

AGILE TESTING

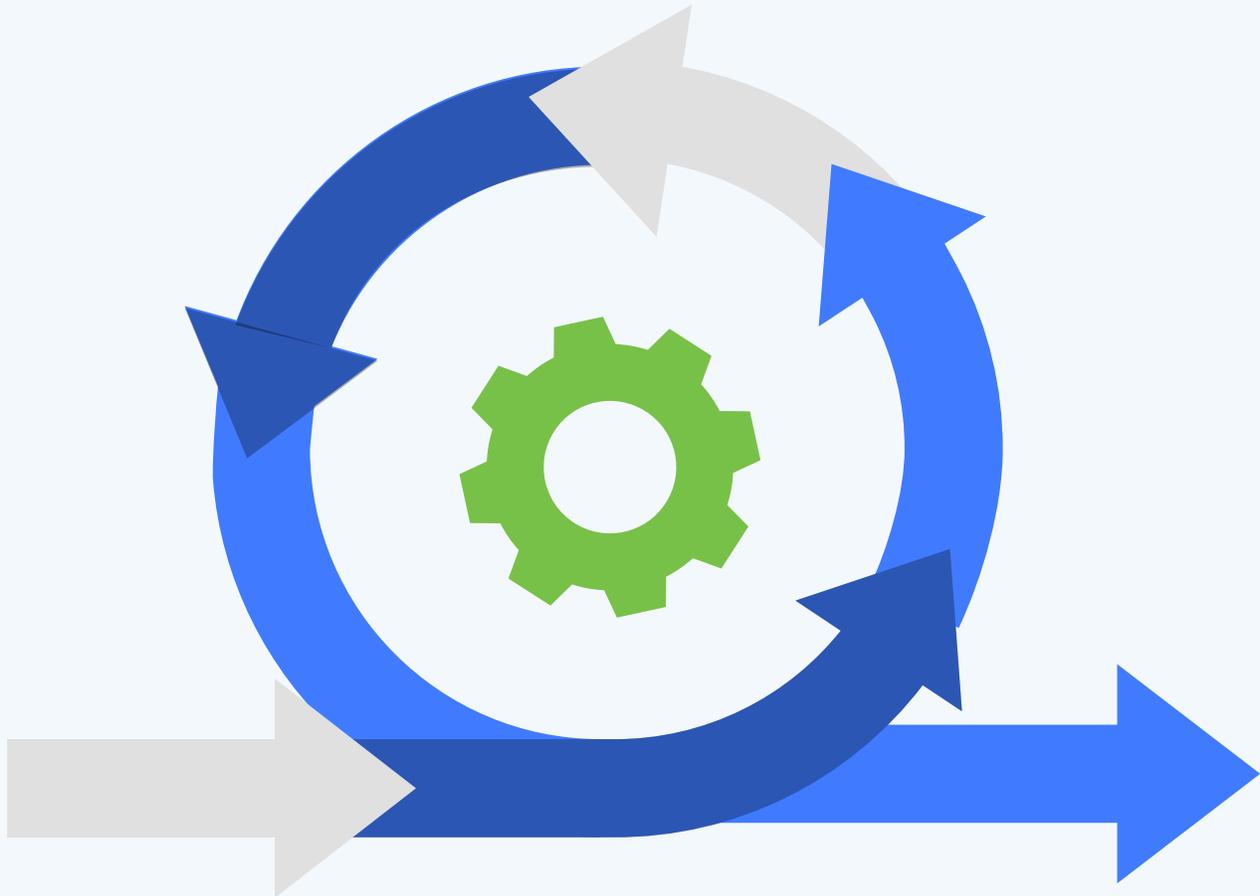
Build, Test, Deploy with Confidence, and Repeat!



