Implementing Quality Assurance in an Agile Software Development Process

Wumi AJAYI
Software Engineering Dept,
Babcock University, Illisan-Remo, Ogun State, Nigeria.

OLAYINKA Wasiu Olakayode,
Department of Computer Science,
Faculty of Basic Science

Fasina Temilade,
Department of Computer Science,
Faculty of Basic Science

Mariam Gbadegesin,
Department of Computer Science,
Faculty of Basic Science

Abstract:- Extreme programming (XP), Scrum, Lean, and other agile software methods depend on best practices to increase software development quality. Best practices attempt to include software quality assurance (SQA) within the project. According to some agile academics, quality in agile software projects should be a natural result of the used strategy. As a result, agile quality is required to be part of agile software processes. Is it true that many reports tout the benefits of agile approaches for quality assurance? This paper's ambitious purpose is to showcase efforts done to comprehend quality. It summarizes and analyzes research findings in the area of software engineering, specifically for the issue of agile methodologies and software quality.

Keywords:- *Agile, Lean, Programming, Scrum, Software.*

I. INTRODUCTION

To summarize, from the perspective of quality assurance, agile methodology is broad and includes many aspects, including the similarities and differences between traditional quality assurance and agile quality assurance methodologies, the identification and evaluation of quality in agile software development, and the use of test-driven development[1]. In this article, we examine how software quality assurance is achieved in an agile development strategy, taking into account the techniques, processes, and technologies that are utilized to attain the highest possible level of quality in software development.

Agile, in its most basic definition, refers to the capacity to move fast and effortlessly. Agile software development adheres to the spirit of its concept in its most basic form[2]. In corporate application development, agile approaches are most often used because of their ability to quickly respond to changes that arise as a result of both strategic and operational transformations[3]. Because agile makes it simple to adapt to changes in customer requests, it is also straight forward to incorporate such modifications into the Software development life cycle. Because Software Quality Assurance extends across the whole program development life cycle, it is necessary to guarantee that the agile culture and practices are followed in order to assure the quality of the software[4]. Communication between stakeholders is made seamless, resulting in a reduction in the amount of paperwork necessary and, as a result, a reduction in the length of time it takes to complete a software development project.

Agile software development is characterized by short iteration cycles, incremental development, and a focus on people. Agility in software development refers to a collection of software development methodologies that are focused on iterative and incremental development, in which requirements and solutions are developed via cooperation amongst self-organizing, cross-functional teams[5]. Using the examination of software quality assurance that is achieved during agile software testing, the paper addresses testing practice, does an evaluation of the advantages and problems associated with it, and concludes with recommendations for further research. An appropriate research topic and a pilot study with the goal of discovering answers or developing a method to filling a gap in current research will result as a result of this process. Last but not least, the conclusion and future effort involved in agile testing are stated.

A software quality assurance process (often known as software quality assurance or SQA for short) is a continuous procedure that assures that a software product satisfies and adheres to the organization's specified and standardized quality criteria[6]. Software quality assurance (SQA) is a collection of operations that ensures everyone participating in the project has appropriately executed all procedures and processes[7]. SQA is carried out in tandem with software development and is a continuous activity that is carried out throughout the software development life cycle[8]. Instead of doing quality assurance tests once a project is completed, software quality assurance examines the code throughout each development step.

The practices of software testing activities provide firms with the assurance that the software satisfies the criteria and will do all of the duties that have been specified. Testing is a critical activity in software engineering because it allows developers to determine if a system is functioning as intended and to identify and correct any errors[9]. Due to the fact that it gives realistic feedback on the behavior of the system, it is commonly employed in the manufacturing sector for quality assurance. In today's world, software systems are employed to execute important jobs where the margin of error is very small. As a result, these systems must be devoid of errors and of excellent quality. Software testing is a critical technique that may aid in the achievement of software quality assurance objectives. The testing operations carried out by companies consume a significant portion of their budgets.

Development Operations (DevOps), as the term implies, is considered to be a critical component of the Agile style of working since it increases business agility while simultaneously reducing frictions between the many teams engaged in value generation[10]. Thus, it should come as no surprise that 74 percent of those who answered the poll had already began using DevOps in pilot projects or in at least one of their teams[10]. The upshot of this is that many company approaches and methodologies are now adopting an agile methodology.

