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

- Check process must work ensure that results meet specifications & standards & also the test process is followed correctly. If no problem is found in the test process one can release the output in terms of test results. If problems are found in test process. Following fig. shows schematic diagram of a workbench.

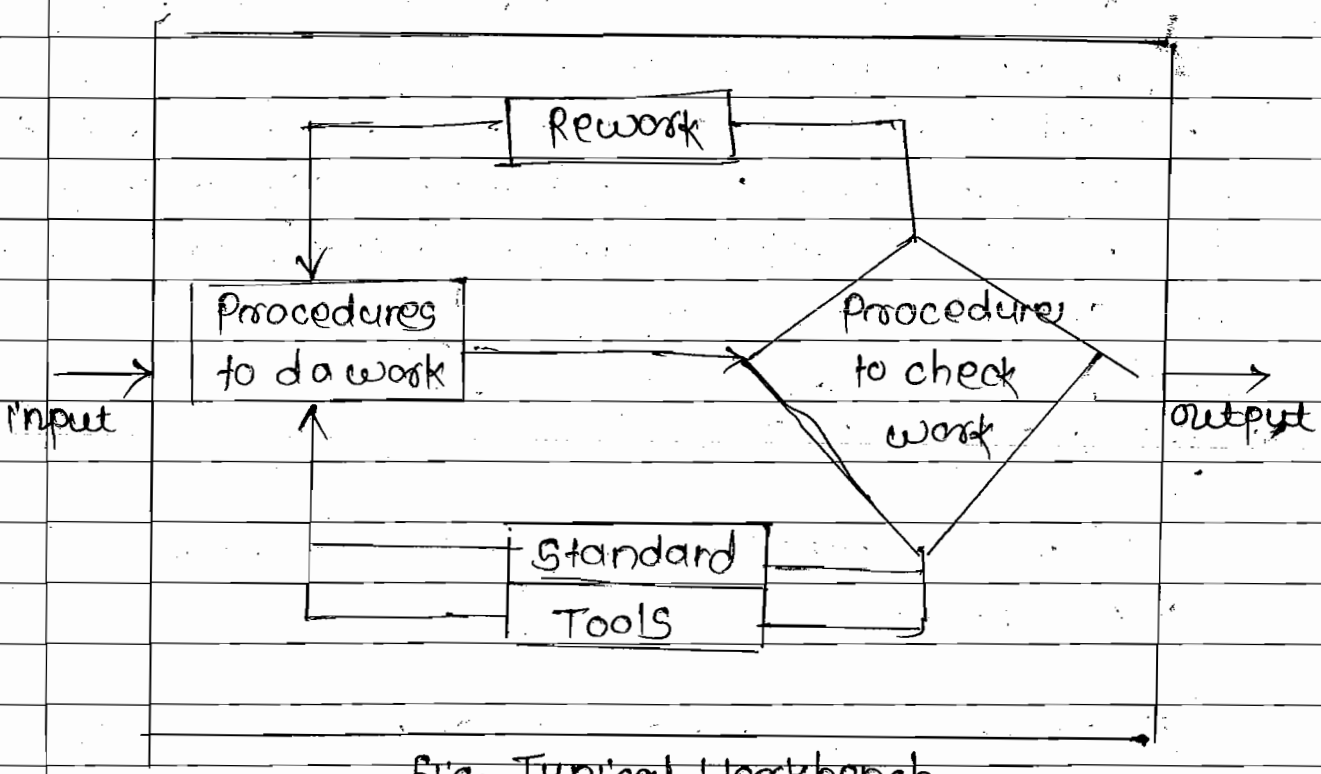

Fig. Typical Workbench

* Important features of Testing Process :-

(Ref Page No. 80)

Testing is characterised by some special features.

① Testing is Destructive process. But it is constructive Destruction.
② Testing Needs a sadistic Approach with a consideration that there is a defect.

(3) If the test does not detect a defect present in the system, it is an unsuccessful test

(4) A test that detects a defect is a valuable investment for development & customer it helps in improving a product.

