

Agile Testing && TDD

Tests come first in Agile development. When you create a user story, you need to define the acceptance criteria. This drives testing and validation of the user stories

Agenda

What is Agile Testing?

What Does an Agile Tester Do?

Agile testing vs. Waterfall testing

Agile Testing Principles

Test Plan for Agile QA

Advantages of Agile Testing

Methodology

Agile Testing Life Cycle

Agile Testing Qadrants

Agile Testing Methods

Test-Driven Development

Different Kind of Tests

- **Unit tests** – ensure that individual units of the app, such as functions and classes, work as expected. Assertions test that said units return the expected output for any given inputs
- **Integration tests** – ensure that unit collaborations work as expected. Assertions may test an API, UI, or interactions that may result in side-effects (such as database I/O, logging, etc...)
- **End-to-end tests** – ensure that software works as expected from the user's perspective and that every unit behaves correctly in the overall scope of the system. Assertions primarily test the user interface

What is Agile Testing?

Agile testing is software testing that follows the best practices of Agile development

For example, Agile development takes an incremental approach to design. Similarly, Agile testing includes an incremental approach to testing. In this type of software testing, features are tested as they are developed.

What Does an Agile Tester Do?

QA is everyone's responsibility in Agile. So, Agile testers and developers need to work closely together. Communication and collaboration are key.

Agile development is often driven by tests. Developers use Agile testing methods like TDD (test-driven development) to write the test first. Then they write the code that will be verified by the test

What Does an Agile Tester Do?

Once development and testing are underway, communication remains important. Agile testers should be testing as developers write code. Plus, developers will probably do some testing. And Agile testers will probably do some coding.

In Agile development, the definition of done is a shared, standardized understanding among the team that a particular user story has been completed.

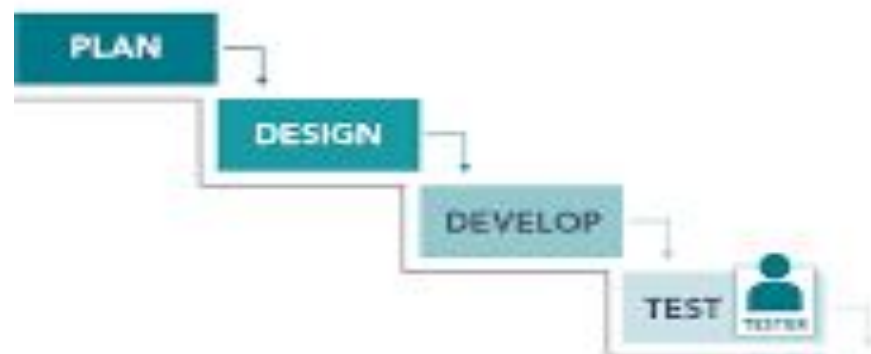
What Does an Agile Tester Do?

The acceptance criteria in a user story Are what will help drive the definition of done. If the user story passes the acceptance criteria, it can be considered done. This includes testing or validating the acceptance criteria. So, a test verifies that you've completed the user story.

It's important that both Agile testers and developers know what has been tested and what defects still need to be resolved.

Software Development Life Cycle (SDLC)

Waterfall
Methodology



Agile
Methodology



Agile testing vs. 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.
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.

Agile testing vs. Waterfall testing

Agile testing	Waterfall testing
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.

Agile Testing Principles

There are eight main principles that guide agile testing:

Continuous testing Continuous feedback Involving the whole team

Quick feedback

High-level software quality Less documentation Test-driven

Customer satisfaction

Test Plan for Agile QA

The scope of the testing

Consolidating new functionalities to be tested

Types of testing

Performance & load testing

Consideration of infrastructure

Risks Plan

Planning of resources

Deliverables & Milestones

Agile Test Plan

Project Name	Calculadora	Browser:	Firefox,Google chrome
Test Case ID	12	Version:	1
Written By:	Anilson Monteiro	Description:	
Tested By:	Sandro Fonseca	Tested On:	