Many harmful testing approaches are avoided in Agile methodologies because they place a major emphasis on client or user participation. Some of the approaches, like as Xp, give a highly strict set of constructive development practices that are intended to achieve excellent enough quality without the need for further testing other than user acceptance tests, which are the duty of the client[11]. Other agile techniques, on the other hand, do not give such a comprehensive set of activities and realize the necessity for particular testing processes at the integration, system, and acceptance test levels, among other things. For example, by creating the position of an independent tester that tests each finished feature in partnership with the developer, heartbeat QA processes might be further improved[12]. This gives immediate feedback on the attained quality, which is based on the results of independent destructive tests rather than merely the developer's own constructive techniques, and allows for more rapid improvement. Only a few agile approaches include techniques for software testing, and the majority of the activities are carried out by the software developer in the traditional sense. As a result, the issue arises as to whether or not these actions are sufficient to produce a high-quality product. The purpose of this study is to demonstrate the necessity of software testing in the agile development process via the use of an independent software testing team, which will be established.

The following is the outline of the paper's organizational structure: The first chapter provides an overview of the situation and associated efforts. The second chapter describes the difficulties and objectives of this investigation. The approach that was used to perform this study is described in the third section of the book. Fourteenth chapter discusses Agile software development approaches, Agile development practices in the industry, and some significant difficulties that arise in the industry while implementing agile methodologies in practice. The primary discussion segment is found in the fifth chapter. This section discusses a variety of topics such as the software testing process in agile development and why it is vital to have a distinct testing team. Testing, both automated and manual, industrial practices of agile development, survey findings, and significant concerns encountered by practitioners in industry are all covered. References are provided at the conclusion of the report.

II. PROBLEM STATEMENT

Understanding the similarities and distinctions between traditional quality assurance and agile software development is crucial, as is how quality assurance is used in this process.

The approaches, practices, and tools for using software quality assurance in an agile development methodology are examined in this study.

The capacity for swift and fluid movement is known as agility. The foundation of Agile is software development. Due to its flexibility in responding to both strategic and operational changes, agile approaches are frequently used in the creation of corporate applications. Reacting to shifting market expectations is made easier by agile software development. To guarantee software quality, software quality assurance is employed throughout the software development process. Stakeholder interactions reduce the amount of paperwork needed to complete a software project.

Agile software development is rapid, incremental, iterative, and user-centered. "Agile" refers to a grouping of ways to software development that stress iterative, incremental progress, including cross-functional communication within agile teams.

The main topic of this paper is software quality control in an agile setting.

The methodologies, advantages, and difficulties of agile software testing are examined in this article. New research questions and a pilot study to uncover solutions or a strategy to fill the knowledge gap will be the outcome. The outcomes and future goals are then summed up.

A continuous process called "software quality assurance" makes sure that the software product complies with the organization's quality standards (SQA). SQA is a collection of protocols and practices that ensures each person working on the project complies with them.

Throughout the whole software development life cycle, SQA runs in parallel with software development. Software quality assurance looks for problems before the development process is finished, as opposed to waiting until it is finished. What Exactly Is Software Quality Assurance, and Which Techniques Work the Best? See [2] as well.

Software testing may contribute to increasing trust in the software's ability to satisfy business needs. Software engineering needs testing to make sure the system functions as intended and to spot any problems. Since it correctly depicts system behavior, it is helpful for industrial quality assurance.

As a result, software systems are being employed more frequently to complete crucial jobs accurately. This suggests that the program must be faultless and of the highest caliber. Software quality assurance depends on software testing. Businesses invest a lot of money on testing and evaluation.

Due of its increased business agility and less team friction, DevOps (short for Development Operations) is frequently recognized as a crucial component of Agile working methodologies.

Unsurprisingly, 74% of respondents had utilized DevOps in pilots or on at least one of their teams in the past [3]. The result is a more vibrant workplace environment.

Agile methodologies encourage user participation and steer clear of harmful testing techniques. User acceptance testing are the responsibility of the customer. A rigorous set of constructive development methods are offered by XP and other techniques that are similar to it in an effort to attain sufficient quality without further testing. The need for distinct testing procedures at the integration, system, and acceptance levels is acknowledged by other agile techniques. An impartial tester might collaborate with the developer to test every feature in order to improve Heartbeat QA procedures. This offers immediate feedback on the level of quality attained as opposed to just using the developer's own constructive tactics. Software testing techniques are rarely used in agile methodologies because most developers handle this work themselves. The thought "Is this enough?" appears as the product is being produced. We wish to highlight the importance of software testing in the agile development process with an independent software testing team.

