IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

A REVIEW AND SURVEY OF SOFTWARE DEVELOPEMENT METHODOLOGIES

Ms. Purvi Sankhe
SAGE University, Bhopal,
Madhya Pradesh.

Dr. Mukesh Dixit
SAGE University, Bhopal,
Madhya Pradesh.

Abstract: **Software development process, which is really defined as software development lifecycle (SDLC) basically consists of various phases like planning, requirements analysis, design, implementation, testing, deployment and maintenance. SDLC is a set of predefined tasks performed at every step of software development. Software Development lifecycle can consist of various process methodologies that can be undertaken by developers for the development of software. These methodologies are derived from two basic methodologies: Traditional methodologies and agile methodologies. Traditional methodologies are more planning driven. Agile methodologies are based on more incremental and iterative development where requirements and solutions evolve via flexibility and collaboration of self-organising teams involved in the development process. In this paper, we will describe the various methodologies that are used for software development process, their merits and their demerits, a detailed review of literature focusing on key research in this area that has been done till now, how the selection of these methods can affect the overall development process and the product and also the results of an empirical survey conducted regarding the different methodologies adopted by some well-known organisations in the field.**

The results of survey primarily focus on two important aspects: 1. there is no one method that an organisation can adopt for all the projects. An organisation needs to take several parameters into consideration while selecting a method of software development, the key parameters of which are: Project focus, expected product of the project, flexibility of timeline. Allotted budget and the number and type of teams working on the project, along with many other parameters. It can be concluded that depending upon the client requirements and feasibility, an organisation can also develop

an altogether a new model of working which is termed as hybrid model i.e., using combination of different methods at the same time, in order to gain more efficiency and deliver high quality end product.

Keywords: ***Software development lifecycle; traditional methods; waterfall method; agile method; scrum.***

I. INTRODUCTION

To begin with the basics, software literally means a set of instructions which can be read by a machine, and which directs the computer's processor to perform the operations as per the instructions fed. Software development process, which is really defined as software development lifecycle (SDLC) basically, consists of various phases like planning, requirements analysis, design, implementation, testing, deployment and maintenance. SDLC is a set of predefined tasks performed at every step of software development. International organisation of standardization/ International Electrotechnical Commission 12207 is an international standard which is developed for the software lifecycle process, and these are the standards that the software companies adhere to, in order to deliver a high-quality software.[1] A brief overview of SDLC is depicted in Figure 1.

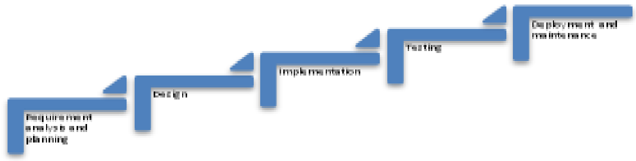

Fig 1: Software development life cycle

Software Development lifecycle can consist of various process methodologies that can be undertaken by developers for the development of software. These methodologies are derived from two basic methodologies: Traditional methodologies and Agile methodologies. Traditional methodologies, as the name suggests, are more planning driven where the process begins with selection and documentation of entire set of requirements and specifications at the first step which is then followed by high level designing, development and inspection. But with the changing times and changing concepts, developers have come up with more agile and flexible methodologies and these processes can be grouped under the Agile methodologies. Agile methodologies are based on more incremental and iterative development where requirements and solutions evolve via flexibility and collaboration of self-organising teams involved in the development process. Agile methods are more test-driven rather than planning-driven. [1]. The different types of processes that the traditional and agile methodologies use, can be understood from the figure below:

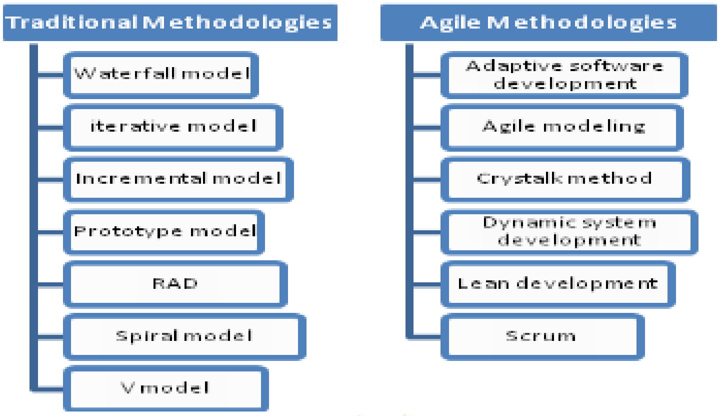

Fig 1 : Different methodologies for SDLC

With the changing times, changing market requirements, changing ideologies it becomes imperative for companies to adopt a method which not ensures delivery of a high-quality software but also ensure consumer satisfaction. The positives of the traditional approach can be combined with the positives of the agile methods and there is also a new method that can be used for software development i.e. Hybrid approach. The entire project can be planned as per the traditional methods and the subparts of the project can be managed using agile principles. [2]