(5) It is risky to develop software and not to Test it

(6) With high pressure to deliver software as quickly as possible, Test process must provide maximum value in shortest timeframe.

(7) Highest payback comes from detecting defect early in software development life cycle & preventing defect leakage/ defect migration from one phase to another.

* Misconceptions :- (Ref. page no. 7)

Misconceptions about software testing is listed below:-

(1) Testers can test quality of product at the end of development process

- This is a typical approach where system testing or acceptance testing is considered as qualification testing for software.

- Few test cases out of infinite set of possibilities are used for certifying whether the software appln. works or not.

- The customer may be dissatisfied as the appln does not perform well as per his expectations

(ii) Defects found in Testing are blamed on Developers
- Another common misconception regarding defects found in testing in blaming developers for defects.
- Though two-third of defects are due to wrong requirements, yet developers are mostly blamed for defects in software development.
- Developers are responsible for converting into code by using standards or guidelines available.

(iii) Defects found by customer are blamed on Testers
- Tester perform testing by executing few test cases & try to cover some part of software program to check whether the program performs as intended or not.
- If no defect is found during testing, it does not indicate that the software program is defect free
- One must do a root cause analysis of the defects found & try to learn from the experience to ensure that a better product is produced.

* Principles of Software Testing :- (Ref Page No: 83)

- Testing needs to be performed according to processes defined for it. It needs skilled & trained people to break the application & demonstrate the problems or defects in the software product.

(i) Programmers/ Team must avoid Testing their own work products :-
Everybody is in love with the work product he/she has made. Also the approach of an individual

remains the same & bench approach related defects can't be found in self review.

-A second option is essential which can add value to a work product.

(ii) Throughly Inspect Results of Each Test to Find

-Test results show possibilities of weaker areas in the work product and the problems associated with the processes used for developing a work product.

- The defects found in test log do not form an exclusive list of all problems with the appln but indicate the areas where development team & management must perform a root cause analysis.

- Defects indicate process failures.

(iii) Initiate Actions for Correction, Corrective Action & Preventive Actions :-

- Defect identification, fixing and initiation of action to prevent further problems are the natural ways of making better products & improve processes.

- Corrections of the defect are done by the developer But one must ensure that corrective and preventative actions are initiated for making better products & again & again.

- Established that a program does what it is supposed to do, is not even half of the battle & rather easier one than establishing that program does not do what is not supposed to do. This is negative testing driven by risk assessment for final users.

* Salient features of Good Testing :- (Ref Page No. 84)

-Defects indicate the quality of software under testing, development & test process used for making it.

- Testing is a life cycle activity where the testers take part in right from proposal stage till the application is accepted by the customer / user.

- Good software Testing involves.

(i) Capturing user requirements :-

-The requirements defined by the users or customer & some implied requirements represent the foundation on which software is built.

-User requirements involve technical, economical, legal, operational & system requirements

(ii) Capturing user needs :-

-User needs may be diffn from user requirement specified in software requirement specifications.

(iii) Design objectives :-

- Design objective stay why a particular approach has been selected for building software.

(iv) User Interfaces :-

- User interfaces are the ways in which the user interacts with the system.

(v) Internal Structures :-

- These are mainly guided by software design & guidelines or standards used for designing & development.

(vi) Execution of code :-

Testing is execution of a work product to ensure that it works as intended by customer of user & prevented from any proable missuse of risk of failure.

* Test Policy :- (Ref Page No 85).

-Test policy is generally defined by the senior management covering all aspects of testing. It decides the framework of testing and its stately in overall mission of achieving customer satisfaction. For project organisations test policy may be defined by the client while for product organisation, it is decided by senior management.

* Test Strategy or Test Approach :- (Ref Page No 85)

-Test strategy defines the action part of test policy.

-It defines the ways and means to achieve the test policy.

- Generally there is a single test policy at organisation level for product organisations while test strategy may differ from product to product customer to customer and time to time.

- Some of the examples of test strategy may be as follows

① Definition of coverage like requirement coverage or functional coverage or feature coverage defined for particular product / project of customer.