The following is the breakdown of the essay: Detailed background information on the project is provided in the first chapter. The goals and challenges of the research are explained in this chapter. The third chapter provides a description of the research process. The fourth chapter will focus on the industry acceptance and issues of Agile's software development methodology. The majority of the discussion in the book is found in this chapter. This section discusses topics like the agile software testing process and the necessity of a separate testing team. The survey results demonstrate that both automated and manual testing present substantial obstacles to industry practitioners. References are included at the study's end.

- What are the current software testing practices of agile development in industry?
- What can be the effects of separate software testing team in Agile development?
- How and when the manual and automated software testing can be used to attain better results in Agile development?
- The Roles of Quality assurance in software development Life Cycle

III. LITERATURE REVIEW

In order to continuously deliver customer value through working software[14], "Agile is a set of concepts and practices that promote team collaboration and feedback." Agile software development methodologies are built on iterative development. In agile procedures, feedback serves as the primary control mechanism rather than planning.

We are aware that some of the traditional techniques, such as Waterfall and V-Model, aren't the best to use anymore because they aren't flexible enough to accept changes made when products are developed for a variety of reasons. Because of this, the trend and recommendation for SDLC technique is Agile in today's market where items must be supplied "just in time" and value is generated by minimizing waste. [4]

For a long time, the most widely used software development methodology was plan-based. Only one phase may begin before the preceding one has been finished. Using this method may lead to a long-term trajectory for a project as well as several hazards.

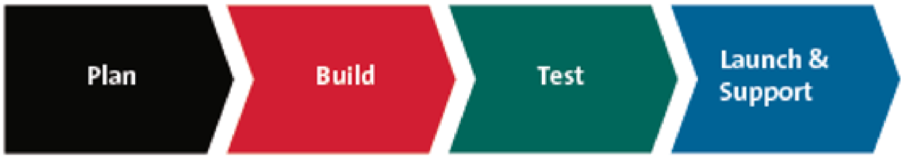

Fig 1. Plan Based Software Development Model

To accept a change in the Waterfall technique, all stages of the software development endeavor must be reconsidered. As a result, the change would need to be evaluated, reviewed, and approved by a CCB before requirements for the change could be developed and a comprehensive design put in place for the change. It may take a long time and need a lot of work to make changes when using Waterfall technique, since formal documentation is required at different stages in software development to examine and record any changes. This shows that while using the Waterfall technique, modifications are typically avoided and even frowned upon because of the possible schedule implications that may be caused by following formal change control methods, as shown could not determine whether Agile is better for small, medium, and large-scale projects[5].

Plan-based software development methodologies were the most popular for a while. Only one phase may start before the one before it is complete. This approach could result in a project's long-term trajectory as well as a number of risks.

IV. METHODOLOGY

A research methodology lays down the rules for how a study will be conducted, how it will be evaluated, and how it will know whether it has been a success. Various methods may be used to perform a study. In which qualitative, quantitative, and mixed techniques approaches are increasingly common.

It's common to see qualitative and quantitative studies compared to one other. Human experience may be better understood and described with the use of qualitative research. It is the goal of the qualitative paradigm to comprehend the social environment via extensive descriptions of cognitive and symbolic acts, as well as the richness of meaning connected with observed behavior, from the point of view of the respondents.

It is common to draw a comparison between quantitative and qualitative research. Mathematical modeling, empirical/statistical survey, and experiment are all components of a quantitative technique. There are two sorts of system experiments: proof of concept and comparison. Experiments might be on simulation or system. Quantitative research focuses on numbers and measurements rather than words in the collecting and analysis of data.

Your research issue may be answered using a combination of quantitative and qualitative methodologies. Because it incorporates the advantages of both quantitative and qualitative research, using a combination of the two may give you a more comprehensive picture.