This paper will describe various methodologies that are used for software development process, their merits and their demerits, a detailed review of literature focusing on key research in this area that has been done till now, how the selection of these methods can affect the overall development process and the product and also the results of an empirical survey conducted by us regarding the different methodologies adopted by some well-known organisations in the field.

II SOFTWARE DEVELOPMENT METHODOLOGIES:

As described earlier, there are basically 2 types of software development methodologies used till date: First is the traditional method which include but are not limited to waterfall method, iterative model, spiral model, incremental model, prototype model and many more. The second type of methodology is the evolution of these traditional methods to more recent agile methodologies which include Scrum, lean development, Dynamic system development, adaptive software development, crystal method, agile modelling and many more. In this paper, we would be focusing in some of these important methodologies of software development.

A *Traditional Method:*

1. ***Waterfall method:*** This is a traditional method in which the software is developed using planning and plotting on every step if development with every step starting only after the previous step is completed. It is basically a linear process flow where every team member is driven to achieve the end result of that particular stage and by advancing to the next stages of development, achieve the final end product. Every individual has a clear and defined part at every stage and these roles are usually not changed. This method is especially suitable for large projects where there are defined timelines and changing of these timelines would prove to be very expensive and discouraging. This model is also suitable to be implemented in organisations where strict and defined rules are regulations are to be followed and minimal changes are accepted. This method usually defines the cost and timelines beforehand and there are no changes to the timelines or the cost, so it is specially a good fit for projects when it is controlled by time and money. Every stage involves pre-set requirements which are to be fulfilled and the next stage would never begin without completion of the previous stage. The advantage of this method is that it can deliver excellent results giving a defined set of features in accordance with the defined financial plan and time. [3]
2. ***Spiral method:*** Several changes were implemented in the traditional waterfall method which led to the development of spiral model. Spiral model, one of the most important models for software development, was initially proposed by Boehm. It is a result of proper incorporation of the waterfall model and prototyping models and so it is also known as meta-model. This means that the software development is done systematically in accordance with the waterfall method and at the same time every iteration can be developed, tested and shown to the user, as per the prototype models. This method is most suitable when the projects are not large, not expensive or not complicated to handle. The unique feature of this model is its capacity to handle risks, along with development in a systematic way. So, this method is considered iterative, and risk driven. There are basically 4 phases of spiral model which are: Objective setting, Risk assessment and mitigation, Development and Validation, Planning. As the names suggest, the first phase involved setting of specific objectives, the second phase involves identification of risks and

ways of handling and reducing them, the third phase involves development and validation of the product or part of product i.e., the current iteration of the prototype is developed and tested, and the fourth phase defines the review of previous spiral and planning of next round of spiral. The main advantage of this model is that it defines and handles risk at every stage/spiral, added functionalities on every stage and ability to have early customer feedback. The figure shown below depicts the basic functionality of spiral model.[1], [4], [5]

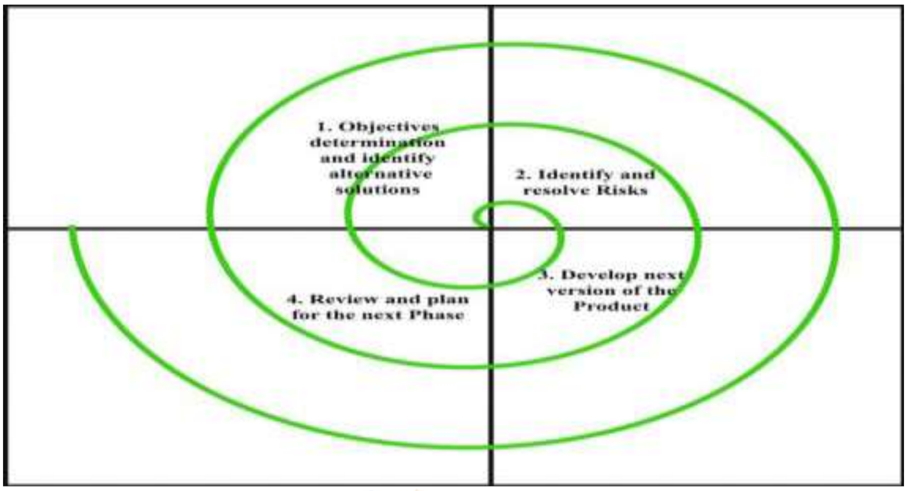
1. Objectives determination and identify alternative solutions
2. Identify and resolve Risks
3. Develop next version of the Product
4. Review and plan for the next Phase

