. Rev., vol. 10, no. 4, pp. 51-64, 2017.
- [3] E. Dantas, M. Perkusich, E. Dilorenzo, D. F. S. Santos, H. Almeida and A. Perkusich, "Effort estimation in agile software development: An updated review", Int. J. Softw. Eng. Knowl. Eng., vol. 28, no. 11, pp. 1811-1831, Nov. 2018.
- [4] S. W. Munialo and G. M. Muketha, "A review of agile software effort estimation methods", Int. J. Comput. Appl. Technol. Res., vol. 5, no. 9, pp. 612-618, Sep. 2016.
- [5] M. Vyas, A. Bohra, C. S. Lamba and A. Vyas, "A review on software cost and effort estimation techniques for agile development process", Int. J. Recent Res. Aspects, vol. 5, no. 1, pp. 1-5, Mar. 2018.
- [6] R. Hoda, N. Salleh, J. Grundy and H. M. Tee, "Systematic literature reviews in agile software development: A tertiary study", Inf. Softw. Technol., vol. 85, pp. 60-70, May 2017.
- [7] E. D. Canedo, D. P. Aranha, M. D. O. Cardoso, R. P. D. Costa and L. L. Leite, "Methods for estimating agile software projects: A systematic review", Proc. 30th Int. Conf. Softw. Eng. Knowl. Eng. (SEKE), pp. 34-39, Jul. 2018.
- [8] A. A. Mohammed, A. Ahmad and M. Omar, "Improvement of agile software development size & effort estimation methods", Int. J. Innov. Technol. Exploring Eng., vol. 8, no. 8, pp. 357-362, Jun. 2019.

- [9] V. Nepomuceno and S. Soares, "On the need to update systematic literature reviews", *Inf. Softw. Technol.*, vol. 109, pp. 40-42, May 2019.
- [10] K. R. Felizardo, E. Mendes, M. Kalinowski, E. F. Souza and N. L. Vijaykumar, "Using forward snowballing to update systematic reviews in software engineering", *Proc. 10th ACM-IEEE Int. Symp. Empirical Softw. Eng. Meas. (ESEM)*, pp. 53:1-53:6, 2016.
- [11] Schön, E.-M, "A Framework for Modeling and Improving Agile Requirements Engineering", Ph.D. Thesis, University of Seville, Sevilla, 2017.
- [12] Sedeño, J. et al. "An Approach to Transform Public Administration into SOA-Based Organizations", *10th International Conference on Web Information Systems and Technologies*, Barcelona, 3-5 April 2014, 135-142.
- [13] Torrecilla-Salinas et al. "Using an Agile Framework to Deliver e-Government Services in Public Administrations", *21st Annual Software Quality Management*, London, 2-5 September 2013, 167-175.
- [14] Torrecilla-Salinas et al. "Estimating, Planning and Managing Agile Web Development Projects under a Value-Based Perspective", *Information and Software Technology*, 61, 124-144. <https://doi.org/10.1016/j.infsof.2015.01.006>, 2015
- [15] Schön, E.-M et al. "Enterprise Experience into the Integration of Human-Centered Design and Kanban", *Proceedings of the 11th International Joint Conference on Software Technologies*, Lisbon, 24-26 July 2016, 133-140. <https://doi.org/10.5220/0005942601330140>
- [16] Begel A., N. Nagappan, "Usage and Perceptions of Agile Software Development in an Industrial Context: An Exploratory Study", *First International symposium on empirical software engineering and measurement*, pp. 255-264, 2007.
- [17] Maher P. "Weaving Agile Software Development Techniques into a Traditional Computer Science Curriculum", *Proc. of 6th IEEE International Conference on Information Technology: New Generation*, pp. 1687-1688, 2009.
- [18] Zuo A., A. Jing, X. Chen, "Research of Agile Software Development Based on Formal Methods", *International Conference on Multimedia Information Networking and Security*, pp. 762-766, 2010.
