Unit-II
Test Planning and Management

* Review of fundamentals of Software Testing

(Ref page No. 64-73)

Defination of Testing

- Testing is defined as execution of a work product with intent to find a defect.
- The primary role of software testing is not to demonstrate the correctness of software product but to expose hidden defects so that they can be fixed.
- Testing is done to protect the common user from any failure of system during usage.
- Testing is necessary due to following reasons

(i) Understanding of customer requirements may differ from person to person. One must challenge the understanding at each stage of development & there must be some analysis of customer expectations. Everything is considered as ok unless there is an independent view of a system.

(ii) Gaps between requirements design and coding may not be traceable unless testing is performed in relation to requirements.

- Software Testing is the process of analysing a software item to detect the difference between existing and required conditions and to evaluate the feature of the software item.

- Decting Specification-related errors and deviations of working applications with respect to the specifications. Requirement mismatches and misinterpretation must be dected by testing.

Page No.:

(1) manager's view of software Testing :-

The senior management from development organisation and customer organisation have the following views about testing the software product being developed.

i) The product must be safe & reliable during use & must work under normal & adverse condn when it is actually used by the intended users.

ii) The product must exactly meet the user's requirements. These may include implied & defined requirements.

(2) Tester's view of software Testing :-

i) The purpose of testing is to discover defects in the product & process related to development & testing. This may be used to improve the product & processes used to make it.

ii) Testing is a process of trying to discover every conceivable fault or weakness in a work product so that they will corrected eventually.

(3) Customer's view of software Testing :-

Customer is the person or entity who will be paying for it. Testing considered as the representatives of the customer in system develop-ment.

i) Testing must be able to find all possible defects in the software alongwith related docume so that these defects can be removed.

ii) Testing must ensure that any legal or regulatory requirements are complied during development.

Page No.:

Common Testing terms :-

Debugging :-

It is part of development activity that identifies, analyzes & removes defects. Debugging is performed by developers on their piece of code.

Testing :-

It is the activity of identifying defects & is performed by Testers.

Review :-

It can be performed on deliverables like documents, code, test plan & test cases. Testing can be done when the executable code is ready.

- Role of Software Testing :-
  - Rigorous testing is necessary during software development & maintenance to
    - Identify defects.
    - Reduce failures in the operational environment
    - Increase quality of the operational system.
    - meet contractual or legal requirements.
    - meet Industry specific Standards which may specify the type of techniques that must be used or percentage of the software code that must be executed.

- Objectives of Software Testing :-
  - finding defects which prevent the probability of their occurence in production.

- Gaining confidence in the quality of the software application.
- Providing information helps Go or No Go decision making while moving to the next phase.
- Find a scenario where the product does not do what is supposed to do. This is derivation from requirement specification.

* Testing during development life cycle :-

① Requirement Testing :-

- Requirement testing involves mock running of future application using the requirement statements to ensure that requirements meet their acceptance criteria.
- This type of testing is used to evaluate whether all requirements are covered in requirement statement or not.
- This type of testing is similar to building use cases from the requirement statement.
- Requirement testing differs from the verification of requirements.

Characteristics of requirements verification include

- completeness of requirement statement as per organisation standards & formats.
- clarity about what is expected by the users at each step of working while using an application.
- Traceability of requirements further down the development life cycle must be ensured.

② Design Testing :-

- Design testing involves testing of high-level design (system architecture) & low level design (detail)
- High level design testing covers mock running of future appln with prerequisites as if it is being executed by the targeted user in production environment.
- This testing is similar to developing flow diagrams from the design, where flow of information is tracked from start to finish.

* Design verification ensures that designs meet their exit criteria-

- completeness of design, in terms of covering all possible outcomes of processing & handling of various control as defined by requirements.
- Testability of a design which talks about software structure & structural testing.
- Traceability with requirements
- Design must cover all requirements

③ Code Testing :-

- code files, Tables, Stored procedures etc. are written by developers as per guidelines, standard and detail design specifications.

* Code review is done to ensure that code files written are,

- Readable & maintainable in future. There are adequate comments available.
- Testable in unit testing.
- Testable in integration & system testing.
- Optimised to ensure better working of softwar Reusability creates a lighter system.