OVERVIEW

Agile development is a project management methodology that emphasizes flexibility and collaboration in the software development process. It prioritizes delivering small, working increments of a product rather than a complete product all at once. According to a 2021 survey by VersionOne, the leading provider of agile project management software, 94% of surveyed organizations reported using agile methodologies in some form. Additionally, according to the same survey, 69% of organizations reported using agile as their primary method of project delivery. The key principles of agile development are:

- **Individuals and Interactions:** Agile development values people over processes and tools. Teams work closely together, and communication is key.
- **Working software:** The priority is delivering functional, usable software as quickly as possible.
- **Customer collaboration:** The customer is involved in the development process and provides feedback on the product as it evolves.
- **Responding to Change:** Agile development is flexible and adapts to changes in customer requirements or market conditions.
- **Continuous Improvement:** Agile development is a continual learning process and teams continuously reflect on their work to improve.



Agile development is typically implemented using Agile frameworks such as Scrum, Kanban, or Lean. These frameworks provide specific practices, roles, and rituals for teams to follow and help ensure that the principles of Agile are being followed.

Agile testing, on the other hand, is an integral part of Agile development and plays a crucial role in ensuring the quality of the software being delivered. Here's how Agile testing helps Agile development:

- **Early and Continuous Testing:** Agile testing starts early in the development process and occurs continuously throughout the project. This allows for early detection and resolution of defects, reducing the cost of fixing them.
- **Collaboration Between Developers and Testers:** In Agile development, developers and testers work closely together, allowing for continuous feedback and rapid resolution of issues.
- **Prioritization of Testing:** Agile testing prioritizes testing activities based on the risk associated with each feature. This helps to ensure that critical functionality is thoroughly tested, while less critical areas receive proportionate testing effort.
- **Integration of Testing into the Development Process:** Agile testing is integrated into the development process, allowing for testing to be performed in parallel with development activities. This results in a faster feedback loop and faster delivery of working software.
- **Flexibility and Adaptability:** Agile testing are flexible and adaptable, allowing for changes in requirements or conditions to be easily accommodated. This helps to ensure that the testing process stays aligned with the rapidly evolving software.

Agile testing helps to ensure the quality of the software being delivered, increase collaboration between developers and testers, prioritize testing activities, and provide a flexible and adaptable testing process that keeps pace with the rapidly evolving software.

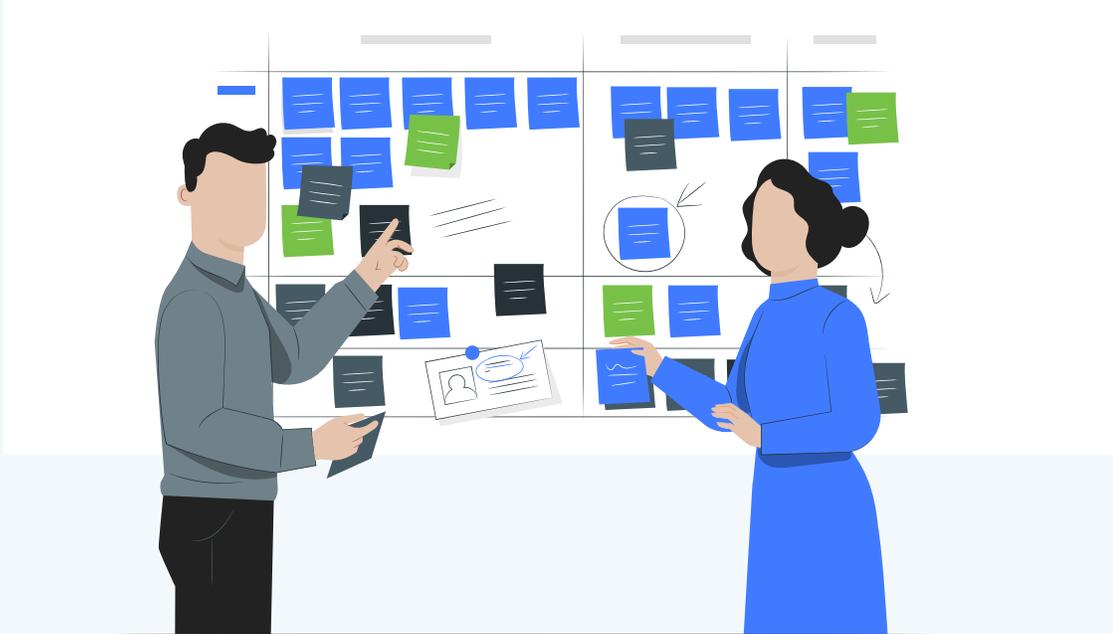
Atlassian Jira is a popular tool used to support Agile development. It provides teams with the ability to plan, track, and release software efficiently and effectively. Here's how Jira can help with Agile development:



- **Project Planning and Management:** Jira provides teams with a centralized location to plan and manage their Agile projects. This includes features such as sprint planning, backlog management, and real-time tracking of project progress.
- **Task Management:** Jira allows teams to break down complex projects into smaller, manageable tasks. These tasks can be assigned to team members, tracked, and updated in real-time.
- **Agile Boards:** Jira provides teams with visual Agile boards that provide real-time insight into project progress. Teams can use these boards to see the status of tasks, prioritize work, and identify any potential roadblocks.
- **Reporting and Analysis:** Jira provides teams with a wide range of reports and dashboards to help track project progress and identify areas for improvement.
- **Customization and Integration:** Jira can be customized to meet the unique needs of each team and can also integrate with a variety of other tools and systems, including continuous integration and deployment (CI/CD) tools, testing tools, and communication tools.

Jira provides teams with a comprehensive toolset to support Agile development. Its project planning and management capabilities, task management, Agile boards, reporting and analysis, and customization and integration options make it an essential tool for Agile teams.

Test management is critical in Agile testing for the following reasons:



- **Improved Efficiency:** Test management helps to organize and streamline the testing process, allowing teams to identify and resolve issues more quickly and efficiently.
- **Better Collaboration:** Test management tools provide a centralized location for teams to collaborate on testing activities, which helps to ensure that everyone is working towards the same goals and objectives.
- **Increased Visibility:** Test management tools provide teams with real-time visibility into the testing process, allowing them to track progress, identify any potential roadblocks, and make data-driven decisions.
- **Improved Quality:** Test management helps to ensure that the right tests are being run at the right time, increasing the likelihood of detecting defects early in the development process. This can result in improved overall quality of the software being delivered.
- **Better Risk Management:** Test management helps teams to prioritize testing activities based on risk, which can help to minimize the impact of defects on the overall project.
- **Better Tracking and Reporting:** Test management tools provide teams with the ability to track and report on testing activities, which can be used to demonstrate the value of testing to stakeholders and improve the overall testing process.

In summary, test management is important in Agile testing as it helps to improve efficiency, collaboration, visibility, quality, risk management, and tracking and reporting. By effectively managing testing activities and teams.

The goal of this whitepaper is to provide an overview of test management for agile teams, including best practices and strategies for effective testing in an agile environment.

WHAT IS AGILE?

One approach that can be used to automate testing is to use a test automation framework. A test automation framework is a set of tools and libraries that can be used to create and execute automated test cases. Some popular test automation frameworks for agile teams include Selenium, and Appium.



IMPORTANCE OF AGILE

The importance of agile lies in its ability to effectively manage projects and deliver products in today's fast-paced and constantly changing environment. Agile emphasizes adaptability, which allows teams to respond quickly to changes and shifts in customer requirements and market conditions. By prioritizing collaboration and communication, agile helps to break down silos and encourages teamwork, leading to better decision making and problem solving. Furthermore, by delivering working software incrementally, agile helps to reduce risk and increase transparency, ensuring that stakeholders are regularly updated on progress and can provide feedback. Ultimately, the use of agile methodologies can lead to improved efficiency, higher product quality, and increased customer satisfaction.

WHY WE NEED AGILE TESTING?

