

Agile Testing

Agile Testing is a testing practice that follows the rules and principles of agile software development. Unlike the Waterfall method, Agile Testing can begin at the start of the project with continuous integration between development and testing. Agile Testing methodology is not sequential (in the sense it's executed only after coding phase) but continuous.

- Agile testing is an informal process that is specified as a dynamic type of testing.
- It is performed regularly throughout every iteration of the Software Development Lifecycle (SDLC).
- Customer satisfaction is the primary concern for agile test engineers at some stage in the agile testing process.

Agile Testing Principles

1. **Shortening feedback iteration:** In Agile Testing, the testing team gets to know the product development and its quality for each and every iteration. Thus continuous feedback minimizes the feedback response time and the fixing cost is also reduced.
2. **Testing is performed alongside** Agile testing is not a different phase. It is performed alongside the development phase. It ensures that the features implemented during that iteration are actually done. Testing is not kept pending for a later phase.
3. **Involvement of all members:** Agile testing involves each and every member of the development team and the testing team. It includes various developers and experts.
4. **Documentation is weightless:** In place of global test documentation, agile testers use reusable checklists to suggest tests and focus on the essence of the test rather than the incidental details. Lightweight documentation tools are used.
5. **Clean code:** The defects that are detected are fixed within the same iteration. This ensures clean code at any stage of development.
6. **Constant response:** Agile testing helps to deliver responses or feedback on an ongoing basis. Thus, the product can meet the business needs.
7. **Customer satisfaction:** In agile testing, customers are exposed to the product throughout the development process. Throughout the development process, the customer can modify the requirements, and update the requirements and the tests can also be changed as per the changed requirements.
8. **Test-driven:** In agile testing, the testing needs to be conducted alongside the development process to shorten the development time. But testing is implemented after the implementation or when the software is developed in the traditional process.

Agile Testing Methodologies:

1. **Extreme Programming (XP):** Extreme programming is a customer-oriented methodology that helps to deliver a good quality product that meets customer expectations and requirements.
2. **Test-Driven Development (TDD):** TDD is the software development process relying on creating unit test cases before developing the actual code of the software. It is an iterative approach that combines 3 operations, programming, creation of unit tests, and refactoring.
3. **Behavior Driven Development (BDD):** BDD is agile software testing that aims to document and develop the application around the user behavior a user expects to experience when interacting with the application. It encourages collaboration among the developer, quality experts, and customer representatives.

4. **Acceptance Test-Driven Development (ATDD):** ATDD is a collaborative process where customer representatives, developers, and testers come together to discuss the requirements, and potential pitfalls and thus reduce the chance of errors before coding begins.
5. **Crystal Methodologies:** This methodology focuses on people and their interactions when working on the project instead of processes and tools. The suitability of the crystal method depends on three dimensions, team size, criticality, and priority of the project.

Agile Testing Life Cycle

The agile testing life cycle has 5 different phases:

1. **Impact Assessment:** This is the first phase of the agile testing life cycle also known as the feedback phase where the inputs and responses are collected from the users and stakeholders. This phase supports the test engineers to set the objective for the next phase in the cycle.
2. **Agile Testing Planning:** In this phase, the developers, customers, test engineers, and stakeholders team up to plan the testing process schedules, regular meetings, and deliverables.
3. **Release Readiness:** This is the third phase in the agile testing lifecycle where the test engineers review the features which have been created entirely and test if the features are ready to go live or not and the features that need to be sent again to the previous development phase.
4. **Daily Scrums:** This phase involves the daily morning meetings to check on testing and determine the objectives for the day. The goals are set daily to enable test engineers to understand the status of testing.
5. **Test Agility Review:** This is the last phase of the agile testing lifecycle that includes weekly meetings with the stakeholders to evaluate and assess the progress against the goals.



Benefits of Agile Testing:

Below are some of the benefits of agile testing:

- **Saves time:** Implementing agile testing helps to make cost estimates more transparent and thus helps to save time and money.
- **Reduces documentation:** It requires less documentation to execute agile testing.
- **Enhances software productivity:** Agile testing helps to reduce errors, improve product quality, and enhance software productivity.
- **Higher efficiency:** In agile software testing the work is divided into small parts thus developer can focus more easily and complete one part first and then move on to the next part. This approach helps to identify minor inconsistencies and higher efficiency.
- **Improve product quality:** In agile testing, regular feedback is obtained from the user and other stakeholders, which helps to enhance the software product quality.

