

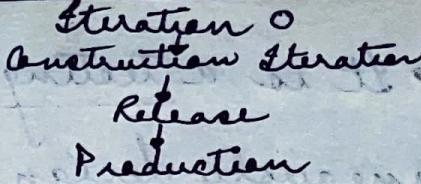
UNIT-4

~~Agile lifecycle & its impact on testing ; Testing user stories, acceptance tests & scenarios, Plan & Manage Agile Testing, Exploratory Testing, Risk based Testing, Regression Tests, Test Automation ; Tools to support Agile Tester~~

- # Agile Testing → It is a type of testing that follows principles of agile software development to test app.
- All members of project including experts & testers are involved.
 - It is not a separate phase & is carried out with development phases i.e. requirements, design, coding, test case generation.
 - Testers help in building s/w acc. to customer requirements with better design.
 - Testers work as a single team for achieving "quality".
 - This methodology is called "delivery driven approach" because it provides better prediction on workable products in less time.

Principles of Agile Testing →

- 1) Shortening feedback iteration → Continuous feedback minimizes cost of feedback response time.
- 2) Testing is performed alongside → It is carried out simultaneously with development.
- 3) Involvement of all members → It includes each one of dev & testing team & experts.
- 4) Lightless documentation → In place of global test documentation, agile testers use reusable checklists to suggest tests.
- 5) Clean Code → defects detected are fixed within same iteration.



Agile Testing Lifecycle →

- (i) Iteration 0 → 1st stage of testing process & initial setup is done. Testing environment is set.
- (ii) Construction Iteration → 2nd phase of a major phase as most of work is performed at this stage. It is a set of iterations to build an increment of solution.
- (iii) Release → It includes full system testing & acceptance testing. The product is tested more relentlessly while it is in "construction iterations". Testers work on defect stories.
- (iv) Production → Last phase where "product id" is finalized after removal of all defects & issues.

Agile Testing Activities →

- i) Participation in iteration planning ii) Estimating tasks
- iii) Writing test cases iv) Unit Testing v) Integration Testing
- vi) Feature testing vii) Defect fixing viii) Acceptance Testing
- ix) Acceptance Testing x) Status Reporting xi) Defect Tracking

Agile Test Plan → It includes types of tests done in that iteration like test data requirements, infrastructure, test environments and test results.

→ If test plan is written & updated for every release.

→ Test plan in agile includes :

- (i) Testing Scope ii) New functionalities tested iii) Type of Testing based on feature complexity iv) Load & performance testing.
- v) Infrastructure consideration vi) Mitigation
- vii) Resourcing viii) Deliverables

i) Exploratory Testing → It is a testing approach in which testers simultaneously design and execute tests without pre-defined test cases, leveraging their expertise to uncover defects & gain insights.

→ It identifies functional & technical faults.

→ It aims to optimize s/w.

→ Often performed as black box testing

→ It is an unscripted testing technique

When to use it ???

o) Need to learn quickly about application → when there is a need to review product from user perspective.

o) Where early iteration is required, this tends to be helpful.

o) It ensures that tester doesn't miss edge case which can cause failures.

o) can be used to document test cases so that it can be used in later sprints.

Process → (i) Learn → 1st phase in which tester learns about faults. He/she uses his/her knowledge, skill to observe & find what kind of problem s/w is suffering from.

(ii) Test Case Creation → When fault is identified, then tester creates test cases acc. to defects.

(iii) Test Case Execution → Tester executes test cases. This includes computational & operational task performed by s/w to get desired op.

(iv) Analysis → Result is analyzed & observed whether s/w is working properly or not. If defects found, then they are fixed & above 3 steps are repeated.