(ii) Level of testing, starting from requirements & going upto acceptance phases of the product.
(iii) How much testing would be done manually & what can be automated?
(iv) Number of developers of testers.

* Test Planning :- (Ref page No 85)

- Test planning is the first activity of test team. If one does not plan for testing then he/she is planning for failure.
- Test plans are intended to plan for testing throughout software development life cycle.
- Test plans are defined in the framework created by test strategy & established by test policy
- Test plan should be realistic and talk about the limitations & constraints of testing.

① Plan Testing Efforts Adequately with an Assumption that defects are there :-
- Software products have defects.
- These test planning should know the number of defects it is intending to find by executing the given test plan.
- Test plan should cover the number of iterations required for software testing to give adequate confidence required by customer.
- Test plan is successful if intended number of defects are found.

(ii) If Defects are not found, it is failure of Testing Activity :-

- There are many defects in Software.
- If no (less) defects are found in testing, then it does not mean that there are no (less) defects in the product.
- If more number of defects are found it means the development process is problematic.

(iii) Successful Tester is Not One Who Appreciates Development But One Who Finds Defects in the product.

- Success of testing is in finding a defect and not certifying that application or development process is good.
- Successful tester can find more defects with higher probabilities of occurence & higher severities of failure.

(iv) Testing is not a be completed at the end of development cycle :-

- Testing is not a certifying process.
- It is life cycle activity and should not be the last part of a development life cycle before giving the application to customer / user.

• Software testing includes.

- Verification or checking wether a right process is followed or not during development life cycle.
- Validation or checking whether a right product is made or not as per customers need.

Difference between verification & validation

	Verification	Validation
i)	Verification is an activity where we check the work products with reference to standards, guidelines & procedures.	Validation is an activity to find whether the software achives whatever is defined by requirements.
ii)	Verification is prevention based. It tries to check the process adherence.	Validation is detection based. It checks the product attributes.
iii)	Verification talks about process, standards and guidelines.	Validation talks about product
iv)	Verification is also termed as 'white box testing' or 'static testing' as the work product undergoes a review.	Validation is also termed as 'black box testing' or 'dynamic testing' as work product is executed
v)	Verification can find about 60% of the defects.	Validation can find about 30% of the defects.
vi)	Verification involves the - review - walkthroughts - inspection - audits	Validation involves all kinds of testing - system testing - user interface testing - Stress testing

Page No.:

Testing Process & Number of Defects Found In Testing :- (Ref Page No 87)

- Testing is intended to find more numbef of defects.
- Generally it is belived that there are fixed number of defects in a product & as a testing finds more defects, chances of the customer finding the defect will reduce.
- Actually the scenario is reverse. As we find more & more defects in a product, there is probability of finding some more defects.
- This is based on the principle that every appln has defects & every test team has some efficiency of finding defects.
- It is governed by test team's defect finding ability.

