International Journal of Core Engineering & Management
Volume-7, Issue-10, 2024 ISSN No: 2348-9510

QUALITY ASSURANCE IN AGILE METHODOLOGY

Bhupinder Paul Singh Sahni
bhupinder.sahni@gmail.com

Abstract

Agile software development methodologies have claimed to improve the quality of software products, faster software deliveries and a better software development approach, since their inception. Many Software Organizations adopting Agile methodology have also accepted that use of the agile approach has greatly optimized how things are done. This white paper talks about how Quality Assurance fits in Agile Software Development methodology, principles of Agile Quality Assurance, how Agile Quality Assurance is different from Quality Assurance in other software methodologies and what factors have led to a better-Quality Software Product under Agile methodology.

Keywords: Software Quality Assurance; Agile methodology; Waterfall Model; Scrum; Continuous Integration/ Continuous Development (CI/CD); Automation Testing.

I. INTRODUCTION

Software now rules the world, and its development is integral to all the IT companies. Software development is a very complex process and requires multidimensional growth. Quality Assurance (QA) engineers play a crucial role in ensuring that the software being developed meets the required standards of quality. There have been many different Software methodologies that have come up over the last few years, like Waterfall model, V-Model, Agile methodology to name a few. There are many software development models available today that an organization can use. The way technology is speeding up with newer innovations, innovative products and growth each year, these organizations do not want a software development model that consumes a lot of time and effort. More time taken to develop software and release it to production would mean increased costs, increased competition and delayed time to market. To mitigate the impact of lengthy development times, most of the business houses are moving towards Agile approach for software development or the iterative development. Agile Software development offers a range of benefits that align with modern business needs, particularly in fast-paced and competitive environments.

With iterative development, shortened development cycles and adoption of Agile methodology, the big question is how to ensure the product developed meets the quality standards of customer requirements and product overall. To ensure software product is of good quality, comprehensive testing is crucial for a stable release. More rigorous testing and quality assurance processes can extend development time but reduce post-release issues. So, there needs to be a balance of how much testing can be done on the product to ensure quality with respect to the time available to release the product. Quality Assurance (QA) has taken different forms and is performed at different steps of Software Development Life Cycle (SDLC), based on the software methodology being adopted by the Company, but the ultimate objective has remained the same i.e. to ensure best quality of the product being delivered. This white paper explains Agile methodology,

International Journal of Core Engineering & Management
Volume-7, Issue-10, 2024 ISSN No: 2348-9510

elaborates the role of Quality Assurance Engineers in agile and the best way to integrate quality assurance at different stages of Agile methodology to ensure a stable, reliable product according to customer's requirements.

II. WHAT IS AGILE METHODOLOGY

Agile methodology is a project management approach that breaks a project into stages. It is an iterative and incremental approach to software development that emphasizes flexibility, collaboration, and customer feedback [1]. Agile divides projects into smaller phases and prioritizes adaptability and responsiveness to change throughout the development lifecycle [2].
Some important characteristics of Agile methodology.

1. **Iterative Development:** Agile projects are divided into small, manageable iterations or time-bound periods called sprints. Each iteration or sprint typically lasts from one to four weeks where potential product increment is achieved which can be released to production].

2. **Customer Collaboration:** Customer feedback is gathered on regular basis throughout the development process, allowing for continuous refinement of requirements and features.

3. **Cross-functional Teams:** Agile teams are self-organizing and cross-functional, comprising members with diverse skills and expertise, including developers, testers, designers, and product owners. This structure promotes collaboration and collective ownership of the project's goals.

4. **Adaptive Planning:** Agile projects adapt to changing requirements and priorities by embracing flexible planning practices and embracing change even at later phases of the Release. Rather than creating detailed plans upfront, Agile teams focus on creating a prioritized backlog of user stories or features and adjust their plans based on feedback and progress.