Fig 2 : Spiral model[5]

B *Agile Software Development Methodology:*

As the name suggests, these methods are driven by various qualities like readiness for motion, dexterity in motion, being agile. As described by Williams and Cockburn in 2003, agile methodology is all about “feedback and change”. This is a nonlinear process of software development where short feedback loops are incorporated to achieve a desired and predictable outcome. In the year 2005, Ericksson et al., described the lightness of agile development methodology; defining agility as to “strip away as much of the heaviness, commonly associated with the traditional software-development methodologies, as possible to promote quick response to changing environments, changes in user requirements, accelerated project deadlines and the like”. [6]Thus, agile methods of software development are defined differently from the traditional methods focusing on adaptation to change and hence delivery of high-quality products. This method is based on repeated strategies for working on constant issues and customer feedback. Adaptive Software Development, Agile Modelling, Crystal Method, Dynamic System Development, Lean Development and Scrum are some of the methods categorised under the agile software development methodology. This method is best suited for projects having limited primary needs and need not adhere to the strict regulations and offers the luxury of flexibility. It is also suitable for projects that need to be delivered in a short period of time because the processes and results are given more importance than the large requirements and documentation. It is suitable for projects offering flexibility of time and money. [6]

1. ***Scrum model:*** This model was originally proposed by Ken Schwaber and Jeff Sutherland in the year 1995. The word “Scrum” is derived from the game of rugby where the scrum restarts the game after the ball has gone out. Applying the same concept to software development, implies that the scrum gets the team back together and everyone is assigned particular role and direction. It is a process framework that consists of following steps:

- Sprint planning: The entire collaborates and defines the work to be completed in a sprint and sets up sprint goals.
- Daily Scrum: this process helps the team members to track the progress of the work nd the amount to of work required to be completed in order to achieve the sprint goals.
- Sprint review: It is planned at the end of sprint to analyse the increment and adapt to the backlog if requirement
  - Sprint retrospective: It is a retrospection of the previous sprint and improvisations to be made for the planning of next sprint to be smoother than the previous sprint. [1]

As goes with the advantages of an agile development methodology, scrum method is iterative in nature. It is lightly controlled and offers the flexibility of adapting the change as required by the project. Due to shorter sprints in the project, it is easier to get customer feedback and enable improvisations where required. It can help reduce the overhead costs and hence produce cheaper results. It improves collaboration between the team members and within teams.

2. ***Extreme programming (XP):*** This agile method of software development is usually suitable for small teams dealing with vague or rapidly changing requirements. This is very different from the traditional methods in different ways like: there is almost negligible documentation, in fact the only documentation that is available is the original source code; there is no software specification; there are no separate design or testing phases, designing, testing and execution are all done together in incremental phases; there are no formal reviews or inspections. XP basically combines special practices like pair programming i.e., two developers work together on a single code, working at same terminal, constant improvisation i.e. constant integration of changes, automated regression testing, development in small increments, collective ownership, on-site customer and occasional refactoring or restructuring. The main feature of this method that the change is quickly embraced in a constantly evolving system. This system can also be customised or modified as per the requirements by either customising the phases or the practices. The downside of this method is that it requires constant meetings which cost time and money. It also requires many modifications which can be tiresome for some developers. [7]