A variety of research approaches, such as mixed methods, are often used in behavioral, health, and social science research. [6]. Quantitative and qualitative methodologies are combined in a mixed method. Qualitative and quantitative research methods are used in this study. An eclectic, pragmatic, and sensible approach to mixed research that a researcher blend quantitative and qualitative in a manner that best serves the study issue. Mixture of inductive and deductive methodologies, as well as quantitative and qualitative data, are used in mixed research.

In order to accomplish our objectives, we employed a variety of research methods. We've done a lot of research, both quantitative and qualitative. The first stage of our research was a literature review, in which we looked at existing research on the topic of our research questions. The second half is an industry study, in which we investigated a survey in which different software development processes were analyzed to identify their strengths and weaknesses. We looked at the results of a study to see how organizations are really implementing testing in Agile development from a management and technical standpoint when it comes to software testing.

Figure 4 depicts the steps used to arrive at the desired outcomes. We've identified the issue, done some research to have a better understanding of the domain, and determined our scope. After that, we planned out our whole project in relation to the time. The next stage was to choose a technique of research for the study. In order to meet our objectives, we've chosen a hybrid approach. The first step was to do a literature review, which included reading up on what other researchers had said about the topic at hand. The second element is an industry survey, which includes interviews with people in the software business in order to get a better understanding of current practices and to support our notion.

The different methodologies of Agile development are given below:

- Scrum Extreme programming (Xp)
- Feature Driven Development (FDD)
- Crystal Clear Methodology (CC)
- Dynamic Systems Development Method (DSDM)
- Adaptive Software Development

However, the first four strategies have become more common and are employed in the workplace. This chapter will focus on Scrum, Xp, FDD, and CC methods. Most businesses use Scrum and Xp, therefore it's not viable for us to cover all of the procedures for each style in detail. As a result, these four techniques were chosen and explained as succinctly as possible.

- *Scrum*

From the approach used in rugby, the term SCRUM was coined. It entails regaining possession of an out-of-play ball via teamwork. Scrum is a project management framework that was created specifically for managing the development of software[15]. For the implementation phase, this strategy does not focus on particular software development approaches. To put it another way, it instructs management on how to best use team members in order to maximize system adaptability in an ever-shifting business environment. Scrum is based on the premise that a number of the project's technical and environmental elements will change over time. These factors include the needs, finances, time, and technology of each individual. In order for the development team to produce a functional system, it is essential that their method be flexible enough to accept these unforeseen and challenging adjustments. Because Scrum offers a number of regular activities for management to examine the system for any form of faults or obstructions, it may assist firms accomplish better engineering tasks such as software testing.

Using an iterative and incremental approach to software development, Scrum is an agile methodology. Scrum is an agile framework that is meant to give value to the client throughout the project's development[15]. Scrum's fundamental goal is to meet the needs of the customer in an atmosphere of open communication, shared accountability, and continuous improvement. The development process begins with a rough understanding of what needs to be produced and a prioritized list of desired qualities (the product backlog).

Scrum incorporates a variety of system development tasks. This process has a few easy steps. Rapid prototyping in Scrum implies that the system's requirements are obtained in an imperfect form from the beginning. In this approach, the client provides the team with a general idea of what they want[16]. The original criteria are not only insufficient, but they may also be modified throughout the course of the project. Management and development procedures are included in Scrum. Because of this, it helps both the development team and the company's management to better oversee the whole process. Planning begins when all of the needs have been collected. The system is also designed in a straightforward manner. Planning and designing in this manner is unconventional. They have a quick meeting to wrap up the whole process of conceptualizing and creating. Prior to each sprint, the teams plan sprints, identify the backlog items, and allocate them to various teams so that the whole project may be completed in a short period of time. Backlog items are worked on with each sprint[16]. The quantity of work that has to be done is known as a backlog.

There is a defined number of items from a project's backlog that are implemented in each sprint. After each sprint, the team members count the amount of backlog items that need to be implemented, and they also evaluate it. Members of the team gather at the end of each sprint to go through the backlog items they've worked on and those that need to be finished. They also talk about how they achieved their prior assignments and the difficulties they encountered in their last sprint. When they're done, they merge their efforts and deliver it to the client for testing. Having completed acceptance test cases, the customer now tests the system and gives management with comments. This item will be added to the product backlog for the next iteration if there are any bugs or implementation issues left. The product backlog is a list of all the defects, customer requests, and usability and performance improvements that need to be implemented. If the consumer is satisfied with the product they received, no additional development will be done. When the client is satisfied with the release, the development team ends the project.