- # Advantages → (i) less documentation & preparation
- (ii) it helps to find critical defects
- (iii) it improves productivity
- (iv) encourages creativity
- (v) helps uncover bugs that are ignored by other techniques

Disadvantages →

- 1) Testing cannot be reviewed in advance.
 - 2) Testing is dependent on tester's knowledge & skill.
 - 3) Keeping track of tests is difficult due to ad-hoc nature.
 - 4) Determining best test case is difficult.
- 2) Regression Testing → It is a type of testing to confirm that a code change has not affected existing features.
- It is full or partial selection of test cases already executed which are re-executed to ensure existing functions work fine.
- # Need → (i) When there is requirement to change the code & we need to test whether modified code affects other parts or not.
- (ii) When a new feature is added to an application.
- In order to do Regression Testing we need to first debug code to identify bugs.
- Once bugs are identified, changes are made to fix it, then regression testing is done by selecting relevant test cases that covers both modified & affected parts of code.

→ Regression Testing can be carried out using following methods:

- a) Retest all → All tests in existing test bucket / ~~suite~~ should be re-executed. This is expensive as it requires huge time and efforts.
- b) Regression Test Selection → It is a technique in which some selected test cases from test suite are executed whether modified code affects s/w or not.
- c) Prioritization of Test Cases → Prioritize test case depending on business impact, frequently used functionalities. Selection of test cases based on priority will reduce regression test suite.

Selecting Test Case for regression testing

- i) Test cases having frequent defects.
- ii) Functionalities which are visible to users.
- iii) Test cases which verify core features of product.
- iv) Test cases of functionalities which has gone under more changes.
- v) All integration test cases.
- vi) All complex test cases.
- vii) Boundary value test cases.
- viii) Sample of successful test cases.
- ix) Sample of failure test cases.

REGRESSION TESTING TOOLS →

- i) AVO ASSURE → i) auto generates test cases with 100% no code approach; ii) tests across web, desktop, mobile, ERP app.
- ii) executes test cases in single VM independently.
- iii) Integrate with JIRA, Jenkins, Salesforce, TFS etc.
- iv) Define test plans & design test cases through "Mindmap features".

2) EGGPLANT → It is AI driven tool that streamlines regression testing through prioritization of test cases.

(i) tests more important areas

(ii) reduces time

(iii) discovering bugs missed by exploratory testing

(iv) reduce the burden of test maintenance

(v) reuse testing models & scripts

3) SELENIUM → Open Source Tool for automating web applications. Selenium can be used for browser based regression testing.

• QTP → HP's Quick Test Professional is automated s/w designed to automate functional & regression test cases. It uses VBScript for automation. It is a data-driven, keyword based tool.

• RFT → IBM's Rational Function Tester is a Java tool used to automate test cases of s/w applications. This is used for automating regression test cases & it also integrates with Rational Test Manager.

Configuration Management during Regression Testing becomes imperative where a code is continuously modified. To ensure effective RT, observe following:

(i) Code being regression tested should be under CM tool.

(ii) ~~No code~~ Regression Test code must be immune to changes.

(iii) The dB must be isolated as no changes are required.

Advantages →

- 1) ensures no new bugs have been introduced after adding new functionalities.
- 2) can be easily automated by automation tools.
- 3) helps maintaining quality of source code.

Disadvantages →

- 1) Time Consuming
- 2) Resource consuming
- 3) it is required even after small changes in code.

3) Risk Based Testing → It is a testing type which is based on probability of risk. It involves assessing risk based on criticality of business, frequency of use.

- RBT prioritizes testing of features & functions which are likely to have defects & more impactful.
- Risk is an occurrence of uncertain event with +ve / -ve effect.
- +ve Risks → referred to as opportunities and help in business sustainability. e.g. investing in new project, developing new products.
- -ve Risks → referred to as threats & recommendation to minimize them.

Q) Where to use RBT ?

- i) Projects having time, resource & budget constraints.
- ii) Project where risk based analysis can be used to detect vulnerabilities.
- iii) Security in Testing Cloud Computing Environment.
- iv) Incremental & Iterative Models.
- v) New project with lack of domain knowledge, lack of experience with technology.