*3. **Feature driven development:*** It is a software development methodology which is based on iterative and incremental process. The software is developed according to the client valued functionality. Every feature is assigned a number and that number represents the priority of the feature for the client. FDD consists of five basic process which include: developing an overall model, building an overall model, planning by feature, designing by feature and building by feature. These processed are based on a well known model called ETVX , an important model for writing template of process. The process begins with the development of overall model with an informal features list. Alternative model may also be built and the best model is selected for further process. The team then begins communicating with the client about the feature list and the features are prioritised as per the client values. If a feature is complex and large, it can be divided into small parts and processed accordingly. The feature with the highest priority should be considered for the first iteration. The development team then builds a design in form sequence diagram, classes with prototypes and thus refining the overall model. The design is further inspected and if suitable, the developer writes a code for the design and the code is further tested and inspected. After the successful implementation of codes for every feature, every feature is then developed into a system. The drawback of this model is that modelling takes more time than the actual iteration. This model does not provide efficient process for implementation of changes during the development process. If the requirements are changes, the entire plan needs to be changed and the model needs to be redeveloped right from the scratch. This drawback reduces the overall agility in the process. [8], [9]

III Review Of Literature:

In this section of our paper, we intend to summarise the various studies that have been carried out in the past for addressing the challenges as well as the success factors of both the basic methodologies of software development process i.e. the traditional methodology and the scaling of these methods to agile transformations over the time.

The table below includes an extensive review of literature for all the studies detailing the success factors and challenges during agile transformation processes.

Table 1: Studies detailing the success factors and challenges during agile transformation processes

Study details	Key observations	Study limitations
Thesing et al., -empirical data from 15 expert interviews across different industries in Germany,2020 .[2]	Differentiating factors between traditional and agile methods were compared and a decision model was proposed to select the suitable approach Traditional project management is characterised by holistic planning in advance, stability and a long-term perspective, while the planning in agile projects is more flexible, step by step and is short term oriented. Another relevant difference lies in the working structure of the teams. The traditional project management approach is entirely dependent on the execution of initial project plan whereas in the agile project management, the team requires to be more transparent in communication with the customer, with short and frequent feedback cycles from the customers and a high level of flexibility.	The results of the expert interviews are not representative and cannot be applied to all the industries as the results are derived from a limited database.Use of a brader dataset with quantitative and qualitative results may produce better results and may be helpful in refining and validating the results of the present study.
Henry et al., - a systematic literature review, 2022[10]	Empirical studies on large-scale agile software development are increasing rapidly and steadily in recent years. There were 31 challenges identified which were grouped in 09 categories, in using large-scale agile methods. Fluidity of agile roles and no direct mapping from old job roles to the new ones, and Scaling agile practices to non-development units are some of the persistently reported challenges.49 organisations predominantly used Scrum method, either alone or in combination with other methods for large-scale development. This was followed by SAFe, being used by almost 19 organisations. There were 13 organisations which did not used any specific large-scale method, but used a combination of different methods based on the requirement.	While agile approaches to large scale development are becoming increasingly prevalent, they are by no means the only approaches for large scale project delivery.This study did not consider the levels of method adoption across the papers studied.
Dikert, Paasivaaa &Lassenius- an extensive review of primary studies regarding the challenges and success factors of agile transformation. (52 primary	The most important success factors: management support, choosing the right agile model, mindset, and alignment with the organization's value. Challenges: resistance to change, lack of training, and misunderstanding agile.	Limitations: The studies are dates earlier than 2010 and almost all studies involve transformation from waterfall model to agile processes.

studies) -2016[11]		
Uludag et al. – a structured literature review of studies detailing the challenges of agile transformation from the perspective of stakeholders. (73 primary studies)-2018[12]	Top 3 challenges include: coordinating multiple teams working on the same project, considering dependencies in integration, and coordination among geographically distributed teams. The studies emphasize on development of teams roles with minimum higher-level roles.	The stakeholder perspective Focuses on development team roles, with only a few higher-level roles included.
Kalenda, M., Hyna, P.,& Rossi,B- developed an action research approach to a meta-review -2018[13]	Conducted literature review based on 8 common features identified for scaling frameworks SAFe and LeSS. Described challenges and successed from each paper	Limitations: Only focuses on the selective frameworks and multiple frameworks are excluded.
Klünder et al. take a different approach by addressing the challenges involved in agile transformation process in a question-answer way-2019[14]	There is no model which can transform large organisations into agile. Transformation should be started with a small-scale team and then carried forward depending on the feedback and experience of these small scale teams. Management commitment, training, knowledge and risk planning are the preconditions required for transformation processes. Work distribution and setting up of proper infrastructure are the key steps	Not reported
Ozkan&Tarhan- a study undertaken to report challenges faced during transformation -2019[15]	Reports newer challenges like physical dependencies, fragmentation, and feudalism, narrow focus on product, construction, and bottlenecks from one to many.	Not reported
Shameem et al.- a study that reports success factors for large scale agile projects-2017[16]	Success factors were classified into major categories: client and vendor, defining success factors from the clients and vendors end separately.	Not reported
Saeeda et al.- a study defining different challenges for agile scalability-2015[17]	Important challenges reported in the study: documentation, time period issues, budget overflow, communication challenges, distributed teams.	Not reported

