**Agile Testing Methodology**

School of MCA, CUK, Kalaburagi

MCA Project, Software testing document - 20 March 2020

Agile testing is a core part of agile software development. Unlike in previous software methodologies, where testing was a separate stage that occurred after development was complete, in an agile methodology testing begins at the very start of the project, even before development has started. Agile testing is continuous testing, which goes hand in hand with development work and provides an ongoing feedback loop into the development process.

Another evolution in agile testing is that testers are no longer a separate organizational unit (there is no “QA department”). Testers are now part of the agile development team. In many cases, agile organizations don’t have dedicated “testers” or “QA engineers”; instead, everyone on the team is responsible for testing. In other cases, there are test specialists, but they work closely with developers throughout the software development cycle

# Introduction

**Agile Testing – Principles, methods & advantages**

As the complexity of software development process is increasing continuously, the software testing approaches needs to evolve to keep up with the development approaches. Agile testing is a new age approach which focuses on testing smarter rather than putting a lot of efforts yet it delivers high-quality products.

The testers and developers need a higher level of collaboration in ***Agile Testing***. The testers have to provide corrective feedback to the development team during the development cycle. This is the age of on-going integration between testing and development approaches.

**Agile testing vs. Waterfall testing**

Agile testing is adopted while working with agile development approach whereas waterfall testing is used in the [waterfall development model](https://reqtest.com/agile-blog/combining-agile-and-waterfall-methodologies-overkill-or-genius-idea/). Below are some high-level

***Differences Between Agile Testing And Waterfall Testing***.

|  |  |
| --- | --- |
| **Agile testing** | **Waterfall testing** |
| Agile testing is unstructured as compared to the waterfall approach and there is minimal planning. | In the Waterfall model, the testing process is more structured and there is a detailed description of the testing phase. |
| Agile testing is well suited for small projects. | Waterfall testing can be adopted for all sorts of projects. |
| As testing begins at the start of the project, errors can be fixed in the middle of the project. | In the waterfall model, the product is tested at the end of the development. For any changes, the project has to start from the beginning. |
|  |  |
| There is very less documentation required for agile testing. | The testing in the waterfall approach requires elaborate documentation. |
| In this approach, every iteration has its own testing phase. The regression tests can be run every time new functions or logic are released. | The testing begins only after the completion of the development phase. |
| In agile testing shippable features of the product are delivered to the customer at the end of an iteration. | In this traditional approach, all features developed are delivered altogether after the implementation phase. |
| Testers and developers work closely in Agile testing. | Testers and developers work separately. |
| User acceptance is performed at the end of every sprint. | User acceptance can only be performed at the end of the project. |
| The testers need to establish communication with developers to analyze requirements and planning. | Developers are not involved in analyzing requirements and planning process. |

**Principles of Agile Testing**

**Testing is continuous:** Agile team tests continuously because it is the only way to ensure continuous progress of the product.

**Continuous feedback-**Agile testing provides feedback on an ongoing basis and this is how your product meets the business needs.

**Tests performed by the whole team:** In a traditional [software development life cycle](https://reqtest.com/testing-blog/what-test-managers-should-know-about-the-software-development-life-cycle/), only the test team is responsible for testing but in agile testing, the developers and the business analysts also test the application.

**Decrease time of feedback response:** The business team is involved in each iteration in agile testing & continuous feedback shortens the time of feedback response.

**Simplified & clean code:** All the defects which are raised by the agile team are fixed within the same iteration and it helps in keeping the code clean and simplified.

**Less documentation:**Agile teams use a reusable checklist, the team focuses on the test instead of the incidental details.

**Test Driven:**In agile methods, testing is performed at the time of implementation whereas, in the traditional process, the testing is performed after implementation.

**Advantages of Agile Testing**

The benefits of the agile testing approach are as follows:

* It saves time and money
* Agile testing reduces documentation
* It is flexible and highly adaptable to changes
* It provides a way for receiving regular feedback from the end user
* Better determination of issues through daily meetings

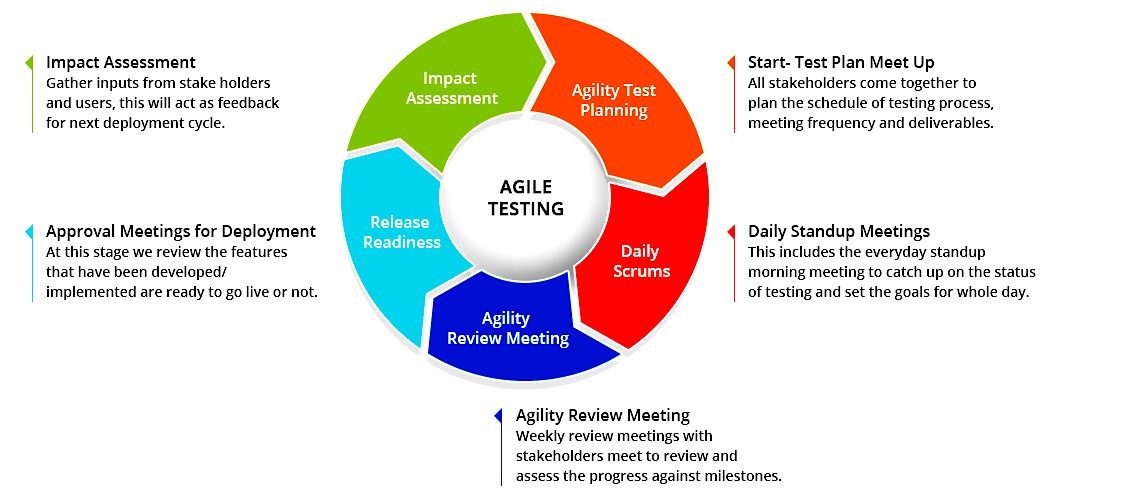
**Test Plan for Agile**

In agile testing, the test plan is written as well as updated for every release. A test plan in agile includes:

* The scope of the testing
* Consolidating new functionalities to be tested
* Types of testing/[Levels of testing](https://reqtest.com/testing-blog/differences-between-the-different-levels-of-tests/)
* Performance & [load testing](https://reqtest.com/testing-blog/load-testing/)
* Consideration of infrastructure
* Risks Plan
* Planning of resources
* Deliverables & Milestones

**Agile Testing Lifecycle**

The agile testing lifecycle includes the following 5 phases:

* Impact assessment
* Agile Testing Planning
* Release Readiness
* Daily Scrums
* Test Agility Review

**Conclusion**

Agile testing not only facilitates the early detection of defects but also [reduces the cost of bugs](https://reqtest.com/testing-blog/how-to-reduce-the-cost-of-bugs/) by fixing them early. This approach also yields a customer-centric approach by delivering a high-quality product as early as possible.