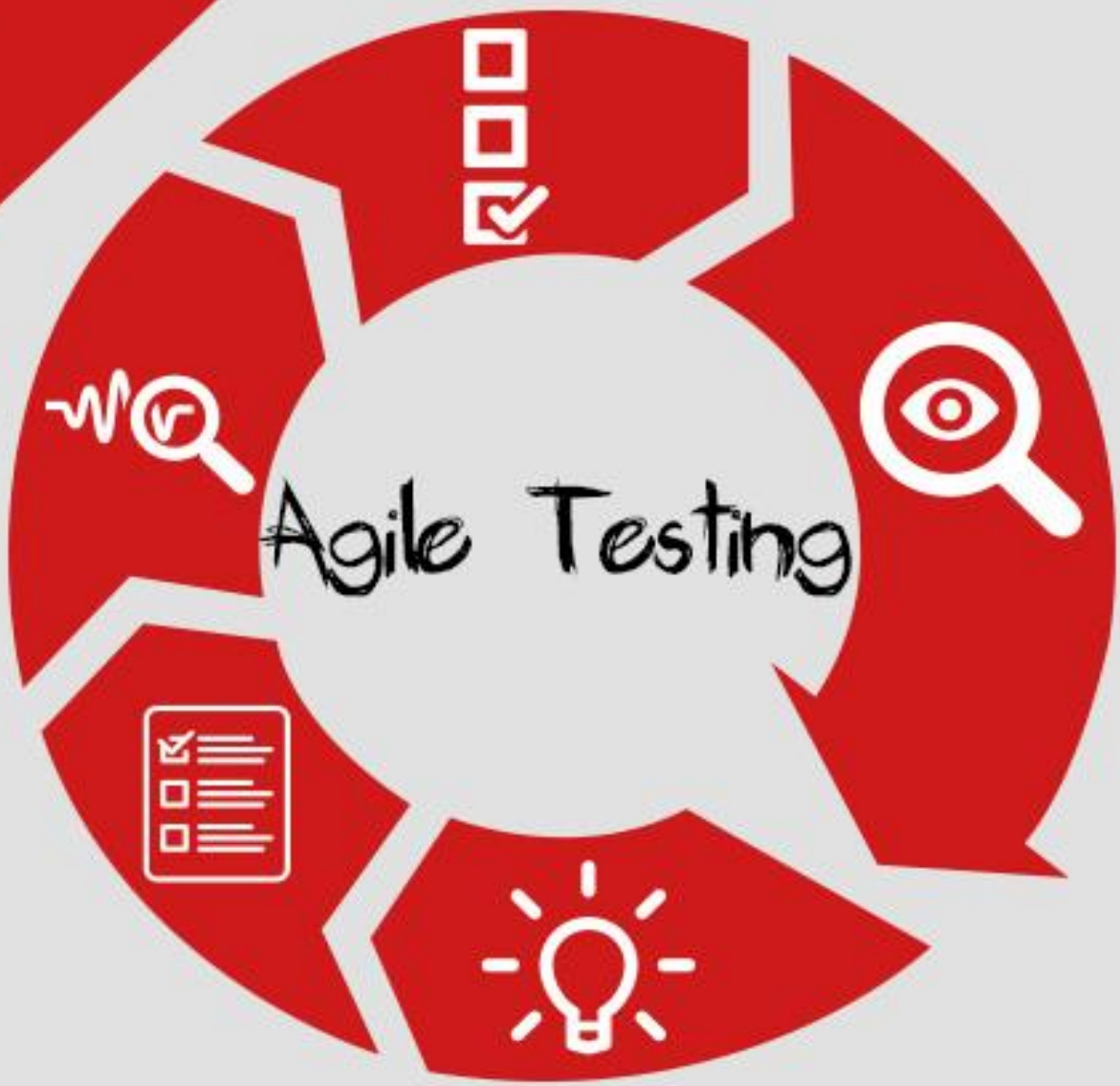


tutorialspoint

SIMPLY EASY LEARNING

www.tutorialspoint.com

https://www.facebook.com/tutorialspointindia https://twitter.com/tutorialspoint

About the Tutorial

Agile Testing is a software testing practice that follows the principles of agile software development.

Agile Testing involves all members of the project team, with special expertise contributed by testers. Testing is not a separate phase and is interwoven with all the development phases such as requirements, design and coding and test case generation. Testing takes place simultaneously through the Development Life Cycle.

Audience

The target audience for this tutorial is Software Testing Professionals, Software Quality Experts, and Software Developers.

Prerequisites

Before proceeding with this tutorial, you should have a basic understanding of software development life cycle (SDLC). A basic understanding of software testing (manual or automation) will be beneficial.

Copyright & Disclaimer

© Copyright 2016 by Tutorials Point (I) Pvt. Ltd.