Agile testing is an essential part of the agile software development process as it helps to ensure that the product meets the required quality standards and that it delivers value to the customer. Some of the key reasons why agile testing is necessary include:

-  **Faster Feedback:** Agile testing enables the team to receive early and continuous feedback on the product, which helps to identify and resolve issues more quickly.
-  **Improved Quality:** By integrating testing into the development process, agile testing helps to catch and fix defects earlier in the development cycle, reducing the likelihood of problems later on.
-  **Increased Collaboration:** Agile testing encourages collaboration between development and testing teams, ensuring that testing is aligned with development activities and that issues are addressed in a timely manner.
-  **Enhanced Customer Satisfaction:** Agile testing enables the team to deliver a high-quality product that meets the customer's requirements and expectations.
-  **Better Risk Management:** Agile testing helps to identify and manage risks more effectively by providing regular and frequent feedback on the product's quality and progress.

Agile testing helps to ensure that the software development process is efficient, effective, and delivers high-quality products that meet the customer's needs.

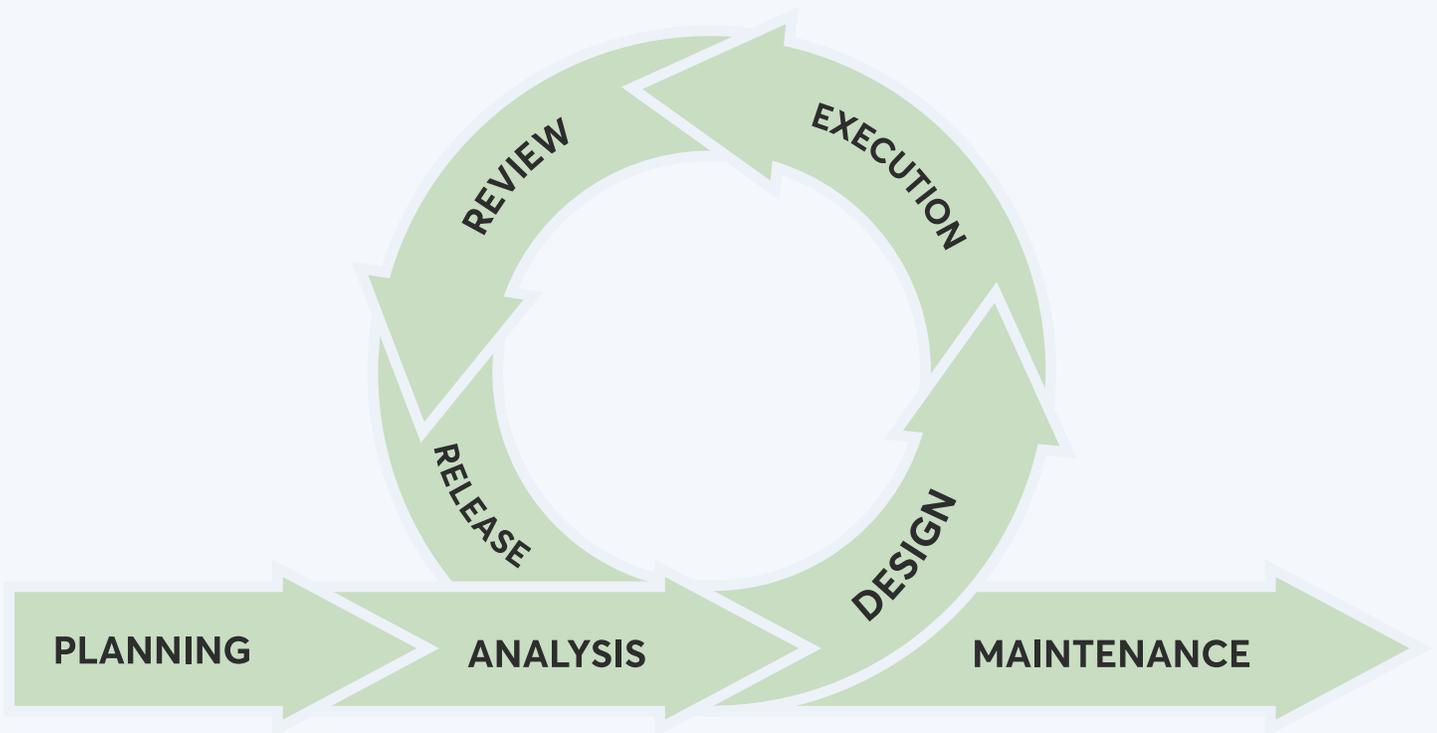
PURPOSE OF AGILE TESTING

The main purpose of Agile testing is to ensure that high-quality software is delivered to the end user in a timely and efficient manner. This is achieved by integrating testing into the development process, ensuring close collaboration between testers and developers, and adopting a flexible and adaptable testing approach. Agile testing is focused on delivering value to the end-user, and it emphasizes the importance of early and frequent testing, automation, continuous integration and delivery, and customer focus.



AGILE TESTING LIFECYCLE

The Agile testing lifecycle typically includes the following phases:



- **Planning:** This phase involves defining the testing goals, creating a test plan, and determining the resources and tools needed for testing.
- **Analysis:** In this phase, testers work with developers to understand the requirements and design tests that will validate the software.
- **Design:** This phase involves designing and implementing the tests, including automated tests.
- **Execution:** In this phase, tests are executed, and the results are analyzed. Any issues are reported and tracked, and the necessary fixes are made.
- **Review:** This phase involves reviewing the test results and reflecting on the testing process. Any areas for improvement are identified and addressed.