Teams in scrum meet everyday for 15 minutes. In order to keep the meeting to a minimum, they all stand the whole time. Discuss what each person will do that day, and if someone has an issue during implementation, they may go to other senior team members for advice. They also talk about the obstacles that need to be overcome. Each backlog item is developed, wrapped, reviewed, and adjusted throughout the sprints. It is said that throughout development, the teams create code, test and record the changes they have made to the backlog. They produce executables as part of the wrapping process in order to illustrate the changes. Finally, they aggregate all of the data from the review and update the change to reflect the new items in the backlog and any risks and features.

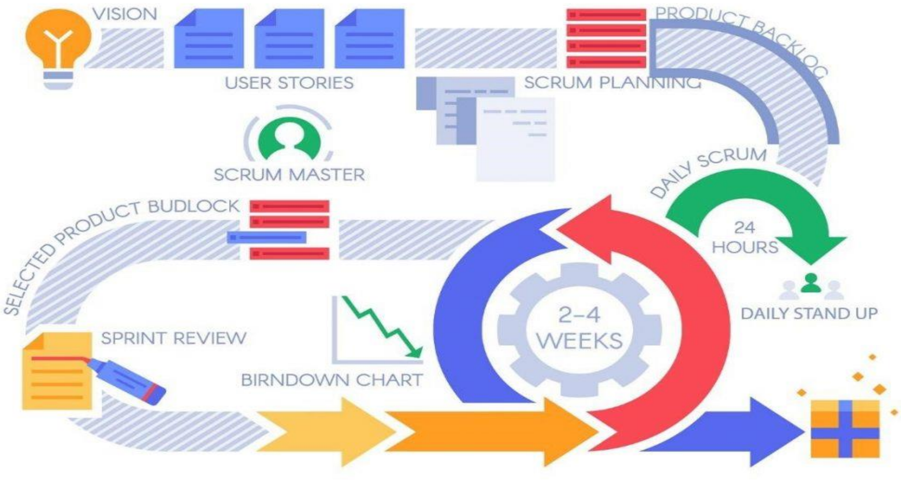

Fig 2: Scrum Pregame and Postgame Psases [17]

In Scrum, the pregame, development, and postgame phases are all shown in Figure 2. In the pregame, the customer's requirements are collected and the system's design is planned. Standards, norms, technology, resources, and architecture all play a role in system planning. Scrum places a great deal of emphasis on the design and architecture of high-level systems. A product backlog is managed based on strategic planning. As new information becomes available, it will be included here. All tasks are ranked and sent to the sprint backlog items based on their degree of importance. During the prototyping phase, the team has sprint meetings to determine the individual duties of each member. Daily meetings are held for coordination and discussion of any concerns that may arise. When a project is in the midst of development, it is in a "closed" phase, meaning no new needs may be added. As part of the sprint process, each phase includes analytic, design and testing and delivery. Sprint review meetings are held after each sprint in order for the team to reflect on their progress. The system is integrated and sent to the client for testing after the development team has completed all of the sprint backlog items. Customer-reported bugs are added to the product backlog for consideration in the next iteration. As a result, the system is created in modest increments, and the customer's input is solicited on a regular basis.

➢ *Quality Assurance Activities in Scrum*

Scrum is an approach to project management that emphasizes iterative, incremental improvements. Rules and practices for management may be found in this technique, which can assist managers structure and improve their operations[15]. The process is not one of engineering with clearly defined actions for quality control. Scrum is a project management framework without any support for testing or development techniques, so management may use whatever activities they see fit to produce a high-quality result. Scrum and Xp are the most popular combo for firms using Scrum. As a result, Scrum helps with project management, while Xp practices influence development. When it comes to testing, a typical tester will have a hard time adjusting to a Scrum project. Scrum has no testing handbook, whereas Xp has some testing information, but it's not enough to call it a guidebook for testers. Scrum organizations use a wide range of quality assurance tasks, including:

Unit tests Continuous testing and improvement Sprint meetings are held on a regular basis. Meetings are held on a daily basis. Code and design standards are adhered to. Acknowledgement testing Automated testing Experimentation with a purpose For a tester, the most crucial parts of SCRUM are the iterative cycle and regular communication. There are two things that the tester has to bear in mind while adjusting these two. During each iteration, the product is tested, rather than at the conclusion of the development process. When the product isn't ready, it's up to the tester to select what needs to be tested. Scrum emphasizes teamwork and cooperation from the outset. As a result, rather of working alone, a tester should instead collaborate closely with the rest of the team. A tester should be present at daily status meetings, which should last no more than 15 minutes, and be given an opportunity to actively participate. To be a good tester, one must collaborate with other testers and find out what to test, rather than just testing from the requirements.

