**Day 7: Test Execution & Reporting**

When to start testing?

* The simple answer – “as early as possible”
* The entry criteria for test execution is:
  + Approved requirements
  + Approved test plan
  + Test cases ready and reviewed
  + Test infrastructure and test environment ready (smoke tested)
  + Test data is available
  + Availability of code to test
* Difficult question – Testing is never ending process.
* The exit criteria for test execution is:
  + Test execution completed
  + Requirement coverage done
  + No critical/blocker defects open
  + Defect rate below accepted rate
  + Deadline is reached

Executing Test Cases

* The code is deployed and smoke test passed
* Regression testing is main time-taker
* Execute test cases – Manual/Automation/both – based on test planning
* Test cases will either Pass, Fail, Incomplete, or Blocked etc
* Failed and blocked test cases should have a defect ID
* Defects should be tracked
* Re-testing should be tracked
* Re-testing should be done once defects are fixed
* Continuous reporting to stakeholders
* Risk discussions and mitigation should be done

Non-Functional testing

* Non-functional testing is important to ensure application meets customer expectations
* Objective of NFT is to verify non-functional aspects, like performance, security, usability
* Common NFT techniques are performance testing, security testing, usability testing, compatibility testing, compliance testing accessibility testing etc.
* NFT is done after functional testing, but is should be planned from beginning
* NFT also includes same process as functional – Planning, Preparation, execution, Recor, Analyze & Improve

Defects and how to log them

* Reporting detailed defect is necessary skill for testers
* Most companies use a defect tracking tool, like JIRA, Bugzilla, Mantis etc
* Sample information present in a reported defect is – Defect id, Title, Build/Version, severity, status, steps, expected & actual results, assignee, reporter, attachments (screenshot etc)
* Best practices when reporting a defect:
  + Provide clear information, with precise steps to reproduce
  + Put in as much information as helpful, but no assumptions
  + Assign severity and priority properly
  + Provide environment, test data, screenshots/videos
  + Do not put multiple issues in 1 defect. Do not log duplicate defects

Defects Life Cycle

REJECTED

NEW

DUPLICATE

ASSIGNED

DEFERRED

OPEN

WAD

REOPENED

READY TO TEST

CLOSED

IN PROGRESS

When the defect is fixed, or, not fixed

* If the defect is fixed, close it with comments and screenshot of proof
* Double check the release version, so you know when it’s going to go live
* Analyze regression – do any additional scenarios need to be tested?
* If the defect is not fixed, check if the fix build is deployed
* Re-open and assign back to developer with comments, build info, screenshot
* Also sure there was no confusion in reproducing the defect
* Make sure there was no confusion in reproducing the defect
* Talk to the developer (if possible) and give a walk-through of the failing scenario

Cost of defect:

* It is very important to find defects as early as possible
* The earlier a defect is found, the quicker, easier, cheaper to fix it
* Requirements phase – cost = rewriting requirement document
* Coding phase – cost = development hours, unit testing
* Testing phase – cost = analysis, development hours, unit testing, re-testing, regression, BA hours, customer hours
* UAT phase – cost = all of above and re-deployment. May also lead to loss of business, customer dissatisfaction, legal implication etc.

Execution status & Defect status reporting:

* During execution, a report of entire testing status should be sent out
* This should go out to Development team, Business team, System admin team, and any other stakeholders
* The report should contain test execution status, defects and their status, any show-stoppers, any risks etc.
* A good test execution report should be:
  + Complete, and accurate
  + Readable with key facts highlighted for quick summary
  + Periodic – example, sent out every day at end of testing