

MANUAL TESTING CHEAT SHEET

Priority: The Importance of the defect to be resolved with respect to the customer.	Severity: The seriousness of the defect in terms of functionality.
Retesting: Retesting is the act of repeating a test to verify that a found defect has been correctly fixed	Regression Testing: The Re-Execution of selected tests on modified build to estimate Completeness and correctness of modification is called Regression Testing.
Black Box Testing 1)Black Box Testing is a software testing method in which the internal structure of the item being tested is NOT known to the tester. 2)Programming Knowledge not required. 3)Mainly applicable to higher levels of testing: 1)Acceptance Testing 2)System Testing	White Box Testing 1)White Box Testing is a software testing method in which the internal structure of the item being tested is known to the tester. 2)Programming Knowledge Required. 3)Mainly applicable to lower levels of testing: 1)Unit Testing 2)Integration Testing
QA 1)It is process oriented 2)It involve in entire process of software development. 3)It is prevention oriented.	QC 1)It is product oriented. 2)Work to examine the quality of product. 3)It is detection oriented.

Unit Testing: The testing of individual software components

Integration Testing: Testing performed to expose defects in the interfaces and in the interactions between integrated components or systems.

System Testing: The process of testing an integrated system to verify that it meets specified requirements.

Monkey Testing: Due to lack of time, the testing team concentrates on some of the main activities in the software build for testing. This style of testing is known as Monkey testing or Chimpanzee testing or Gorilla testing.

Acceptance Testing: Formal testing with respect to user needs, requirements, and business processes conducted to determine whether or not a system satisfies the acceptance criteria and to enable the user, customers or other authorized entity to determine whether or not to accept the system.

Testing is enough when the 90% of requirements are covered. Maximum defects are rectified except (some) low level defects are not covered, customer satisfy that project and time is less, then we are closing the testing.

- 1) **Error:** programmatically mistake leads to error.
- 2) **Bug:** Deviation from the expected result.
- 3) **Defect:** Problem in algorithm leads to failure.
- 4) **Failure:** Result of any of the above.

Software Testing Life Cycle

Test Initiation
Test Planning
Test Design
Test Execution
Test Closure
UAT
Signoff

SDLC: Software Development Life Cycle defines the steps/stages/phases in the building of software.

Information gathering -> Analysis -> Design -> Coding -> Testing ->Maintenance

Waterfall Model is suitable when the customer requirements are clear and complete

Prototype Model is suitable when the customer requirements are unclear and confusing

RAD Model is suitable when the customer requirements are clear but not complete, because client is giving requirements in installment basis.

Spiral Model is suitable when the customer requirements are clear and complete but enhancing in future

Advantages	Disadvantages
Waterfall Model: 1) It is a linear model, and follows a proper sequence and order 2) Documentation is produced at every stage, it is easy to track down mistakes, deficiencies and any other problems that may arise 3) The cost of resources at each stage get minimized due to the linear sequencing as well	1) If the customer is ambiguous about his needs, the design process can go horribly wrong. This factor is further highlighted by the fact that if some mistake is made in a certain stage and is not detected or tracked; all the subsequent steps will go wrong. Therefore the need for testing is very intense
Spiral Model 1) High amount of risk analysis 2) Good for large and mission-critical projects. 3) Software is produced early in the software life cycle.	1) Can be a costly model to use. 2) Risk analysis requires highly specific expertise. 3) Project's success is highly dependent on the risk analysis phase.
RAD Model: 1) It increases speed of developing software. 2) Re-usability of components helps to speed up development. 3) It increases the quality. 4) Some systems also deliver advantages of interoperability, extensibility, and portability.	1) Unknown cost of product. 2) Difficult to commit the time required for success of the RAD process. 3) Short iteration may not add enough functionality, leading to significant delays in final iterations. 4) Success depends on the extremely high technical skills of the developers.
Prototype Model 1) Helps in reducing risk associated with the project. 2) There is a great involvement of user in s/w development.	5) This model is time consuming and expensive. Because if the user is not satisfied by the developed prototype then a new prototype is developed. This process goes on until a perfect prototype is developed.
V-Model 1) Simple and easy to use. 2) Each phase has specific deliverables. 3) Higher chance of success over the waterfall model due to the development of test plans early on during the life cycle. 4) Works well for small projects where requirements are easily understood.	1) Very rigid, like the waterfall model. 2) Little flexibility and adjusting scope is difficult and expensive. 3) Software is developed during the implementation phase, so no early prototypes of the software are produced. 4) Model doesn't provide a clear path for problems found during testing.

Bug Life Cycle:

Defect defect
Report defect
Resolve defect
Re-test defect
Close defect

Bug Status Cycle

New
Assigned
Open/Rejected/Differed
Fixed
Closed
Reopen

Testing: software testing is an activity to check whether the actual results match the expected results and to ensure that the software system is defect free

Quality: The degree to which a component, system or process meets specified requirements and/or user/customer needs and expectations

DRE: Defect Removal Efficiency

$$DRE = A / A + B$$

A -> Bugs found during testing at development

B -> Bugs found at the client

Defect Density: Defect Density is the number of confirmed defects detected in software during a defined period of development divided by the size of the software (Size -> Functional points and LOC)

Defect age: Defect Age is the difference in time between detected on and resolved on.

RTM: Traceability matrix is nothing but mapping between Requirements and test cases to ensure that all requirements covered.

Test Engineer Responsibilities:

- 1)Review of software requirements
- 2)Preparation of test cases
- 3)Execution of tests
- 4)Reporting of defects
- 5)Preparation of test reports

Integration testing Approaches:

1)**Top-Down Approach:** Top Down is an approach to Integration Testing where top level units are tested first and lower level units are tested step by step after that. This approach is taken when top down development approach is followed. Test Stubs are needed to simulate lower level units which may not be available during the initial phases.

2)**Bottom-Up Approach:** Bottom Up is an approach to Integration Testing where bottom level units are tested first and upper level units step by step after that. This approach is taken when bottom up development approach is followed. Test Drivers are needed to simulate higher level units which may not be available during the initial phases.

3)**Sandwich Approach:** Sandwich/Hybrid is an approach to Integration Testing which is a combination of Top Down and Bottom Up approaches



FOLLOW ME ON :



NOOR E ARAFIN RAFI



TESTING WITH RAFI