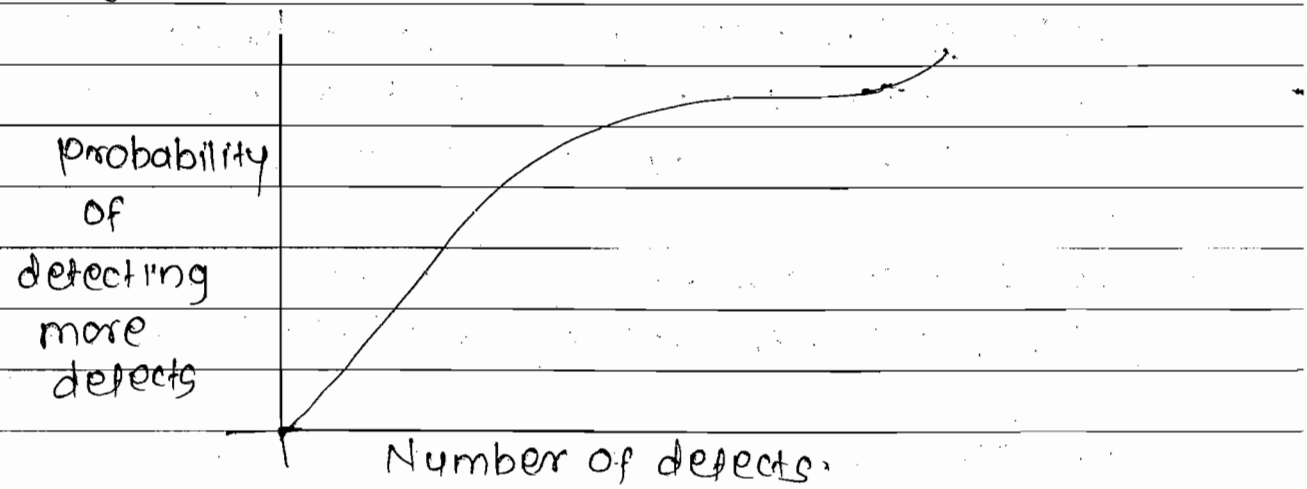

fig. Defect trend

- fig. shows relationship betn number of defects found and probability of finding more defects.
- The number of defects found after considerable testing will alway indicate possibilitiey of existing numbef of defects.

* Test Team Efficiency :- (Ref. Page No 87)

- Test team efficiency is a very important aspect for development team & management.
- If test team is very efficient in finding defects less iterations of testing are required.
- on other hand if development team is less efficient in fixing defects, mor iteration of testing & defect fixing may be required.
- The test team hay some level of efficiency of finding defects.
- Suppose the application has 100 defects and test team is very efficiency of 90% then it will be able to find 90 defects.
- Thus if the test team finds 180 defects (considering same efficiency), it means that there are 200 defects in the software product.
- Every test manager must be aware of the effciency of a test team that he/she is working with.
- Often test managers & project managers try to assess the test team efficiency at some frequency.
- The process is test team efficiency must be 100% but in reality, it may not possible to have test teams with efficiency of 100%. It must be very close to 100% in order to represent a good test team.
- Test team efficiency is dependent on Organisation culture & may not be improved easily unless organisation makes some deliberate efforts.

Page No.:

Example :-

The development team introducey some defects in a software product & gives it to the test team. The test team completey testing iterations as planned & gives number of defects found. The development team then analyses the defects found by the test team to understand how many defects have been found by test team. The ratio gives test team efficiency.

Suppose,

Defects deliberately introduced by development = X
Total defects found by testing team = Y
Defects found by testing team but not belonging to defects deliberated introduced by development = Z

The ratio of (Y-Z)/X will give test team efficiency

* Mutation Testing :- (Ref Page No 88)

- Mutation Testing is used to check the capability of test program and test cases to find defects.
- Test cases are designed and executed to find defects.
- If test cases are not capable of finding defects it is loss for an organisation.
- This is also termed as test case efficiency.

Example :-

A program is written & set of test cases are designed and executed on the program. The test team may find out few defects. Test cases are designed & executed to find defects. If test cases are not capable of finding defects, it is loss for an organisation The original program is changed and some defects are added deliberately. This is called mutant of the first pgm and process is termed mutation. It is subjected to the same test cases execution again. The test cases find out defects introduced deliberately in the mutant.

Suppose,

Defects deliberately introduced by development = X
Defects found by test case in original program = Y
Defects found by test cases in mutant.

The ratio of (Z-Y)/X will give test case efficiency
Theoretically it must be 100% but may not be exactly 100% due to following reasons

* Reasons for deviation of Test team efficiency from 100% for test team & mutation analysis

i) Camouflage effect :-

It may be possible that one defect may camouflage another defect & the test may not be able to see that defect or test case may be able to locate hidden effect.

ii) Cascading effect :-

It may be possible that due to existence of a certain defect, few more defects are introduced by testers.

iii) Coverage effect :-

It is understood that nobody can test 100% and there may be few lines of code or few combinations which are not tested to all due to some reasons.

iv) Redudant code :-

There may be parts of code which may not get executed under any condition as the conditions may be impossible to occur or some other conditions may take precedence over it.

* Challenges in Testing :- (Ref. Page No 89)

- Testing is challenging job in on diffn fronts.
- One one front it needs to tackle with problems associated with development team.
- One second front it has customers to tackle with.
- management may have problems with understanding testing approach & may consider it as an obstacle to be crossed before delivering the product to the customer.