All the content and graphics published in this e-book are the property of Tutorials Point (I) Pvt. Ltd. The user of this e-book is prohibited to reuse, retain, copy, distribute or republish any contents or a part of contents of this e-book in any manner without written consent of the publisher.

We strive to update the contents of our website and tutorials as timely and as precisely as possible, however, the contents may contain inaccuracies or errors. Tutorials Point (I) Pvt. Ltd. provides no guarantee regarding the accuracy, timeliness or completeness of our website or its contents including this tutorial. If you discover any errors on our website or in this tutorial, please notify us at contact@tutorialspoint.com

Table of Contents

About the Tutorial i
Audience i
Prerequisites i
Copyright & Disclaimer i
Table of Contents ii

1. Agile Testing – Overview 1
Agile Manifesto 1
What is Agile Testing? 1
Agile Testing Vs. Waterfall Testing 2
Agile Testing Principles 2
Agile Testing Activities 4

2. Agile Testing – Agile Methodologies 5
Continuous Integration, Continuous Quality 5
Agile Methodologies 6
Agile Testing Methodologies 7
Agile Testing Lifecycle 8

3. Agile Testing – Tester in Agile Team 9
Role of Tester in Agile Team 10

4. Agile Testing – Tracking Testing Activities 13
Test Progress 13
Product Quality 13
Key Success Factors 14

5. Agile Testing – Significant Attributes 16
Agile Testing Benefits 16
Best Practices in Agile Testing 16
Challenges in Agile Testing 17
Agile Testing Guidelines 18

6. Agile Testing – Quadrants 19

7. Agile Testing in Scrum 23
Collaborative User Story Creation 23
Agile Testing Practices 25

8. Agile Testing – Methods 27
Test Driven Development 27
Acceptance Test Driven Development 27
Behavior Driven Development (BDD) 28

9. Agile Testing – Techniques 29
Definition of Done 29
Test Information 30
Functional and Non-Functional Test Design 30
Exploratory Testing 31
Risk-Based Testing 31
Fit Tests 31

tutorialspoint
SIMPLYEASYLEARNING

10. Agile Testing – Workproducts 32
Test Metrics Reports 33
Sprint Review and Retrospective Reports 34

11. Agile Testing in Kanban 35
Testing Activities in Product Development 35
Story Exploration 36
Estimation 36
Story Planning 37
Story Progression 37
Story Acceptance 37

12. Agile Testing – Tools 38
Agile Test Automation Tools 39

tutorialspoint
SIMPLYEASYLEARNING

1.Agile Testing – Overview

Agile is an iterative development methodology, where both development and testing activities are concurrent. Testing is not a separate phase; Coding and Testing are done interactively and incrementally, resulting in quality end product, which the meets customer requirements. Further, continuous integration results in early defect removal and hence time, effort and cost savings.

Agile Manifesto

The Agile Manifesto was published by a team of software developers in 2001, highlighting the importance of the development team, accommodating changing requirements and customer involvement.

The Agile Manifesto is:

We are uncovering better ways of developing software by doing it and helping others do it. Through this work, we have come to value-

- Individuals and interactions over processes and tools.
- Working software over comprehensive documentation.
- Customer collaboration over contract negotiation.
- Responding to change over following a plan.

That is, while there is value in the items on the right, we value the items on the left more.

What is Agile Testing?

Agile Testing is a software testing practice that follows the principles of agile software development.

Agile Testing involves all members of the project team, with special expertise contributed by testers. Testing is not a separate phase and is interwoven with all the development phases such as requirements, design and coding and test case generation. Testing takes place simultaneously through the Development Life Cycle.

Furthermore, with testers participating in the entire Development Lifecycle in conjunction with cross-functional team members, the contribution of testers towards building the software as per the customer requirements, with better design and code would become possible.

Agile Testing covers all the levels of testing and all types of testing.

Agile Testing Vs. Waterfall Testing

In a Waterfall Development methodology, the Development Life Cycle activities happen in phases that are sequential. Thus, testing is a separate phase and gets initiated only after the completion of the development phase.

Following are the highlights of differences between Agile Testing and Waterfall Testing-

Agile Testing	Waterfall Testing
Testing is not a separate phase and occurs concurrently with development.	Testing is a separate phase. All levels and types of testing can begin only after the completion of development.
Testers and developers work together.	Testers work separately from developers.
Testers are involved in coming up with requirements. This helps in requirements mapping to the behaviors in the real world scenario and also framing the acceptance criteria. Also, logical Acceptance Test Cases would be ready along with the requirements.	Testers may not be involved in the requirements phase.
Acceptance Testing is done after every iteration and customer feedback is sought.	Acceptance Testing is done only at the end of the project.
Every iteration completes its own testing thus allowing regression testing to be implemented every time new functions or logic are released.	Regression Testing can be implemented only after the completion of development.
No time delays between coding and testing.	Usual time delays between coding and testing.
Continuous testing with overlapping test levels.	Testing is a timed activity and test levels cannot overlap.
Testing is a best practice.	Testing is often overlooked.