Risk Management Process →

- i) Risk Identification → can be done through risk workshops / checklists / contacting domain & subject matter experts.
- ii) Risk Register → is a spreadsheet which has list of identified risks & root causes. It is used to monitor & track the risks.
- iii) Risk Breakdown Structure → helps in identifying risk prone areas & helps in effective evaluation & risk monitoring
- iv) Risk Analysis → Analyzing identified risks & filter it based on significance.
→ Risk Matrix is used for qualitative Risk Analysis.
→ used to determine probability & impact of risk.
- v) Risk Response Planning → Based on analysis, we decide if risk requires a response.
→ Some risks require response in project planning whereas some in monitoring etc.
- vi) Risk Mitigation → Risk response method used to lessen impact of threats. Can be done by eliminating risks.
- vii) Risk Contingency → possibility of an uncertain even but the impact is unknown.
→ Contingency plan is an action plan for worst case scenarios.
- viii) Risk Monitoring ^{Control} → Used to track risks, monitor, identify new risk update risk register, execute risk response plan & monitor risk triggers. Done by risk assessments & meetings &

Risk Based Testing Approach

- 1) Risk Identification → Consult business & technical staff & prepare a draft register of risks.
- 2) Risk Analysis → Discuss risks; Calculate exposure.
- 3) Risk Response → Formulate test objectives & select test technique; Design test effectiveness score & nominate responsibilities.
- 4) Test Scoping → agree scope of ~~test~~ risk to be addressed; agree responsibilities & budgets.
- 5) Test Process definition → Draft test process from worksheet and complete test stage definitions.

- e.g:-
- 1) Analyze requirements
 - 2) Documents are reviewed to eliminate errors.
 - 3) Requirement's sign off
 - 4) Assess the risk
 - 5) Identify probability of failure & high risk areas
 - 6) Use risk register to list set of risks
 - 7) Risk prioritizing needs to be done.
 - 8) Prioritize the requirements.
 - 9) Risk based test process is defined.
 - 10) Imp.
- (1) In this process, risks are identified & categorized, a draft risk register is prepared. Risk sorting is done! (2) Risk response involves formulating test objectives from risks & selecting techniques. (3) Determines dependencies, requirement, cost, time to calculate test effectiveness score. (4) Test scoping is a review activity that requires participation of everyone. It is imp. to adhere to scope of risk. (5) After scope is final; test objectives, dependencies has to be compiled in standard format.

Risk Based Testing Approach to System Test →

- 1) Technical System Test → refined to as environment & integration test. Env. test includes testing in development, testing & production environment.
- 2) Functional System Test → Testing of all functionalities, features, programs. Purpose is to evaluate if the system meets requirements specified.
- 3) Non-Functional System Test → Testing performance load test, stress test, security test, backup & recovery procedures & documentation.

Advantages → (i) improved productivity
 (ii) cost reduction (iii) improved performance
 (iv) continuous risk monitoring
 (v) improved customer satisfaction
 (vi) improved quality of sw.

Test Automation → It is process of automatically reviewing & validating a sw product to ensure it meets quality standards & user experience.

Test Automation → refers to practice of using sw tools & scripts to automate execution of test cases ensuring quality & reliability of sw.



Components of Test Automation →

- 1) Test Frameworks → provides a structured approach for executing test cases. e.g. - Selenium for web & Appium for mobile.
- 2) Test Script → written using Java, Python to automate test scenarios & verify s/w functionalities.
- 3) Test Data → To stimulate real-world scenarios
- 4) Reporting → Automated test execution generates detailed reports on test results (pass/fail status, error logs).

Advantages →

- 1) Reduces time & effort for testing.
- 2) Improves efficiency.
- 3) Provides accurate results.
- 4) Enables testing for a broader range of scenarios.
- 5) Automated test scripts can be reused.
- 6) Provides high scalability.