- [19] J Rees M, "A Feasible User Story Tool for Agile Software Development", *Proc. Of 9th Asia-Pacific Software Engineering Conference (APSEC' 02)*, 2002.
- [20] OutiSalo, P. Abrahamsson, "Integrating Agile Software Development and Software Process Improvement: a Longitudinal Case Study", pp. 193-202, 2005.
- [21] Wang X. "The Combination of Agile and Lean in Software Development: An Experience Report Analysis", *IEEE Agile Conference*, pp. 1-9, 2011.
- [22] Mordinyi R., E. Kuhn, A. Schatten. "Towards an Architectural Framework for Agile Software Development", *17th IEEE International Conference and workshops on Engineering of Computer Based Systems*, pp. 276-280, 2010.
- [23] A. Livermore J, "Factors that impact implementing an Agile Software Development Methodology", pp. 82-85, IEEE, 2007.
- [24] Ahmed A., S. Ahmad, Dr. N. Ehsan, E. Mirza, S.Z. Sarwar. "Agile Software Development: Impact on Productivity and Quality" , pp. 287-290, IEEE, 2010.
- [25] Wang Y., D. Sang, W. Xie, "Analysis on Agile Software Development Methods from the View of Informationalization Supply Chain Management", *3rd International*

- Symposium on Intelligent Information Technology Application Workshops”, pp. 219-222, 2009.
- [25] J. Lehman T., A. Sharma. “Software Development as a Service: Agile Experience”, Annual SRII Global Conference, pp. 749-758, IEEE, 2011.
- [26] A. Silva, T. Araujo, J. Nunes, M. Perkusich, E. Dilorenzo, H. Almeida and A. Perkusich , A systematic review on the use of definition of done on agile software development projects, in Proc. 21st Int. Conf. Evaluation and Assessment in Software Engineering, 2017, pp. 364–373.
- [27] M. Usman, E. Mendes, F. Weidt and R. Britto , Effort estimation in agile software development: A systematic literature review, in ACM Int. Conf. Proceeding, 2014, pp. 82–91.
- [28] V. Lenarduzzi, I. Lunesu, M. Matta and D. Taibi , Functional size measures and effort estimation in agile development: A replicated study, in LNBIP, Vol. 212, 2015, pp. 105–116.
- [29] S. Grapenthin, M. Book, T. Richter and V. Gruhn , Supporting feature estimation with risk and effort annotations, in Proc. 42nd Euromicro Conf. Software Engineering and Advanced Applications, 2016, pp. 17–24.
- [30] S. Bilgaiyan, S. Mishra and M. Das , A review of software cost estimation in agile software development using soft computing techniques, in 2nd Int. Conf. Computational Intelligence and Networks, 2016, pp. 112–117.
- [31] S. W. Munialo and G. M. Muketha , A review of agile software effort estimation methods, Int. J. Comput. Appl. Technol. Res. 5 (9) (2016), pp. 612–618.
- [32] S. K. Sehra, Y. S. Brar, N. Kaur and S. S. Sehra , Research patterns and trends in software effort estimation, Inf. Softw. Technol. 2017, pp. 1–21.
- [33] K. R. E. Felizardo, M. Kalinowski, É. F. Souza and N. L. Vijaykumar , Using forward snowballing to update systematic reviews in software engineering, in Proc. 10th ACM/IEEE Int. Symp. Empirical Software Engineering and Measurement, 2016, pp. 1–6.
- [34] N. B. Ali, K. Petersen and C. Wohlin , A systematic literature review on the industrial use of software process simulation, J. Syst. Softw. 97 (2014) 65–85.
- [35] Y. Zhou, H. Zhang, X. Huang, S. Yang, M. A. Babar and H. Tang , Quality assessment of systematic reviews in software engineering: A tertiary study, in Int. Conf. Evaluation and Assessment in Software Engineering, 2015, pp. 1–14.