Agile Testing Principles

The principles of Agile testing are-

- **Testing moves the project forward**: Continuous testing is the only way to ensure continuous progress. Agile Testing provides feedback on an ongoing basis and the final product meets the business demands.

tutorialspoint
SIMPLYEASYLEARNING

- **Testing is not a phase**: Agile team tests alongside the development team to ensure that the features implemented during a given iteration are actually done. Testing is not kept for a later phase.

- **Everyone tests**: In agile testing, the entire team including analysts, developers, and testers test the application. After every iteration, even the customer performs the User Acceptance Testing.

- **Shortening Feedback Loops**: In Agile Testing, the business team get to know the product development for each and every iteration. They are involved in every iteration. Continuous feedback shortens the feedback response time and thus the cost involved in fixing it is less.

- **Keep the Code Clean**: The defects are fixed as they are raised within the same iteration. This ensures clean code at any milestone of development.

- **Lightweight Documentation**: Instead of comprehensive test documentation, Agile testers-

    - Use reusable checklists to suggest tests.
    - Focus on the essence of the test rather than the incidental details.
    - Use lightweight documentation styles/tools.
    - Capture test ideas in charters for exploratory testing.
    - Leverage documents for multiple purposes.

- **Leveraging one test artifact for manual and automated tests**: Same test script artifact can be utilized for manual testing and as an input for automated tests. This eliminates the requirement of Manual Test Documentation and then an equivalent Automation Test Script.

- "**Done Done," not just done**: In Agile, a feature is said to be done not after development but after development and testing.

- **Test-Last vs. Test Driven**: Test Cases are written along with the requirements. Hence, development can be driven by testing. This approach is called Test Driven Development (TDD) and Acceptance Test Driven Development (ATDD). This is in contrast to testing as a last phase in Waterfall Testing.

tutorialspoint
SIMPLYEASYLEARNING

Agile Testing Activities

The Agile Testing Activities at Project Level are-

- Release Planning (Test Plan)
  - For every Iteration,
  - Agile Testing Activities during an Iteration
- Regression Testing
- Release Activities (Test Related)

The Agile Testing Activities during an iteration include-

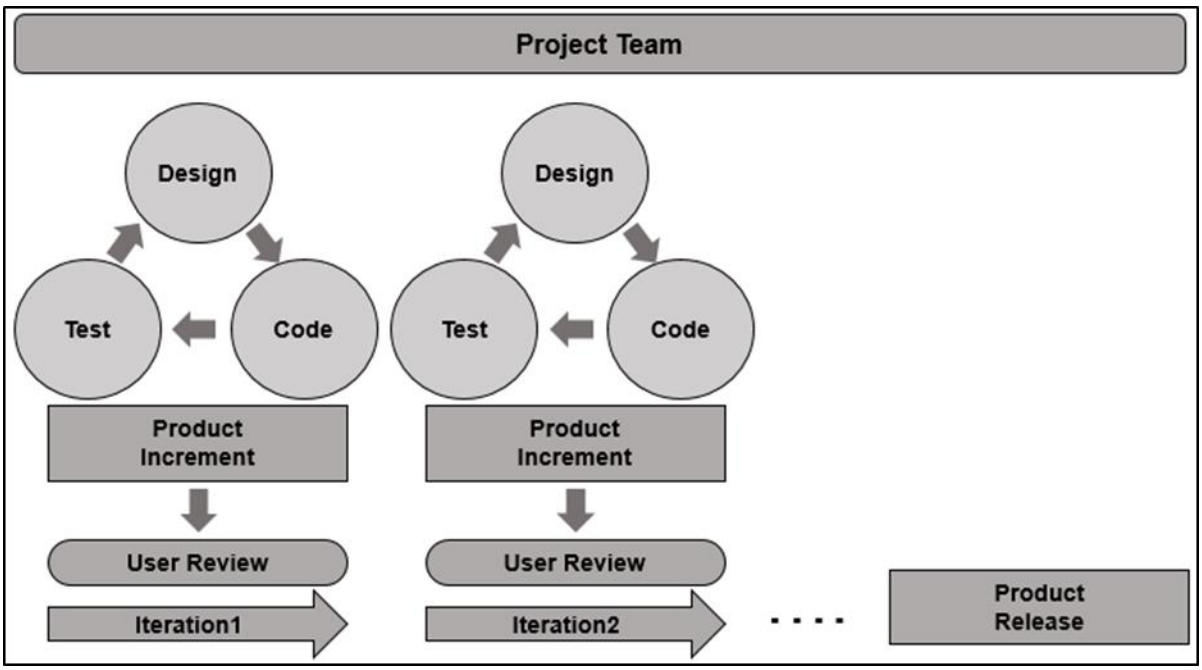
- Participating in iteration planning
- Estimating tasks from the view of testing
- Writing test cases using the feature descriptions
- Unit Testing
- Integration Testing
- Feature Testing
- Defect Fixing
- Integration Testing
- Acceptance Testing
- Status Reporting on Progress of Testing
- Defect Tracking