- In a recent survey-based study undertaken by Kettunen et al., it was observed that during difficult times, especially the pandemic era of COVID-19, agility and resilience have been required to continue and even to survive. Majority of the survey respondents were software professionals from different industrial sectors, and they stated that during this time the companies had improved agility and this agility helped them to respond to the changing demands and needs of the turbulent times. From this study it was concluded that truly agile companies have been more successful in continuous innovation. [18]
- In a case study reported by Prasetya et al. in the year 2020, it was found that the distributed scrum method is better than the waterfall method in terms of project's success from the point of view of time, scope and cost. In this study, distributed scrum model was developed by using the standard points from Scrum method with a new model based on Scrum of scrum and holonic view approach. [19]
- In an empirical survey based study carried by Uludag et al., in the year 2021, the benefits and challenges of adopting agile scaling frameworks, from 15 scaling frameworks, were described. As per their survey results, there were 30 claimed benefits of adopting agile scaling framework, including improved productivity, reducing headcount and fostering servant leadership, which were not previously reported. There were 22 claimed challenges out of which the challenges like using frameworks as cooking recipes and using frameworks without understanding for what reasons they should be applied, were the ones not reported in any previous studies. [12]
- In a recent work, published in the year 2018, Mohammed et al., proposed a new model for software development, known as the AZ- Model, which tried to overcome the limitations of traditional models and hence impacts the production of a quality product in a timely manner. This development lifecycle is broadly divided into three phases: customer involvement phase, development phase and release phase. Before the development phase there is also a brief testing phase during the design phase in order to minimize the risks. The customer involvement in the process makes it very effective to mitigate risks with changing requirements. In order to test the efficacy of the proposed model, a survey was carried out across the users and the statistical results of the survey revealed that the model is extremely effective for software organizations to produce a quality product within a given time and budget.[20]

- Nawaz et al., proposed an enhanced FDD process model in their recent work published in 2021. A feature driven development is an agile methodology which suffers limitations like : weak documentation, lack of communication control with client and within team members, extensive modeling activity, less iteration among the phases and stakeholders, poor handling of change requirements and rapid modifications, time consuming process. In order to overcome these challenges, an extended model of FDD is proposed which is called Enhance Feature Driven Development (EFDD) is proposed in this article and is deemed to be suitable for implementation in small, medium as well as large scale organisations. [8]

- Mokhtar et al., in their recently published work (2022), produce a comparative case study for waterfall and agile management at a well-known insurance company in Saudi Arabia. The results of the survey included in the study clearly demonstrated that out of the two, the agile methodology can be safely considered as the most suitable and useful methodology as it helps the managers to be flexible and hence efficient to complete the project qualitatively. [21]

- Barbareschi et al., in their recent work " Scrum for safety: an agile methodology for safety-critical software systems" , proposed a novel agile methodology for the development and innovation of safety-critical systems, namely S4S, and validated the applicability of the proposed methodology over a real case study from the railway domain. The reported case study highlighted that S4S (i) enables iterative and evolutive development of safety-critical software, even if architecture and/or requirements need to be refined, (ii) allows documentation to be produced – and kept updated – as an output of the entire process, and (iii) makes the entire process much safer and more reactive w.r.t human errors. This indicates that the agile mindset remains effective in a critical context if it embraces all its values in terms of quality.[22]

- Stettina et al., conducted an empirical survey-based study to know the impact of agile transformations on organisation. The survey was carried out across different countries and organisations and about 134 responses from 29 countries were obtained. Based on the result of this survey as well as past studies, it was observed that agile transformations had positive impact on different dimensions of organisation like productivity, responsiveness, quality, workflow health, employee satisfaction and engagement. Overall they concluded that agile transformations had a strong organisational benefits, specially impacting the dimensions mentioned above. [23]