* major challenges faced by test team as follows

(1) Requirements are not clear, complete, consistent, measurable & testable. There may create some problems in defining test scenario & test cases. Sometimes configuration management issue is faced when the development team makes changes in requirements but test team is not aware of these changes.

(ii) Requirements may be wrongly documented & interpreted by business analyst & system analyst. These knowledgeable people are supposed to gather requirements of customers by understanding their business workflow.

(iii) Code logic may be difficult to capture. often tester are not able to understand the code due to lack of technical knowledge.

(iv) Error handling may be difficult to capture. There are many combinations of errors & variety error messages & controls are required such as detective controls, corrective controls, suggestive controls & preventive controls.

* Test Team Approach :- (Ref. Page No 90)

- Type of organisation & type of product testing developed define a test team.

- There may or may not be separate team doing testing if management does not recognise it's importance, or the application under development demand this scenario.

- There are four approaches of software testing team.

(1) Location of test teams in an organisation.

- Generally test team is located in an organisation as per testing policy.

- It may vary organisation to organisation, project to project, customer to customer.

• Independent test team :-

- Independent test team may not be reporting to development group at all and are independent of development activities.

- They may be reporting to senior manager independent of development activities.

- Present of test manager is essential to lead the test team.

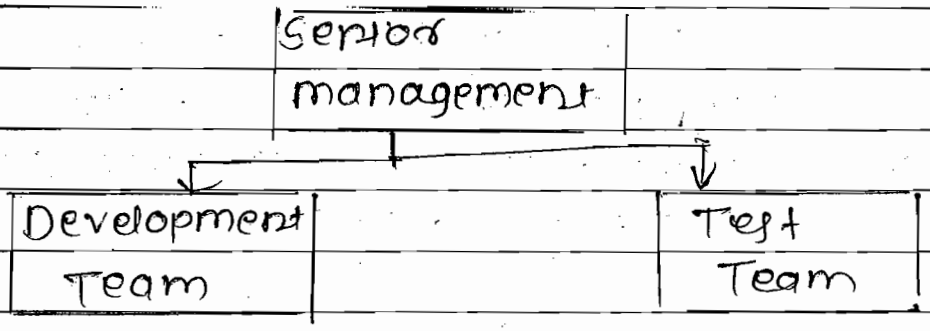

fig. organisational structure of test team independent of development team.

*Advantages of Independent Test Team :-

- Test team is not under delivery pressure.

- They can take sufficient time to execute complete testing as per depth of interations, coverage.

- Expert guidenance & mentoring is required by test team doing effective testing may be available in form of test manager

* Disadvantages of Independent of team

- Team strategy can be lost as developers take pride in what they develop while testers try to break the system.

- Testers may not be have good understanding of development process test team hide the process lacuna

* Test team Reporting to Development manager:-

- If the test team is reporting to development manager then they can be involved from start of project is finally closed. such team is

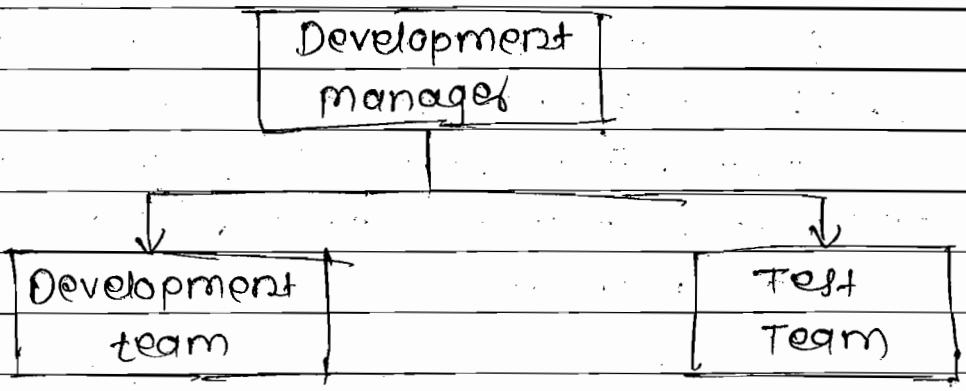

fig. organisational structure of test team reporting to development manager.