- **Release:** In this phase, the software is released to the end-user. This may involve deploying the software to a production environment or making it available for download or use.
- **Maintenance:** This phase involves monitoring the software for any issues, and addressing any issues that are reported.

The Agile testing lifecycle is iterative, meaning that it is repeated for each sprint or iteration of the development process. The goal is to deliver high-quality software to the end-user as quickly and efficiently as possible, and to minimize the risk of delays and issues that could negatively impact the end-user experience.

AGILE TESTING METHODOLOGIES

- **Test Driven Development (TDD):** Test-Driven Development (TDD) is a coding practice that helps developers write new code once an automated test fails. The main goal of TDD is to bring more clarity, simplicity, and accuracy to tests. It begins by designing and developing tests for every small function of an application before the actual coding begins.
- **Behavior Driven Development (BDD):** Behavior Driven Development (BDD) is a software development methodology that focuses on the behavior of the software from the end-user's perspective. It uses natural language constructs to describe the desired behavior of the software and then maps these to automated tests. BDD emphasizes collaboration between development, testing, and business stakeholders to ensure that the software meets the needs of the end-user.
- **Acceptance Test Driven Development (ATDD):** This takes TDD one step further where the entire team of stakeholders discusses acceptance criteria much early in the development process. ATDD is quite ideal for agile testing because it helps close the loops between product and dev teams, increases collaboration and brings efficiency.
- **Exploratory Testing:** Exploratory testing is the simultaneous process of test design and execution as opposed to scripted testing (with preset procedures and processes). Exploratory tests don't follow a precise script decided in advance. In other words, exploratory testing allows testers with the freedom to test the code in an exploratory and somewhat chaotic way. Exploratory tests are then complementary to automated tests because they aim to find possible issues that are beyond the scope of automated tests.
- **Continuous Testing:** Continuous testing is a software testing practice that involves automating the testing process and integrating it into the development pipeline. It enables testing to be performed continuously, at every stage of the development process, from code commit to deployment. The goal of continuous testing is to catch issues early in the development process, when they are easier and less expensive to fix, and to ensure that software is of high quality and ready for release at any time.