Limitations of Agile Testing:

Below are some of the limitations of agile software testing:

- **Project failure:** In agile testing, if one or more members leave the job then there are chances for the project failure.
- **Limited documentation:** In agile testing, there is no or less documentation which makes it difficult to predict the expected results as there are explicit conditions and requirements.
- **Introduce new bugs:** In agile software testing, bug fixes, modifications, and releases happen repeatedly which may sometimes result in the introduction of new bugs in the system.
- **Poor planning:** In agile testing, the team is not exactly aware of the end result from day one, so it becomes challenging to predict factors like cost, time, and resources required at the beginning of the project.
- **No finite end:** Agile testing requires minimal planning at the beginning so it becomes easy to get sidetracked while delivering the new product. There is no finite end and there is no clear vision of what the final product will look like.

Challenges During Agile Testing:

Below are some of the challenges that are faced during agile testing:

- **Changing requirements:** Sometimes during product development changes in the requirements or the specifications occur but when they occur near the end of the sprint, the changes are moved to the next sprint and thus become the overhead for developers and testers.
- **Inadequate test coverage:** In agile testing, testers sometimes miss critical test cases because of the continuously changing requirements and continuous integration. This problem can be solved by keeping track of test coverage by analyzing the agile test metrics.
- **Tester's availability:** Sometimes the testers don't have adequate skills to perform API and Integration testing, which results in missing important test cases. One solution to this problem is to provide training for the testers so that they can carry out essential tests effectively.
- **Less Documentation:** In agile testing, there is less or no documentation which makes the task of the QA team more tedious.
- **Performance Bottlenecks:** Sometimes developer builds products without understanding the end-user requirements and following only the specification requirements, resulting in performance issues in the product. Using load testing tools performance bottlenecks can be identified and fixed.
- **Early detection of defects:** In agile testing, defects are detected at the production stage or at the testing stage, which makes it very difficult to fix them.
- **Skippping essential tests:** In agile testing, sometimes agile testers due to time constraints and the complexity of the test cases put some of the non-functional tests on hold. This may cause some bugs later that may be difficult to fix.

Risks During Agile Testing:

- **Automated UI slow to execute:** Automated UI gives confidence in the testing but they are slow to execute and expensive to build.
- **Use a mix of testing types:** To achieve the expected quality of the product, a mixture of testing types and levels must be used.
- **Poor Automation test plan:** Sometimes automation tests plan is poorly organized and unplanned to save time which results in a test failure.

- **Lack of expertise:** Automated testing sometimes is not the only solution that should be used, it can sometimes lack the expertise to deliver effective solutions.
- **Unreliable tests:** Fixing failing tests and resolving issues of brittle tests should be the top priority to avoid false positives.

• 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

5.2 WEB BASED SYSTEM

Q. Explain Web based Systems :

- (i) Web based system, web technology evaluation
- (ii) Traditional software and web based software
- (iii) Challenges in testing for web based software
- (iv) Testing web based testing

- Web Testing, or website testing is checking your web application or website for potential bugs before its made live and is accessible to general public. Web Testing checks for functionality, usability, security, compatibility, performance of the web application or website.
- During this stage issues such as that of web application security, the functioning of the site, its access to handicapped as well as regular users and its ability to handle traffic is checked.

5.2.1 How to test Web Application ?

In Software Engineering, the following testing types/technique may be performed depending on your web testing requirements.

1. Functionality Testing of a Website

- **Functionality Testing of a Website** is a process that includes several testing parameters like user interface, APIs, database testing, security testing, client and server testing and basic website functionalities. Functional testing is very convenient and it allows users to perform both manual and automated testing. It is performed to test the functionalities of each feature on the website.