Challenges →

- 1) Setting up the infrastructure in initial phase requires time & expertise.
- 2) Automated test needs to be regularly maintained & updated.
- 3) AT may produce false results due to test script issues.
- 4) AT requires proficiency in python script language, test frameworks.

Open Source Testing Tools for Agile Testers →

Priorix™

- 1) Selenium web driver → Most widely used tool for browser test automation.
 - Keyword driven framework can be easily built
 - Stimulates how an actual user ^{can} interact with application using automated test cases.
 - One of main advantage is that it supports various languages such as Java, Python, Ruby for writing automated tests.
 - Often used in Agile because there is a heavy emphasis on automated testing & whole team can contribute to automated tests.
- 2) JMeter → It is open source performing testing tool written in Java.
 - used to "load-test" websites (HTTP, HTTPS) as well as Web Services (SOAP) & databases
 - can be extended by plugins to support more functionalities.
 - There is also a Jenkins plug-in to run performance tests as part of delivery pipeline
- 3) SoapUI → used for functional testing of Web Services.
 - easy to use & create test cases in few minutes.
 - can execute tests & analyse reports all inside SoapUI GUI.
 - ability to create "mock" web services which are handy when you create tests for web service which is not developed yet.

- 4) VirtualBox → These days, most applications need to be tested against multiple browsers & OS.
→ Instead of having physical servers with diff. OS & browsers, VirtualBox provides an easy solution to create virtual machines & diff. configurations.
→ VirtualBox can be run on any OS to create virtual machines & you can have multiple VMs with diff. OS on same box.
- 5) XMind → most popular of feature rich mind mapping tool.
→ used to create stunning mind maps to highlight the features of a website & or create user journey flow through application.
→ "Mind Maps" have become more popular because they provide nice visuals & can be of great help while designing test cases of a s/w application.

- # Testing User Stories — acceptance tests & scenarios
→ User story is an essential aspect of agile development ensuring s/w meets requirements of end users.
→ Acceptance test plays vital role in validating user stories & deliver value to stakeholders. They are written on user story acceptance criteria & focus on verifying functionalities. Acceptance scenarios outline specific scenario that needs to be tested to validate user story. It should be clear, unambiguous & executable.

How acceptance test / scenario is written ?

Priorix™

Scenarios are typically structured as below :-

- 1) Title / Description
- 2) Preconditions
- 3) Sequence of steps that user performs to accomplish goal. Each step should be clear, specific.
- 4) Expected Outcome → expected result after completing step.
- 5) Alternate Paths → s/w can handle various scenarios & user inputs
- 6) Post conditions → ^{users to} db updates / change to UI

e.g. - Online Shopping → Title : User adds item to cart

1. User selects desired product from product catalog

2. user clicks on "Add to cart" button

3. User views shopping cart to confirm added item

↳ Expected Outcome → selected item is added to cart & cart reflects the amount .

↳ Alternate Paths → (i) if user is not logged in , they are prompted to login before adding item to cart .

(ii) if selected product is 'out of stock' user receives a notification & can't add it to cart .

Testing of User Story →

1) Review User story & its acceptance criteria to understand functionality & expected outcomes .

2) Break down user story into scenarios for diff function-

3) Write acceptance tests that verify behaviour of system .

4) Execute acceptance test for each scenario to validate the behaviour .

Havrix

Varilrix™
VARICELLA VACCINE, LIVE ATTENUATED

Tritanrix HB
DTaP + Haemophilus influenzae type b (PRP-T) vaccine

Hiberix
Haemophilus influenzae type b (PRP-T) vaccine

FluaRIX

- 5) Verify acceptance criteria specified in user story
- 6) Test the "edge cases & boundary conditions" to ensure robustness & reliability of system functionality
- 7) Provide feedback on tested user story.
- 8) Perform regression testing to ensure changes made to implement user story does not adversely affect the existing functionalities.
- 9) Once user story has been thoroughly tested, sign off its completion & readiness for release. Ensure all stakeholders are satisfied.