(ii) Developers Becoming Testers:-

- Sometimes those who work as developers in initial stages of development life cycle take the role of testers when the latter stages of life cycle are executed.

- Developers becoming testers can be suitable when the appln technologically heavy.

Adv:- Developers dont need another knowledge transfer while working as a tester.

- Developers have better understanding of design & coding & can test appln easily.

Disadv:- Developers may not find value in performing testing.

- Development needs more of a creation skill while testing needs more destruction skills.

(iii) Independent Testing Team :-

An organisation may create a separate testing team with independing responsibility of testing

Adv :- Seperate test team is supposed to conc more on test planning, approach & strategies
- This is independent view of work product that derived from requirement statement.

Disadv :- Seperate team means additional cost for an organisation.

(iv) Domain Experts Doing Software Testing :-

- An organisation may employ domain experts for doing testing. Generally this's approach is very successful in system testing & acceptance testing where domain specific is required.

Adv :- Domain experts may provide facilitation to developers about defects & customer expectations
- Domain expert understand scenario faced by actual user & testing is realistic

Disadv :- It may be very difficult to get domain experts in diverse area if an organisation has projects in diverse domains.
- It may mean huge cost for the organisation as these experts cost much more than normal developers / tester.

* Process problems faced By Testing :- (Ref Page No 94)

Most organisations consider that defects in the product are due to incorrect processes. In general it is belived that incorrect processes cause majority of the working problems. Defects are introduced in software due to incapable processes of development & testing. If the process of software testing is faulty it is give problems in terms of defects not found during testing but found by customer.

The basic constituent of processes are people, material, machines & methods.

(i) People :-

- Many people are involved in software development and testing, such as customer/user specifying requirements, business analyst documenting requirement, test manager defining test plan, testers defining test scenarios, test cases & test data available.

(ii) Material :-

Testers needs requirement documents, development standards & test standards, guidelines & other materials which add to their knowledge about prospective team

(iii) Machines :-

Testers try to build real life scenarios using various machines, simulators & environment factors.

(iv) methods :-

methods for testing a test planning, defining scenarios, test cases & test data may not be proper.

Page No.:

(V) Economics of testing :-

The cost of customers dissatisfaction is inversely proportional to testing efforts. More involement in testing efforts reduces cost of customer unhappiness.

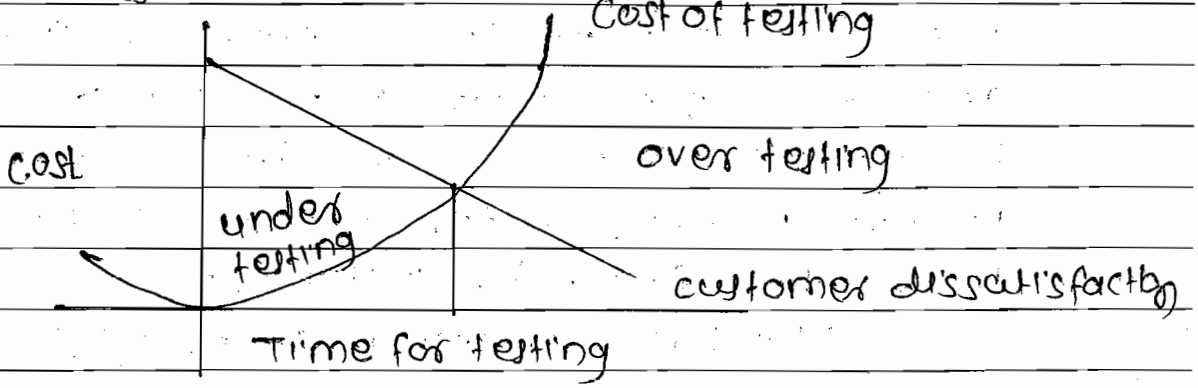

fig:- Cost of testing & cost of customer dissatisfaction.