- Mishra et al., on the other hand, conducted a similar study in 2021 regarding the organisational issues in embracing agile methods in an organisation. The study identifies that absence of suitable culture can hinder the adoption of agile methods in an organisation. The absence of management support also can have adverse impact on adoption of agile methods. Apart from these challenges, a large organization size and traditional organizational culture can also negatively impact the adoption of agile approach. [24]

IV Results And Discussion Of Our Empirical Survey:

As we have already seen in the earlier sections, software development lifecycle (SDLC) is a set of predefined tasks performed at every step of software development which consists of various phases like planning, requirements analysis, design, implementation, testing, deployment and maintenance. We have also reviewed the different methodologies that can be used for software development. The advantages and limitations of these methods i.e. traditional and agile methodologies have been discussed in great depth in the previous section. From all these details we definitely know that traditional methods have their own advantages and newer methods i.e. agile methods have their own advantages. Theoretically, every organisation would want to exploit the advantages of both the methods in the best possible way to gain high quality results at the same time being time and cost efficient.

In order to support our understanding related to the topic and to gain better understanding of the on-going market trends, we carried out a small survey across different IT companies to know which model/methodology is preferred in the real world. For this survey, we used some basic questions like the details of the organisation and the representative details, type of software development model used in the organisation, reason for using software development methodology and parameters considered while selecting the particular methodology. We got responses from 47 different organisations, and we have tried to analyse the results that we obtained from the survey. Out of all the responses, we did not include responses from 02 organisations due to incomplete data. Almost every organisation reported that they did not use a specific single methodology for software development.

Almost all organisations reported that they used multiple methodologies for the development, and they could be both traditional as well as agile methods and sometimes a combination of both. There were 3 companies who reported using only the traditional model (waterfall model) specifically for the development purposes, the reason being that it is easy and convenient to use. There were 12 organisations who reported that they still used the traditional waterfall model along with the agile methodologies for their software development. 31 organisations reported using only agile software development methodologies like scrum development methodology, feature driven development, Extreme Programming. The graph below represents the number of organisations using different software development methodologies as mentioned above. Our results indicate that majority of organisations nowadays, use agile methodologies for the development purposes. There are very few organisations who still use the traditional models in conjunction with the agile methodologies.

The other part of survey focussed on the different parameters that were taken into consideration for selecting different methods of software development. As per the responses received from different companies the parameters which were taken into consideration are : Project Focus; Number and type of teams working on the project; Allotted budget; Flexibility of timeline; Resources needed versus resources available; Feasibility study; Stakeholders; Skill Set of Employees who will work on projects, Expected end product of the project, Customers/stakeholders involvement, Scale of the Development, Complexity of projects; End-Users, level of expertise required. 32 organisations reported "project focus' being an important criterion for the selection of software development method in an organisation. 27 organisations reported "Expected end product of the project" being the second most important parameter for selection. The parameter "flexibility of timeline "was the third most important parameter, reported by 24 organisations. Allotted budget was the parameter reported by 17 organisations. "Number and type of teams" was reported by 19 organisations. The figure below is a graphical representation of the above-mentioned data.