5. **Continuous Delivery:** Agile teams aim to deliver working software incrementally and frequently, often at the end of each iteration. This allows stakeholders to see tangible progress and provide feedback early in the process in sprint demos, reducing the risk of late-stage changes and ensuring that the software meets user requirements.

6. **Emphasis on Individuals and Interactions:** Agile values individuals and interactions over processes and tools. Effective communication, collaboration, and trust among team members are essential for Agile success.

There are many variations in agile development for different types of projects. The most appropriate agile development model or variation depends on an organization's structure and culture [3]. Some popular frameworks and methodologies within Agile include Scrum, Kanban, Extreme Programming (XP), Lean Software Development, and Feature-Driven Development (FDD) [4]. While each approach has its own specific practices and guidelines, they all share the core principles of Agile and aim to deliver high-quality software efficiently and adaptively.

International Journal of Core Engineering & Management
Volume-7, Issue-10, 2024 ISSN No: 2348-9510

III. PRINCIPLES OF AGILE QUALITY ASSURANCE

Agile development is popular for quality delivery of the project in small deliverables. All agile methodologies do not address all the quality assurance factors. Several methodologies address multiple factors. However, combinations of methodologies are followed to ensure the required quality factors [5]. Depending on the Quality, Cost, Project, Culture, deliverables, etc. requirements of the Company, Agile methodology variation is implemented.
Below are some of the principles of Quality Assurance (QA) in Agile methodology [6]:

1. **Early and Continuous Testing:** Quality Assurance in agile methodology emphasizes starting quality checks early in project cycles, also known as "Shift left approach". This approach allows bugs to be found early in the development lifecycle leading to continuous improvement and aligning with the customer goals. Testing is integrated throughout the software development lifecycle (SDLC), starting from the early stages of development and continuing throughout each iteration or sprint. This ensures that defects are identified and addressed promptly, reducing the risk of costly rework later in the process. Testing should be performed after every code enhancement or bug-fix.

2. **Collaboration and continuous Feedback:** QA teams work closely with cross-functional Agile teams, including developers, product owners, and other stakeholders. Collaboration ensures that quality goals are aligned with business objectives, and everyone is involved in identifying and resolving quality issues. Discussions and providing continuous feedback to make the product better and of quality is the sole purpose.

3. **Automation: A**utomation is key to achieving the speed and efficiency required in Agile development. QA teams leverage test automation tools and frameworks to automate repetitive tests, regression tests, and other types of tests to increase coverage and reduce manual effort. But Automation also comes with a cost and some challenges. It is important to automate areas of testing that are repeatable, take time and are not easily performed manually. However, manual testing should still be performed like in cases of exploratory testing, and to ensure that no edge cases are missed.

4. **Continuous Integration and Continuous Deployment (CI/CD):** Agile teams employ CI/CD pipelines to automate the build, test, and deployment processes. This allows for the rapid and frequent delivery of high-quality software while maintaining stability and reliability.
5. Quality Metrics and Monitoring: QA teams identify potential risks and issues of the project through Quality Metrics like defect density, test coverage, code churn, etc. which help inform decision-making and prioritize quality-related activities.

6. **Cross-Functional and Self-Organizing Teams:** Agile QA teams are self-organizing and cross-functional, comprising members with diverse skills and expertise. This ensures that the team takes ownership of their work, promoting accountability and efficiency as they strive to deliver valuable and top-quality software.

International Journal of Core Engineering & Management
Volume-7, Issue-10, 2024 ISSN No: 2348-9510

IV. QUALITY ASSURANCE IN AGILE DIFFERENT FROM QUALITY ASSURANCE IN OTHER METHODOLGIES

There are various software development models that organizations can choose from, but the most common ones are Agile and Waterfall Models. Each methodology has its strengths and weaknesses. As such, there is no solution for all types of projects. Various factors like the number of people in the team, how inclined is the Team with the change of requirements or the duration of the project should be considered [7]. Quality Assurance (QA) in Agile differs from QA in other software methodologies, such as Waterfall, in various ways [8]. It is easier to adopt quality assurance in the traditional Waterfall model as it covers detailed documentation of the requirements. However, the nature of agile development methodologies differs from traditional software development methodology [9].