④ Test Scenario & Test case Testing :-

- Test scenarios are written by testers to address testing needs of a software appln.
- Test cases are derived from test scenarios which are related to requirements & design.
* Test scenarios can be functional & structural depending upon the type of requirement & design are addressing
- Test scenario should be clear & complete, representing end to end relationship of what is going to happen & also the possible outcome of such processing.
- Test scenarios should cover all requirements
- Test cases should cover all scenarios completely.

* Requirement Traceability matrix :- (Ref Page No. 75)

- Some quality management models & standards prescribe complete traceability of software appln from requirements through design & code files upto test scenario, test data, test cases & test results.
- Requirement traceability matrix is one way of doing the complete mapping for the software.
- One can expect a blueprint of an entire appln using requirement traceability

Require-ments	High level design	Low level design	code files/ stored procedures/ TBIs	Test scenario	Test cases	Test Results

fig: Table Requirement traceability matrix

* Advantages of traceability Matrix
  - Entire software development can be tracked completely through requirement traceability matrix.
  - Any test case failure can be tracked through requirements, design, coding.
  - The application becomes maintainable as one has complete relationship from requirement till test result available.

* Problems with Requirement Traceability Matrix
  - Number of requirements in huge. It is difficult to create requirement traceability matrix manually for using some tools one needs to invest money.
  - There may be one-to-many, many-to-one, many-to-many relationship betn various elements of traceability matrix, when we are trying to connect columns & rows of traceability matrix & maintaining these relationships need huge efforts.
  - Requirements change frequently and one needs to update the requirement traceability matrix whenever there is a change.
  - Incremental & iterative developments are the major challenges for maintaining traceability.
  - A customer may not find value in it & may not pay for it.
  - Types of traceability
    i) Horizontal Traceability
    ii) Bidirectional Traceability
    iii) vertical Traceability
    iv) Risk Traceability

* Essentials of Software Testing :- (Ref Page No. 77)

- Software testing is disciplined approach.
- It executes software work products & finds defects in it.
- The intention of software testing is to find all possible defects and failures, so that eventually these are eliminated and good product is given to the customer.
- Software testing is also viewed as an exercise of doing a SWOT analysis (Strength, weakness, opportunity, threats) of software product where we can build the software on the basis of strengths of the process of development & testing & overcome weakness in the processes to the maximum extent possible.

① Strength :-

- Some areas of software are very strong and no (very less) defects are found during testing of such areas.
- The areas may be in terms of some modules, screens, algorithms or processes like requirement definition, design, coding & testing.
- This represents strong processes present in the areas supporting development of a good product.

② Weakness :-

- The areas of software where requirement compliance is on the verge of failure may represent weak areas.
- The processes in this areas represent problems.
- An organisation needs to analyse such process and define the root cause of problems to these possible failures.

③ Opportunity :-

- Some areas of the software which satisfy requirements as defined by the customer or implied requirements but with enough space available for improving in further.
- This improvement can lead to customer delight.
- These improvements represent ability of the developing organisation to help the customer & give competitive advantage.

④ Threats :-

- Threats are the problems or defects with the software which result into failures.
- They represent the problems associated with some processes in the organisation such as requirement clarity, knowledge base & expertise.

* Workbench :- (Ref Page No. 79)

- Workbench is a term derived from the engg set-up of mass production.
- Every workbench has a distinct identity as it takes part in the entire development life cycle.
- It receives something as an input from previous workbench & gives output to the next workbench.
- A workbench comprises some procedures defined for doing a work and some procedures defines to check the outcome of the work done.
- The work may be anything during software development life cycle such as collecting the requirements making design, coding, testing.

Page No.:

(i) Tester's Workbench :-

- Tester's workbench is made of testing process, standards, guidelines and tools used for conducting tests & checking whether the test process applied are effective or not.
- for every workbench there should be a defn of entry criteria, process of doing/checking work & exit criteria.
- for testers there must be a defn of all things that enter the testers workbench.
- These may be defined in a test plan.

Examples of Tester's workbench :-

- considering a typical system life cycle for a product, project, the diffn work benches for a tester may be defined as follows.
  - workbench for creating test strategy.
  - workbench for creating a test plan.
  - workbench for writing test scenario.
  - workbench for writing test cases.
  - workbench for test execution
  - workbench for defect management.
  - workbench for testing.

Some of the typical workbench for system testing execution

(i) Inputs to tester's workbench
(ii) Do process
(iii) check process
(iv) Output
(v) Standards & Tools
(vi) Rework