- **Pair Testing:** Pair testing is a software testing technique where two testers work together, side by side, to test a software application. One tester, the driver, performs the actions while the other, the observer, looks for defects and offers suggestions. The goal of pair testing is to find more defects than either tester could find working alone, and to improve the knowledge and skills of both testers.
- **Scrum Testing:** Scrum testing is a software testing approach that aligns with the Scrum framework for Agile software development. It involves testing during each sprint, or iteration, of the development process. The goal of Scrum testing is to deliver high-quality software in a timely and efficient manner. In Scrum, testing is integrated into the development process, and both developers and testers are responsible for ensuring that the software is of high quality.
- **Kanban Testing:** Kanban testing is a software testing approach that aligns with the Kanban methodology for Agile software development. It involves testing work items as they are pulled into the development process and completing them within a specific time frame. The goal of Kanban testing is to ensure that software is of high quality and ready for release at any time. The focus in Kanban is on continuous delivery and improvement, and testing plays a key role in ensuring the quality of the software being delivered.



BEST PRACTICES FOR AGILE TESTING

01

Collaboration:

Testers and developers should work closely together and regularly communicate throughout the development process.

03

Automation:

Testing should be automated as much as possible to increase efficiency and speed up the delivery of high-quality software.

05

Flexibility:

Testing approaches should be flexible and adaptable to the changing needs of the development process.

07

Customer focus:

Tests should be designed and executed with the end-user in mind, to ensure that the software meets their needs.

09

Cross-functional teams:

Testing should be performed by cross-functional teams, including both developers and testers, to ensure that software is of high quality.

02

Early and frequent testing:

Tests should be designed and executed early in the development process, and testing should be performed frequently throughout the development cycle.

04

Continuous integration and delivery:

Tests should be integrated into the development pipeline and run automatically every time code is committed.

06

Continuous improvement:

The testing process should be continually reviewed and improved to ensure that it is effective and efficient.

08

Regular retrospectives:

Regular retrospectives should be held to reflect on the testing process and identify areas for improvement.

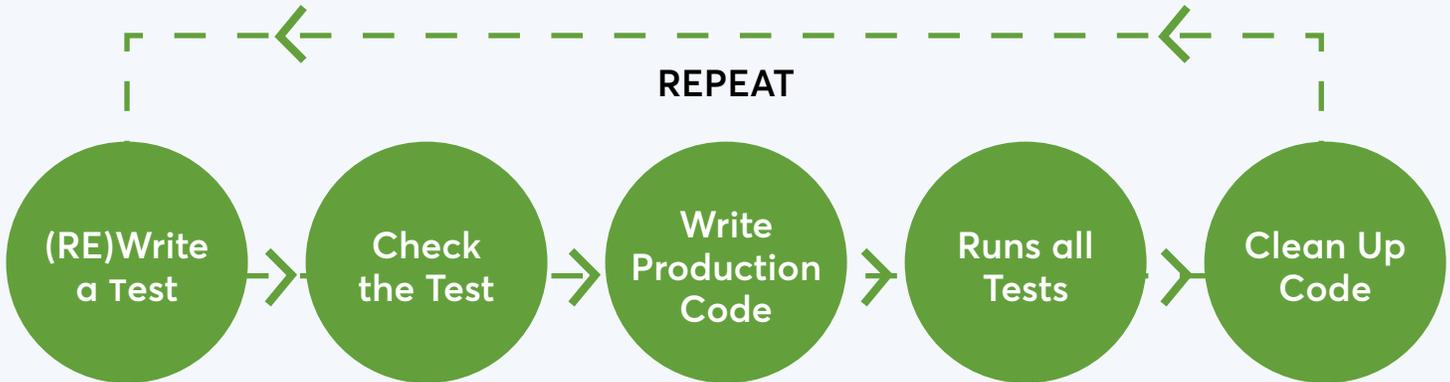
10

Short sprints:

Testing should be performed during short sprints, or iterations, to ensure that issues are identified and addressed quickly.

TEST STRATEGY

One of the key principles of test management for agile teams is to have a test strategy that is closely aligned with the development process. This means that testing should be integrated into the development workflow, with test cases being created and executed alongside the development of new features. This allows for early identification of bugs and issues, which can be addressed before they become major problems.



One approach that can be used to align the test strategy with the development process is to use a test-driven development (TDD) approach. In TDD, developers write automated test cases before writing any code. This helps to ensure that the code is written to meet the requirements of the test cases, and that the code is thoroughly tested before it is released.