tutorialspoint
SIMPLYEASYLEARNING

2.Agile Testing – Agile Methodologies

Agile is an iterative development methodology, where the entire project team participates in all the activities. The requirements evolve as the iterations progress, through collaboration between the customer and the self-organizing teams. As Coding and Testing are done interactively and incrementally, during the course of development, the end-product would be of quality and ensures customer requirements.

Every iteration results in an integrated working product increment and is delivered for User Acceptance Testing. The customer feedback thus obtained would be an input to the next / subsequent Iterations.



Continuous Integration, Continuous Quality

Continuous Integration is the key for Agile Development success. Integrate frequently, at least daily such that you are ready for a release as and when required. Testing in Agile becomes an essential component of all the phases of the development, ensuring continuous quality of the product. Constant feedback from everyone involved in the project adds to the quality of the product.

In Agile, communication is given utmost importance and the customer requests are received as and when necessary. This gives the satisfaction to the customer that all the inputs are considered and working quality product is available throughout the development.

tutorialspoint
SIMPLYEASYLEARNING

Agile Methodologies

There are several Agile Methodologies that support Agile Development. The Agile Methodologies include-

Scrum

Scrum is an Agile development method that emphasizes on team-centric approach. It advocates participation of the entire team in all the project development activities.

XP

eXtreme Programming is customer-centric and focuses on constantly changing requirements. With frequent releases and customer feedback, the end-product will be of quality meeting customer requirements that are made clearer during the process.

Crystal

Crystal is based on chartering, cyclic delivery and wrap up.

- Chartering involves forming a development team, carrying out a preliminary feasibility analysis, arriving at an initial plan and the development methodology.
-
- Cyclic Delivery with two or more delivery cycles focuses on the development phase and final integrated product delivery.
-
- During Wrap up, deployment into the user environment, post-deployment reviews and reflections are performed.
-

FDD

Feature Driven Development (FDD) involves designing and building features. The difference between FDD and other Agile Development Methodologies is that the features are developed in specific and short phases separately.

DSDM

Dynamic Software Development Method (DSDM) is based on Rapid Application Development (RAD) and is aligned to the Agile Framework. DSDM focuses on frequent delivery of the product, involving users actively and empowering the teams to make quick decisions.

Lean Software Development

tutorialspoint
SIMPLYEASYLEARNING

In Lean Software Development, focus is on eliminating waste and giving value to the customer. This results in rapid development and product of value.

Waste includes partially done work, irrelevant work, features that are not used by the customer, defects, etc. that add to delays in delivery.

The **Lean Principles** are-

- Eliminate Waste
- Amplify Learning
- Delay Commitment
- Empower the Team
- Deliver Fast
- Build Integrity in
- See the Whole

Kanban

Kanban focuses on managing work with an emphasis on just-in-time (JIT) delivery, while not overloading the team members. The tasks are displayed for all the participants to see and for the Team Members to pull work from a queue.

Kanban is based on:

- Kanban Board (Visual and Persistent across the Development)
- Work-in-progress (WIP) Limit
- Lead Time

Agile Testing Methodologies

The testing practices are well defined for every project, whether Agile or not, to deliver quality products. Traditional Testing principles are quite often used in Agile Testing. One of them is Early Testing that focuses on-

- Writing Test Cases to express the behavior of the system.
- Early Defect Prevention, detection and removal.
- Ensuring that the right test types are run at the right time and as part of the right test level.

In all the Agile Methodologies we discussed, Agile Testing in itself is a Methodology. In all the approaches, Test Cases are written before Coding.

In this tutorial, we will focus on Scrum as the Agile Testing Methodology.

tutorialspoint
SIMPLYEASYLEARNING

The other commonly used Agile Testing Methodologies are-

- **Test-Driven Development (TDD):** Test-Driven Development (TDD) is based on coding guided by tests.

- **Acceptance Test-Driven Development (ATDD):** Acceptance Test-Driven Development (ATDD) is based on communication between the customers, developers and testers and driven by pre-defined Acceptance Criteria and Acceptance Test Cases.

- **Behavior-Driven Development (BDD):** In Behavior-Driven Development (BDD) testing is based on the expected behavior of the software being developed.

tutorialspoint
SIMPLYEASYLEARNING

End of ebook preview

If you liked what you saw…

Buy it from our store @ **https://store.tutorialspoint.com**

tutorialspoint
SIMPLYEASYLEARNING