➢ *eXtreme Programming (Xp)*

Because of the lengthy development cycles of traditional and conventional development techniques, extreme programming (Xp) was established. It's a well-known Agile development methodology. It's a collection of various practices, norms, and ideas that have been inherited from earlier methods[18]. One of the key features of Xp is its emphasis on short iterations with frequent feedback and modest releases as well as the constant involvement of customers and the ongoing testing and integration of the code by the whole team. Xp is more successful in smaller and medium-sized businesses because of the importance of the physical environment. There should be at least three to twenty persons in a team, according to Beck[18].

In Xp, a team of three to 10 programmers works together in a single room with monitors facing outward in a circular arrangement[18]. In Xp, there are also tiny changes. During the three-week iteration stage, there are two to five revisions before the product is ready for release. The client provides the specifications, which are then compiled into a specification document. Using index cards, requirements may be documented. A plain index card is used by the customer to jot down short tales. There are frequent meetings between developers and customers to address various parts of the system needs.

➢ *Feature Drive Development (FDD)*

Traditional software development lacks good team communication and cooperation. It lacks several features that allow speedy app development. The technique does not support user modification. This reduces its commercial responsiveness. The FDD technique improves project management and software design, implementation, and documentation uniformity[19]. Unlike Scrum, FDD emphasizes documentation. Before moving on to planning, the general application model is created first, followed by a features list, and then a model for each item on the list. FDD development aligns with the project's model. The FDD has five main phases,

- Develop an Initial Model
- Develop a Feature List
- Plan by Feature
- Construction Phase
- Design by Feature

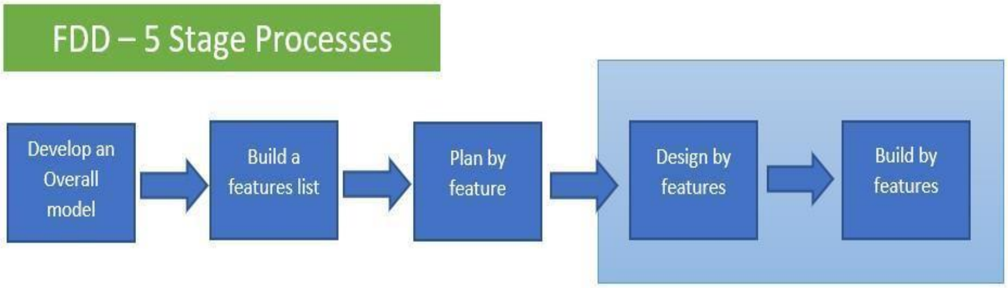

Fig 3: FFD Stages

➢ *Crystal Clear Methodology (CC)*

Alistair Cockburn created the Crystal family of approaches in the mid-1990s. Cockburn's approaches derive from years of research and team interviews. According to Cockburn's study, successful projects were delivered by teams who did not follow formal procedures. The Crystal family is Cockburn's method of recording their contributions to the projects. Crystal techniques are "lightweight methodologies". A crystal is a gemstone whose facets represent multiple perspectives on the "underlying core" of ideas and ideals. The faces represent practices, tools, guidelines, and roles[20].

V. CONCLUSION

After extensive investigation into previous software development methodologies, Agile emerged as the most flexible and adaptable. The strong engagement between the development team and users allows for adjustments requested by users or changes that arise during application development.

External testers might be experts or a test group. Assist software developers and other project team members as needed. Testers should be present at every project meeting to understand the team's aims. While not having a testing team may be advantageous in a small firm with a modest project, it is difficult to produce a quality product in larger enterprises. We decided on a mix of human and automated testing. While automated testing is useful, manual testing is required. The developer may automate testing throughout the sprint, and the tester can test the software before the final release. Pair testing is a valuable activity in agile initiatives. Agile procedures should provide time for test planning. Management is critical in agile. We believe that the project management team can teach and train the project team. This may help them operate in teams. The client should also be educated on why you are utilizing Agile development, since they may be reluctant to collaborate with an Agile team due to their hectic schedules.