Another approach that can be used to align the test strategy with the development process is to use a behavior-driven development (BDD) approach. BDD is a methodology that focuses on describing the behavior of the software from the user's perspective. This helps to ensure that the software meets the needs of the user and that the test cases are closely aligned with the user requirements.

TEST AUTOMATION

Another key principle of test management for agile teams is to have a test automation strategy that is closely aligned with the development process. Agile teams rely heavily on automation to speed up the testing process, and to ensure that the software is thoroughly tested before it is released. Automated tests can be run frequently, with minimal manual intervention, which allows the team to detect and fix issues early in the development process.

One approach that can be used to automate testing is to use a test automation framework. A test automation framework is a set of tools and libraries that can be used to create and execute automated test cases. Some popular test automation frameworks for agile teams include Selenium, and Appium.

ROLE OF ATLASSIAN JIRA IN AGILE TESTING

Atlassian Jira is a popular project management tool that can be used to support Agile testing. It can be used to track and manage the testing process, including:

- Creating and managing testing tasks
- Assigning tasks to team members
- Tracking progress and status
- Adding comments and attachments
- Integrating with other tools for continuous integration/delivery (CI/CD)
- Creating custom workflows to align with Agile testing processes
- Generating reports for testing metrics

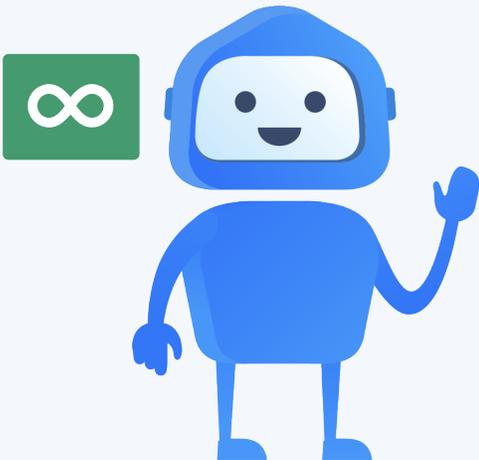


Jira's Agile boards, such as Scrum or Kanban, can be used to visualize the flow of work and prioritize tasks, allowing teams to quickly respond to changes and deliver high-quality software.

QMetry FOR AGILE TESTING

QMetry is a test management tool that can play an important role in Agile testing. It helps organizations to:

- QMetry is an innovative leader in continuous testing offering agile test management, and true codeless automation products for agile and DevOps teams that empower enterprises to build, manage, and deploy quality software at speed with confidence.
- We are **trusted by 500,000+ users globally**, including users from many Fortune/Global 500 across major industry verticals like Hitech, BFSI, healthcare, retail, media, and others.
- QMetry is revolutionizing testing through AI-driven test authoring, test execution, and quality analytics for agile teams globally.
- QMetry is backed by Goldman Sachs Merchant Banking division and Everstone group.
- QMetry Test Management for Jira (QTM4J) is a leading test management app in Atlassian Marketplace that plays an important role in Agile testing. It helps organizations to:



- Plan and manage testing activities within the Agile development process
- Define, organize, and prioritize test cases
- Automate test execution to speed up the testing process
- Track and report on testing progress and results
- Collaborate with team members and stakeholders to improve quality
- Integrate with other tools such as JIRA, Jenkins, and Selenium to provide a comprehensive testing solution

Agile testing solutions have now evolved not only to deliver continuous testing, but more intelligent testing powered by ML and AI. One of the smarter and more efficient ways to test involves the use of predictive and prescriptive QA. Tools led by AI and ML can harness the power of data to anticipate defects even earlier, optimize the test processes, predict outcomes and prescribe recommendations to best the current model.

By using QMetry Test Management for Jira (QTM4J) in an Agile testing environment, organizations can streamline the testing process, improve communication and collaboration among team members, and ensure that quality is integrated into each sprint.

**Start your free trial to
achieve a streamlined and effective
agile testing strategy!**

Start our FREE Trial