- [36] 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, J. Syst. Softw. 80 (4) (2007) 571–583.
- [37] M. Staples and M. Niazi , Experiences using systematic review guidelines, J. Syst. Softw. 80 (9) (2007) 1425–1437.
- [38] M. Petticrew and H. M. Roberts , Systematic reviews in the social sciences: A practical guide, in Cebma. Org, 2006, pp. 336.
- [39] S. J. Taylor and R. Bogdan , Introduction to Qualitative Research Methods: The Search for Meanings (John Wiley and Sons, 1984).

- [40] T. Schweighofer, A. Kline, L. Pavlic and M. Hericko , How is effort estimated in agile software development projects? in 5th Workshop on Software Quality, Analysis, Monitoring, Improvement, and Applications, Budapest, Hungary, 2016, pp. 73–80.
- [41] M. Usman, E. Mendes and J. Börstler , Effort estimation in agile software development: A survey on the state of the practice, in Proc. 19th Int. Conf. Evaluation and Assessment in Software Engineering, 2015, pp. 13:112:10.
- [42] A. Silva and F. Ramos , Ordering the product backlog in agile software development projects: A systematic literature review ordering the product backlog in agile software development projects, in Int. Conf. Software Engineering & Knowledge Engineering, 2017, pp. 74–80.
- [43] A. Philip, S. Ali, I. Roliana and N. M. Mohd , A systematic literature review of software requirements prioritization research, *Inf. Softw. Technol.* 56 (6) (2014) 568–585.
- [44] K. Eetu, V. M. Mika and I. Juha , Using metrics in agile and lean software development: A systematic literature review of industrial studies, *Inf. Softw. Technol.* 62 (2015) 143–163.
- [45] C. Wohlin , Guidelines for snowballing in systematic literature studies and a replication in software engineering, in Proc. 18th Int. Conf. Evaluation and Assessment in Software Engineering, 2014, pp. 1–10.
- [46] S. Jalali and C. Wohlin , Systematic literature studies: Database searches vs. backward snowballing, in Proc. ACM-IEEE Int. Symp. Empirical Software Engineering and Measurement, 2012, pp. 29–38.
- [47] T. Halkjelsvik and M. Jørgensen , Time Predictions: Understanding and Avoiding Unrealism in Project Planning and Everyday Life (Springer International Publishing, 2018), pp. 9–10.
- [48] S. Malathi and S. Sridhar , Analysis of size metrics and effort performance criterion in software cost estimation, *Indian J. Comput. Sci. Eng.* 3 (1) (2012) 24–31.
- [49] L. Lavazza and S. Morasca , On the evaluation of effort estimation models, in Proc. 21st Int. Conf. Evaluation and Assessment in Software Engineering, 2017, pp. 41–50.
- [50] C. V. Melo and E. C. Katayama , The evolution of agile software development in Brazil, *J. Brazilian Computer Society* 19 (2013) 523–552.
- [51] E. Coelho and A. Basu , Effort estimation in agile software development using story points, *Inf. Softw. Technol.* 3 (7) (2012) 7–10.
- [52] M. Cohn , *Agile Estimating and Planning* (Pearson Education, 2015).
- [53] N. VersionOne and K. Okada , Method to estimate parameter values in software prediction models, <https://versionone.com/pdf/VersionOne-11th-Annual-State-of-Agile-Report.pdf>, in 11th Annual State of Agile Development Survey Results. Accessed in: 02-12-2017, 2017.
- [54] R. Britto, M. Usman and E. Mendes , Effort estimation in agile global software development context, in *Agile Methods. Large-Scale Development, Refactoring, Testing, and Estimation*, Vol. 199 (Springer, 2014), pp. 182–192.
- [55] M. Usman and J. Borstler , An effort estimation taxonomy for agile software development, *Int. J. Softw. Eng. [Knowl. Eng.* 27 (4) (2017) 641–674.