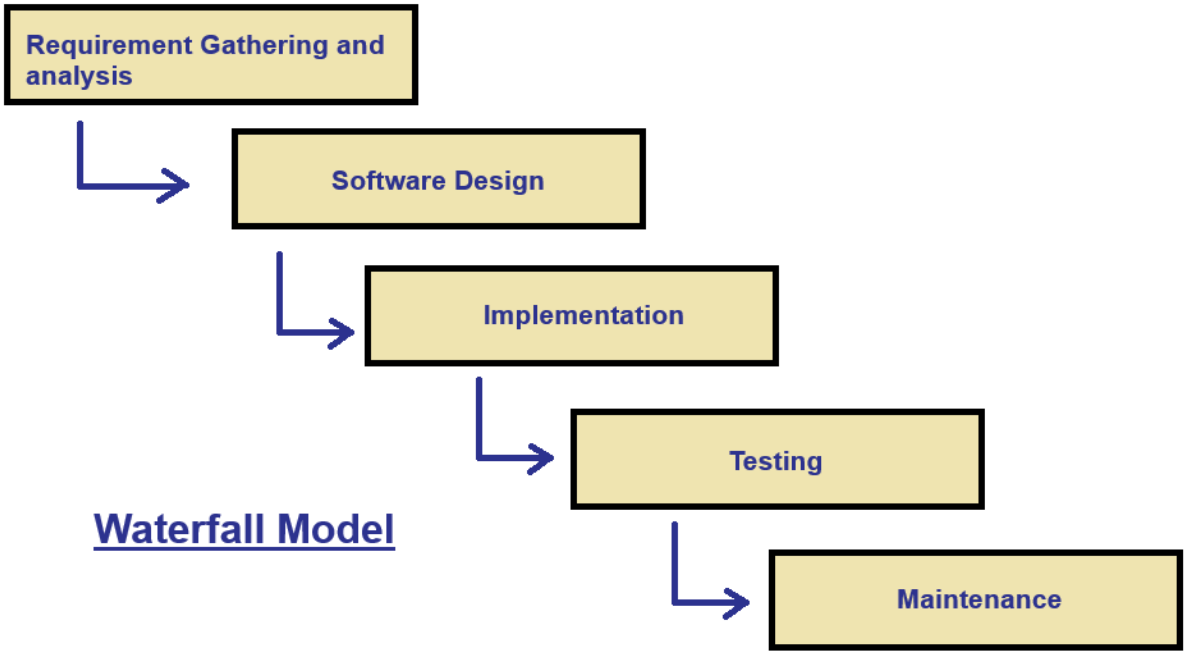

Figure 1: Waterfall Model

1. **Testing phase and Approach:** In Waterfall and other traditional methodologies, QA generally occurs at the end of the development cycle, after development of full functionality/ product is complete. Testing is a separate phase in the end, which means defects are found late in the Release. In Agile, QA is integrated throughout the development process, occurring continuously and iteratively within each sprint or iteration. This allows for early and frequent testing, leading to faster feedback and defect resolution.

2. **Flexibility and Adaptability:** Agile methodologies prioritize adaptability and responsiveness to change. QA processes in Agile are designed to accommodate changing requirements and priorities, allowing QA teams to adjust testing efforts dynamically based on evolving user needs and feedback. In contrast, Waterfall methodologies often follow a rigid, sequential process with little room for flexibility once requirements are finalized.