- Web based Testing Activities includes : Test all links in your webpages are working correctly and make sure there are no broken links. Links to be checked will include -
 - Outgoing links
 - Internal links
 - Anchor Links
 - Mail To Links
- Test Forms are working as expected. This will include-
 - Scripting checks on the form are working as expected. For example- if a user does not fill a mandatory field in a form an error message is shown.
 - Check default values are being populated
 - Once submitted, the data in the forms is submitted to a live database or is linked to a working email address
 - Forms are optimally formatted for better readability
- Test Cookies are working as expected. Cookies are small files used by websites to primarily remember active user sessions so you do not need to log in every time you visit a website. Cookie Testing will include
 - Testing cookies (sessions) are deleted either when cache is cleared or when they reach their expiry.
 - Delete cookies (sessions) and test that login credentials are asked for when you next visit the site.
- Test HTML and CSS to ensure that search engines can crawl your site easily. This will include
 - Checking for Syntax Errors
 - Readable Color Schemas
 - Standard Compliance. Ensure standards such W3C, OASIS, IETF, ISO, ECMA, or WS-I are followed.
- Test business workflow- This will include
 - Testing your end - to - end workflow/ business scenarios which takes the user through a series of webpages to complete.
 - Test negative scenarios as well, such that when a user executes an unexpected step, appropriate error message or help is shown in your web application.

• Tools that can be used : QTP, IBM Rational, Selenium

2. Usability testing

- Usability Testing has now become a vital part of any web based project. It can be carried out by testers like you or a small focus group similar to the target audience of the web application.
 - **Test the site Navigation** : Menus, buttons or Links to different pages on your site should be easily visible and consistent on all webpages
 - **Test the Content** : Content should be legible with no spelling or grammatical errors.



- Images if present should contain an "alt" text
- Tools that can be used: Chalkmark, Clicktale, Clixpy and Feedback Army

3. Interface Testing

- Three areas to be tested here are - Application, Web and Database Server
 - **Application** : Test requests are sent correctly to the Database and output at the client side is displayed correctly. Errors if any must be caught by the application and must be only shown to the administrator and not the end user.
 - **Web Server** : Test Web server is handling all application requests without any service denial.
 - **Database Server** : Make sure queries sent to the database give expected results.
- Test system response when connection between the three layers (Application, Web and Database) cannot be established and appropriate message is shown to the end user.
- Tools that can be used : AlertFox, Ranorex

4. Database Testing

- Database is one critical component of your web application and stress must be laid to test it thoroughly. Testing activities will include :
 - Test if any errors are shown while executing queries
 - Data Integrity is maintained while creating, updating or deleting data in database.
 - Check response time of queries and fine tune them if necessary.
 - Test data retrieved from your database is shown accurately in your web application
- Tools that can be used: QTP, Selenium

5. Compatibility testing

- Compatibility tests ensures that your web application displays correctly across different devices. This would include :
 - **Browser Compatibility Test** : Same website in different browsers will display differently. You need to test if your web application is being displayed correctly across browsers, JavaScript, AJAX and authentication is working fine. You may also check for Mobile Browser Compatibility.
- The rendering of web elements like buttons, text fields etc. changes with change in Operating System. Make sure your website works fine for various combination of Operating systems such as Windows, Linux, Mac and Browsers such as Firefox, Internet Explorer, Safari etc.
- Tools that can be used : NetMechanic

6. Performance Testing

- This will ensure your site works under all loads. Software Testing activities will include but not limited to :
 - Website application response times at different connection speeds
 - Load test your web application to determine its behavior under normal and peak loads
 - Stress test your web site to determine its break point when pushed to beyond normal loads at peak time.
 - Test if a crash occurs due to peak load, how does the site recover from such an event
 - Make sure optimization techniques like gzip compression, browser and server side cache enabled to reduce load times
- Tools that can be used : Loadrunner, JMeter

7. Security testing

- Security Testing is vital for e-commerce website that store sensitive customer information like credit cards. Testing Activities will include :
 - Test unauthorized access to secure pages should not be permitted
 - Restricted files should not be downloadable without appropriate access
 - Check sessions are automatically killed after prolonged user inactivity
 - On use of SSL certificates, website should re-direct to encrypted SSL pages.
- Tools that can be used: Babel Enterprise, BFBTester and CROSS

8. Crowd Testing

- You will select a large number of people (crowd) to execute tests which otherwise would have been executed a select group of people in the company. Crowdsourced testing is an interesting and upcoming concept and helps unravel many a unnoticed defects.
- Tools that can be used: People like you and me !!! And yes , loads of them!
- This concludes the tutorial. It includes almost all testing types applicable to your web application.
- As a Web-tester its important to note that web testing is quite an arduous process and you are bound to come across many obstacles. One of the major problems you will face is of course deadline pressure.
- Everything is always needed yesterday! The number of times the code will need changing is also taxing. Make sure you plan your work and know clearly what is expected of you. Its best define all the tasks involved in your web testing and then create a work chart for accurate estimates and planning.