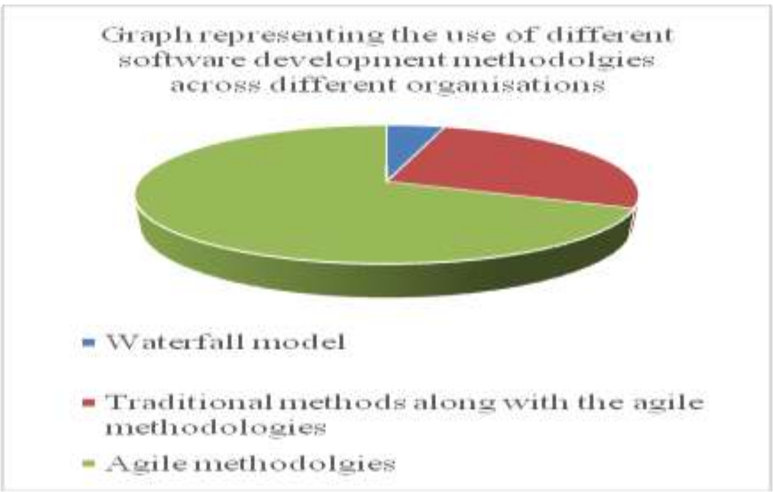

Fig 4: Graph representing usage of software development methodologies

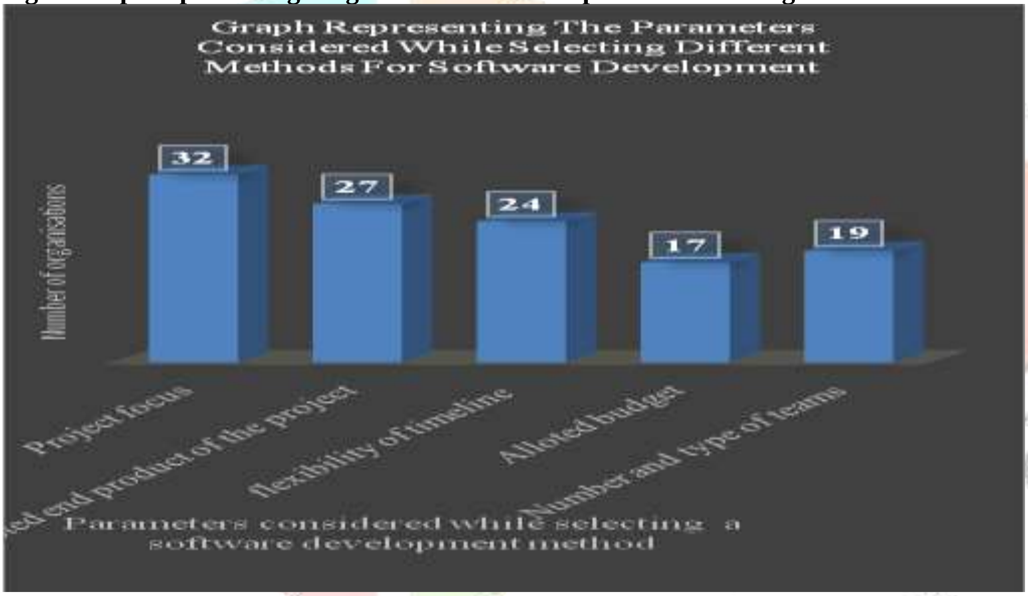

Fig 5: Graph representing parameters considered while selecting software development methodologies

The results of our survey primarily focus on two important aspects: 1. there is no one method that an organisation can adopt for all the projects. An organisation needs to take several parameters into consideration while selecting a method of software development, the key parameters of which are: Project focus, expected product of the project, flexibility of timeline. Allotted budget and the number and type of teams working on the project, along with many other parameters. We also feel that depending upon the client requirements and feasibility an organisation can also develop an altogether a new model of working which is termed as hybrid model i.e., using combination of different methods at the same time, in order to gain more efficiency and deliver high quality end product.

V Conclusion:

Software development methods have evolved during these years. With the changing times, changing market requirements, changing ideologies it becomes imperative for companies to adopt a method which not ensures delivery of high-quality software but also ensure consumer satisfaction. The positives of the traditional approach can be combined with the positives of the agile methods and there is also a new method that can be used for software development i.e. Hybrid approach. Depending upon the client requirements and feasibility an organisation can also develop an altogether a new model of working which can use the features of both traditional and agile methodologies.

VI. Bibliography:

[1] Akanksha Mathur and Archana Acharya, "A Comparative Study on Utilization of Scrum and Spiral Software Development Methodologies: A Review," *Int. J. Eng. Res.*, vol. V4, no. 11, pp. 90–94, 2015.

[2] T. Thesing, C. Feldmann, and M. Burchardt, "Agile versus Waterfall Project Management: Decision model for selecting the appropriate approach to a project," *Procedia Comput. Sci.*, vol. 181, pp. 746–756, 2021.

[3] "No Title." [Online]. Available: https://www.spaceo.ca/blog/software-development-methodologies/.[Accessed: Nov 26

2022]

[4] I. Atoum, “A spiral software engineering model to inspire innovation and creativity of university students,” *Int. J. Eng. Pedagog.*, vol. 9, no. 5, pp. 7–23, 2019.

[5] D. Doshi, L. Jain, and K. Gala, “Review of the Spiral Model and Its Applications,” *Int. J. Eng. Appl. Sci. Technol.*, vol. 5, no. 12, pp. 311–316, 2021.

[6] T. Dingsøyr, T. Dybå, and N. B. Moe, “Agile software development: Current research and future directions,” *Agil. Softw. Dev. Curr. Res. Futur. Dir.*, no. June 2014, pp. 1–238, 2010.