International Journal of Core Engineering & Management
Volume-7, Issue-10, 2024 ISSN No: 2348-9510

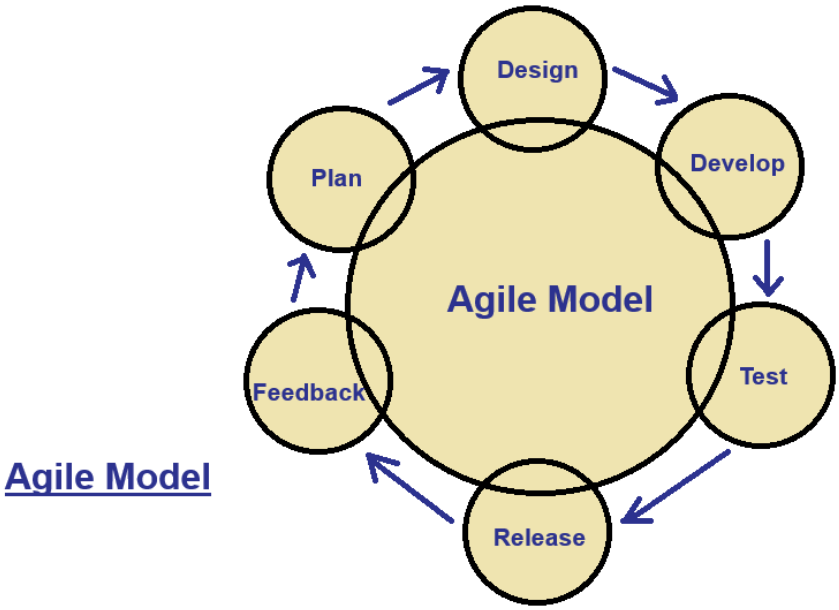

Figure 2: Agile Model

3. **Team Structure:** Agile emphasizes collaboration and cross-functional teamwork. QA engineers work closely with developers, product owners, and other stakeholders throughout the development process, fostering communication, transparency, and shared ownership of quality goals. In Waterfall methodologies, QA teams mostly is a separate team, with limited interaction with other project stakeholders until the testing phase.

4. **Feedback Incorporation:** Agile encourages a culture of continuous improvement, where teams reflect on their processes and practices regularly to identify areas for enhancement. QA teams in Agile participate in sprint retrospectives to discuss what went well, what didn't, and how testing processes can be refined for future iterations. This feedback loop helps drive ongoing improvement in QA practices and outcomes.

Overall, QA in other methodologies like the Waterfall model follows a sequential approach with extensive documentation in which testing phase comes in the end after development is complete, whereas in Agile, Testing is an incremental approach which involves concurrent testing with software development, and puts emphasis on flexibility, collaboration and focus on delivering high-quality software iteratively in response to changing requirements and feedback.

V. RESPONSIBILITIES OF QUALITY ASSURANCE ENGINEER IN AGILE

Agile promotes continuous improvement in QA processes and in the project, emphasizing the critical role played by Quality Assurance Engineers in software development. Below are some key responsibilities of Quality Assurance Engineers in Agile [10]:

1. **Collaborative Planning:** QA engineers actively participate in planning sessions, where they collaborate with the product owner, developers, and other team members to understand the user stories and acceptance criteria for upcoming features. They contribute their expertise in quality assurance to help ensure that requirements are well-defined and testable. Any dependencies on the external systems and teams are raised and discussed in

International Journal of Core Engineering & Management
Volume-7, Issue-10, 2024 ISSN No: 2348-9510

the Planning Sessions – Program Increment (PI) Planning, Sprint Planning or Backlog Refinement sessions.

2. **Requirement Analysis and Test Planning:** QA engineers are responsible for designing test plans and strategies for each sprint. They determine what types of testing (e.g., functional, non-functional, regression) are needed, prioritize testing activities, and define the scope of testing based on the sprint goals and user stories.

3. **Test Case Creation:** QA engineers create detailed test cases based on the acceptance criteria defined for each user story. These test cases mention the steps to be executed, the expected results, and any data or pre-conditions required for testing. Test cases are written to ensure software meets the specified requirements.

4. **Test Case Execution:** QA engineers execute test cases throughout SDLC to verify that the software behaves as expected. They perform various types of testing at different stages of SDLC, which includes manual testing, automated testing, performance, security, and usability aspects of the software.