Test #	Date	Action	Expected Results	Actual Results	Pass?
1	6-May	Logging in	Should get to home screen	User directed to different page	<input checked="" type="checkbox"/>
2	6-Jul	Signing up	Get confirmation email	Confirmation email received	<input type="checkbox"/>
3	7-Jul	Click divider button	get result of this operation	get correct result	<input type="checkbox"/>
4	7-Aug	Click multiplier button	get result of this operation	get correct result	<input type="checkbox"/>
					<input type="checkbox"/>

Advantages of Agile Testing Methodology

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

Agile Testing Life Cycle

The agile testing life cycle includes the following 5 phases:

Impact assessment

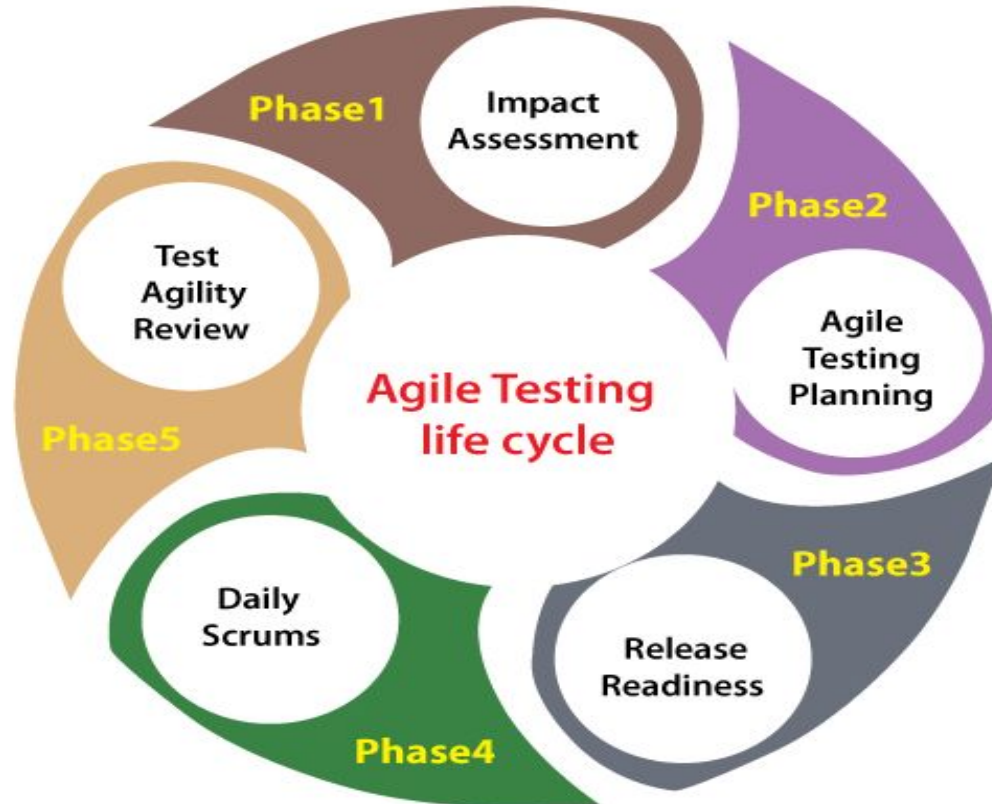
Agile Testing Planning

Release Readiness

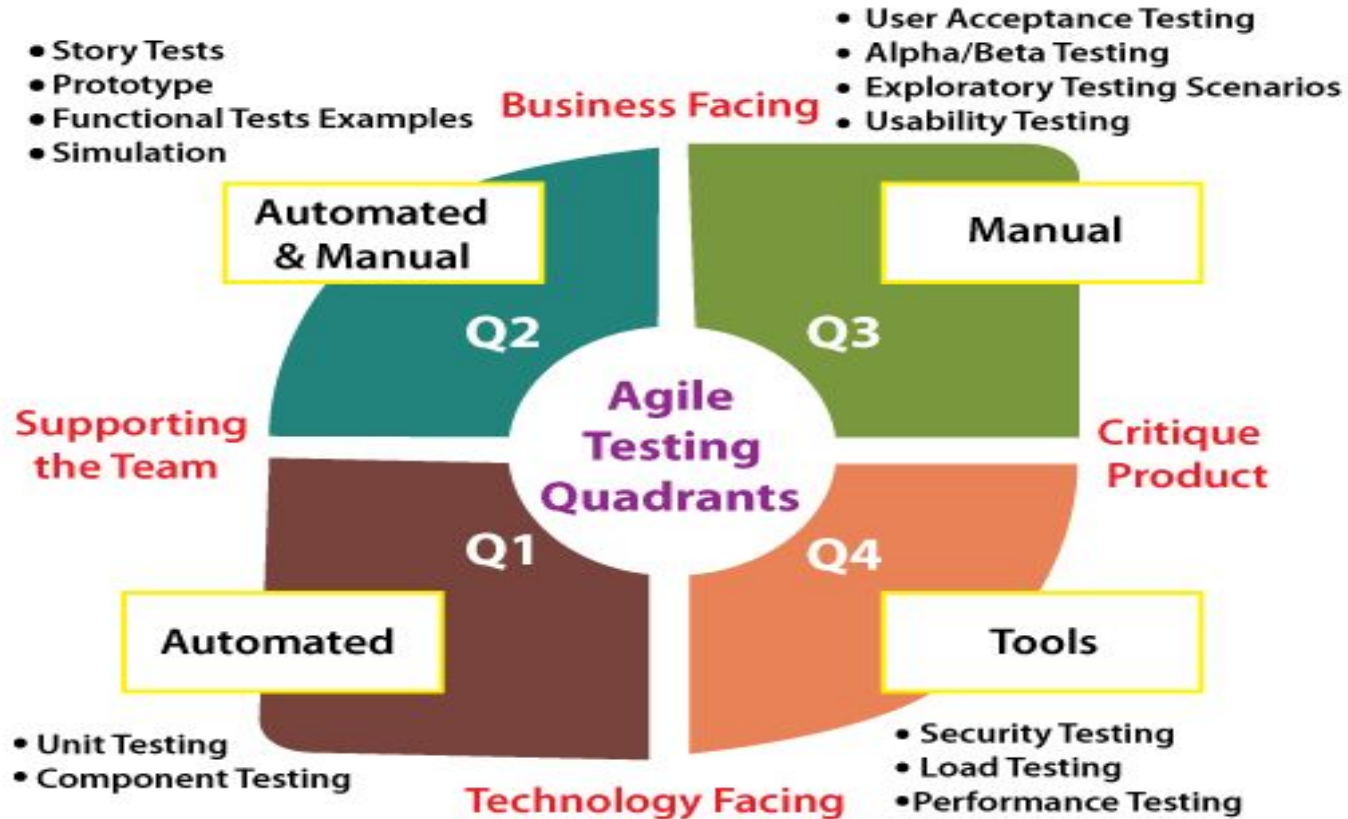
Daily Scrums

Test Agility Review

Agile Testing Life Cycle



Agile Testing Quadrants



Agile testing not only facilitates the early detection of defects but also **reduces 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.

Agile Testing Methods

Behavior Driven Development
(BDD)

Exploratory Testing

Acceptance Test-Driven
Development (ATDD)

**Test-Driven Development
(TDD)**

Test-Driven Development (TDD)

Test-Driven Development is a testing methodology or a programming practice implemented from a developer's perspective.

Is software development approach in which test cases are developed to specify and validate what the code will do. In simple terms, test cases for each functionality are created and tested first and if the test fails then the new code is written in order to pass the test and making code simple and bug-free

How to perform TDD Test

Following steps define how to perform TDD test :

1. Add a test.
2. Run all tests and see if any new test fails.
3. Write some code.
4. Run tests and Refactor code.
5. Repeat.