REFERENCES

[1]. Arcos-Medina, G., & Mauricio, D. (2019). Aspects of software quality applied to the process of agile software development: a systematic literature review. International Journal of System Assurance Engineering and Management, 10(5), 867-897.

[2]. Gren, L., & Ralph, P. (2022). What Makes Effective Leadership in Agile Software Development Teams?. arXiv preprint arXiv:2201.08058.

[3]. Sarangee K, Schmidt JB, Srinath PB, Wallace A. 2022. Agile transformation in dynamic, high-technology markets: Drivers, inhibitors, and execution. Industrial Marketing Management.

[4]. de Vicente Mohino, J., Bermejo Higuera, J., Bermejo Higuera, J. R., & Sicilia Montalvo, J. A. (2019). The application of a new secure software development life cycle (S-SDLC) with agile methodologies. Electronics, 8(11), 1218.

[5]. Nawaz, M., Nazir, T., Islam, S., Masood, M., Mehmood, A., & Kanwal, S. (2021). Agile Software Development Techniques: A Survey: Agile Software Development Techniques: A Survey. Proceedings of the Pakistan Academy of Sciences: A. Physical and Computational Sciences, 58(1), 17-33.

[6]. Quesada Otarola, D. (2018). Quality management in mobile software development.

[7]. Laaraib, E., Maher, Z. A., Solangi, Z. A., Koondhar, M. Y., Memon, M., Depar, M., & Shah, A. (2021). A Methodology for Incorporating Quality Assurance Practices during Software Development Life Cycle. International Journal, 10(3).

[8]. Abubakar, M., Ahmad, A., & Sama'ila, N. A. Software Quality Assurance Activities.

[9]. Umar, M. A., & Zhanfang, C. (2019). A study of automated software testing: Automation tools and frameworks. International Journal of Computer Science Engineering (IJCSE), 6, 217-225.

[10]. Battina, D. S. (2021). The Challenges and Mitigation Strategies of Using DevOps during Software Development. International Journal of Creative Research Thoughts (IJCRT), ISSN, 2320-2882.

[11]. Dudhat, A., & Abbasi, M. A. (2021). Discussion of Agile Software Development Methodology and its Relevance to Software Engineering. ADI Journal on Recent Innovation, 3(1), 105-114.

[12]. Feijter, R. D., Overbeek, S., Vliet, R. V., Jagroep, E., & Brinkkemper, S. (2018). DevOps competences and maturity for software producing organizations. In Enterprise, Business-Process and Information Systems Modeling (pp. 244-259). Springer, Cham.

[13]. Taskesenlioglu, S., Ozkan, N., & Erdogan, T. G. (2022). Identifying Possible Improvements of Software Development Life Cycle (SDLC) Process of a Bank by Using Process Mining. International Journal of Software Engineering and Knowledge Engineering, 32(04), 525-552.

[14]. Al-Saqqa, S., Sawalha, S., & AbdelNabi, H. (2020). Agile Software Development: Methodologies and Trends. International Journal of Interactive Mobile Technologies, 14(11).

[15]. Soukaina, M., Badr, E., Abdelaziz, M., & Nawal, S. (2021). Towards a New Metamodel Approach of Scrum, XP and Ignite Methods. International Journal of Advanced Computer Science and Applications, 12(12).

[16]. Science and Service Science (NISS), 2011 5th International Conference on New Trends in. 2011: IEEE. O'Regan, G., Software Quality Assurance, in Concise Guide to Software Engineering. 2017,

[17]. https://www.digite.com/agile/scrum-methodology/

[18]. Merzouk, S., Elhadi, S., Ennaji, H., Marzak, A., & Sael, N. (2017). A Comparative Study of Agile Methods: Towards a New Model-based Method. Int. J. Web Appl., 9(4), 121-128.

[19]. DYBÅ, T., & DINGSØYR, T. POSTPRINT: Empirical Studies of Agile Software Development: A Systematic Review1.

[20]. 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.