5. **Defect Logging & ReTesting:** During testing, QA engineers identify defects or issues in the software and report them to the development team using defect tracking tools. They collaborate with developers to investigate and resolve defects promptly. After developers fix issues, QA engineers test the Defects again to ensure that the software meets quality standards before the release.

6. **Automation:** QA engineers develop and maintain automated test scripts to streamline testing activities and increase test coverage for future releases. They identify opportunities for test automation, select appropriate tools and frameworks, and integrate automated tests into the CI/CD pipeline to support continuous testing and delivery.

7. **Continuous Improvement:** QA engineers actively participate in sprint retrospectives to reflect on the testing process, identify areas for improvement, and propose actionable solutions. They contribute feedback and insights to help the team iterate and refine their testing practices for future sprints.

VI. CONCLUSION

Rapid changes in requirements and frequent releases can be challenging for testing. Continuous feedback and collaboration are required at every stage to ensure a defect free product to production. Quality Assurance in Agile is much more than just writing the Test Cases and doing Testing. It ensures continuous feedback and iterative testing, with much focus on collaboration. The Agile Quality Assurance process is known for its customer-centric approach. Agile emphasizes regular feedback from customers and stakeholders, which ensures all project requirements are met.

Agile QA practices, such as continuous integration and automated testing, help in detecting defects early in the development cycle which reduces the project costs. While automation testing is useful, exploratory testing is also required to ensure covering areas which automated testing might miss.

International Journal of Core Engineering & Management
Volume-7, Issue-10, 2024 ISSN No: 2348-9510

The waterfall approach may be better for huge projects that require excessive documentation, agile testing is great for small and medium-sized projects with a focus on quality and user experience. Overall, Agile methodology integrates QA into every aspect of the development process, promoting a culture of quality and continuous improvement.

REFERENCES

1. "These are the Differences Between Agile and Scrum, and How They Differ From Waterfall" scrumalliance.org. Available at: https://resources.scrumalliance.org/Article/differences-agile-scrum-differ-waterfall.
2. "What Is Agile Methodology?" wrike.com. Available at: https://www.wrike.com/project-management-guide/faq/what-is-agile-methodology-in-project-management/.
3. Y. Harb, C. Noteboom, S. Sarnikar, "Evaluating Project Characteristics for Selecting the Best-fit Agile Software Development Methodology: A Teaching Case", Journal of the Midwest Association for Information Systems (JMWAIS), vol. 1, no. 1, pp. 34-47, 2015. Available at: http://aisel.aisnet.org/jmwais/vol1/iss1/4.
4. O. Timperi, "An Overview of Quality Assurance Practices in Agile Methodologies", SEMINAR IN SOFTWARE ENGINEERING, pp. 1-10, 2004.
5. M. Sirshar, F. Arif, "Evaluation of Quality Assurance Factors in Agile Methodologies", International Journal of Advanced Computer Science, vol. 2, no. 2, pp. 73-78, 2012.
6. H. Son. "Agile QA Process: Principles, Steps, and Best Practices", Available at: https://www.testrail.com/blog/agile-qa-best-practices/.
7. B. Alexandru, A. Cosmin, S. Catalin, C. Anton, "A study on using waterfall and agile methods in software project management", Journal of information systems & operations management, vol. 13, no. 1, pp. 125-135, 2019.
8. P. Kanthariya. "Agile Vs Waterfall – What's the Difference?", Available at: https://qacraft.com/agile-vs-waterfall-whats-the-difference/.
9. R. Shuib, S. Hassan, "Towards Adopting Software Quality Assurance in Agile Development Methodology", Turkish Journal of Computer and Mathematics Education, vol. 12, no. 3, pp. 2152-2156, 2020.
10. A. Rao. "Agile Testing: The role of QA in the Agile world", Available at: https://testvox.com/agile-testing-the-role-of-qa-in-the-agile-world/.