[7] M. M. Müller and W. F. Tichy, “Case study: Extreme programming in a university environment,” *Proc. - Int. Conf. Softw. Eng.*, vol. 2001-May, no. October, pp. 537–544, 2001.

[8] Z. Nawaz, “Proposal of Enhanced FDD Process Model,” *Int. J. Educ. Manag. Eng.*, vol. 11, no. 4, pp. 43–50, 2021.

[9] Shahbaz Ahmed Khan Gahyyur, Abdul Razzaq, Syed Zeeshan Hasan, Salman Ahmed, Rafi Ullah, “Evaluation for Feature Driven Development Paradigm in Context of Architecture Design Augmentation and Perspective Implications,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, 2018.

[10] H. Edison, X. Wang, and K. Conboy, “Comparing Methods for Large-Scale Agile Software Development: A Systematic Literature Review,” *IEEE Trans. Softw. Eng.*, vol. 48, no. 8, pp. 2709–2731, 2022.

[11] C. Dikert, K., Paasivaara, M., Lassenius, “Challenges and success factors for large-scale agile transformations: A systematic literature review.,” *J. Syst. Softw*, no. 119, pp. 87–108, 2016.

[12] F. Uludag, Ö., Kleehaus, M., Caprano, C., Matthes, “Identifying and structuring challenges in large-scale agile development based on a structured literature review.,” in *2018 IEEE 22nd International Enterprise Distributed Object Computing Conference (EDOC)*, 2018, pp. 191–197.

[13] B. . Kalenda, M., Hyna, P., Rossi, “Scaling agile in large organizations: Practices, challenges, and success factors.,” *J. Softw. Evol Process*, p. e1954, 2018.

[14] K. . Klünder, J.A.C., Hohl, P., Prenner, N., Schneider, “Transformation towards agile software product line engineering in large companies: a literature review.,” *J. Softw. Evol. Process*, pp. 1–23, 2019.

[15] A. K. . Ozkan, N., Tarhan, “Investigating causes of scalability challenges in agile software development from a design perspective,” in *2019 1st International Informatics and Software Engineering Conference (UBMYK)*, 2019, pp. 1–6.

[16] A. . Shameem,M., Kumar, C., Chandra, B., Khan, “Systematic review of success factors for scaling agile methods in global software development environment: a client-vendor perspective.,” in *2017 24th Asia-Pacific Software Engineering Conference Workshops (APSECW)*, 2017, pp. 17–24.

[17] F. Saeeda, H., Khalid, H., Ahmed, M., Sameer, A., Arif, “Systematic literature review of agile scalability for large scale projects.,” *Int. J. Adv. Comput. Sci. Appl.*, pp. 63–75, 2015.

[18] T. Kettunen, P., Laanti, M., Fagerholm, F., Mikkonen, T., Männistö, “Industrial agile transformations lacking business emphasis: results from a Nordic survey study.,” *Klotins, E.,Wnuk, K. ICSOB 2020. LNBIP*, vol. 407, pp. 46–54., 2021.

[19] K. D. Prasetya, Suharjito, and D. Pratama, “Effectiveness Analysis of Distributed Scrum Model Compared to Waterfall approach in Third-Party Application Development,” *Procedia Comput. Sci.*, vol. 179, no. 2019, pp. 103–111, 2021.

[20] M. A. Akbar *et al.*, “Improving the quality of software development process by introducing a new methodology-Az-model,” *IEEE Access*, vol. 6, pp. 4811–4823, 2017.

[21] R. Mokhtar and M. Khayyat, “A Comparative Case Study of Waterfall and Agile Management,” *SAR J. - Sci. Res.*, vol. 5, no. 1, pp. 52–62, 2022.

[22] M. Barbareschi, S. Barone, R. Carbone, and V. Casola, “Scrum for safety: an agile methodology for safety-critical software systems,” *Softw. Qual. J.*, no. 0123456789, 2022.

[23] C. J. Stettina, V. van Els, J. Croonenberg, and J. Visser, “The Impact of Agile Transformations on Organizational Performance: A Survey of Teams, Programs and Portfolios,” *Lect. Notes Bus. Inf. Process.*, vol. 419 LNBIP, pp. 86–102, 2021.

[24] A. Mishra, S. Abdalhamid, D. Mishra, and S. Ostrovska, “Organizational issues in embracing Agile methods: an empirical assessment,” Int. J. Syst. Assur. Eng. Manag., vol. 12, no. 6, pp. 1420–1433, 2021.