Cost of testing curve is guided by the following
i) defect finding ability of testing team (test team efficiency)
ii) defect fixing ability of development team

* Cost Aspect of Testing :- (Ref page No. 95)

- Cost of quality includes cost of prevention, cost of appraisal & cost of failure.
- Testing may take some portion of these parts of costs.
- Testing is a costly affair & an organisation must try to reduce the cost of testing to the maximum extent possible.

e.g. project manager may get more rate than a developer or an architect may get more billing than a tester.

Example:-

Let us suppose that project duration is 10 months with 22 days of working per month, 8 hours working per day & 100 people are working. Also if conversion rate is say Rs. 500 per person hour then the cost of development & testing taken together will as as follows

Total efforts spent on project = 10×22×8×100
= 176.00hrs

Total cost = 176,000×500
= Rs 88,000,000

An organisation may have some additions to this cost in terms of contingencies, overheads & profit expected to arrive at sale price. If contingency is considered at 10% overhead apportionment is considered at 10% & expected profit is considered 20% then

Sales price would be = 81800,000×110%×120%
= Rs 127,776,000

* Cost of development / manufacturing includes the cost spent with following

i) Capturing the requirements, conducting analysis, asking queries & elicitation of requirements.
ii) Cost spent in designing the appln including high level designing & low level designing.
iii) Cost spent in writing code, integrating and creating the final product.

* Establishing Testing policy :- (Ref page No 99)

- Good testing is a deliberate planned effort by the organisation. It does not happen on its own, but detailed planning is required. Testing efforts need to be driven by test policy, test strategy or approach, test planning.

- Test policy is an intent of test management about how an organisation perceives testing & customer satisfaction. It should define test objectives & d. test deliverables.

- Test strategy or approach must define what steps are required for performing an effective testing.

- Test objectives define what testing will be targeting to achieve.

- Testing must be planned & implemented as per plan. Test plan should contain test objectives & methods applied for defining test scenario, test cases, test data.

- It should also explain how the results will be declared & how retesting will be done.

* Methods :-

- Generally methods applied for testing efforts are defined at organisational levels. They are generic in nature & hence need customisation. They are customised into a test plan and any tailoring required to suite a specific project may be done management directives are defined in test strategy.

- Testing strategy may be discussed with users/customer to get their views/buy in about testing. It may be accomplished through meetings & memorandums. User/customer must be made aware of cost of finding & fixing defects. Methods of using data or inputs provided by a customer must be analysed for sufficiency & correctness.

* Structured Approach To Testing :- (Ref. Page No 100)

- Testing that is concentrated to a single phase at the end of development cycle, just before deployment is costly.
- Testing is a life cycle activity & must be a part of entire software development life cycle.
- If testing is done only in the last phase before deliver to customer, results obtained may not be accurate & defect fixing may be very costly.
- Four components of wastes involved in this type of testing are given below :-

① Waste in wrong development :-

- Wrong specifications used for development or testing will result into a wrong product & wrong testing. This may lead to high customer dissatisfaction huge rework, retesting etc.

② Waste in Testing to Detect Defects :-

- If testing is intended to find all defects in product, then cost of testing will be very high. Effective reviews can reduce the cost of software testing and development.

(iii) Wastage as Wrong Specifications, Designs, Codes and Documents Must Be Replaced By correct specifications, Designs, Codes and Documents

- The cost of fixing defects may be very high in the last part of testing as there are more number of phases between defect introduction phase & defect may percolate through the development phases.

(iv) Wastage as System must be Retested to Ensure that the corrections are correct :-

- For every fixing of defect, there is a possibility of some part of software getting affected in a negative manner.

* Categories of Defect :- (for Page No. 101)

Software defects may be categorised under diffn criteria. The categories of defects must be defined in a test plan.

① On basis of requirement / design specification

- variance from product specifications as documented in requirement specifications or design specifications represents specification related defects. These defects are responsible for 'producers gap'.

- Variance from user/customer expectations as business analyst is not able to identify customer needs correctly. These are responsible for 'users gap'.

(ii) Types of Defects :-

- Wrongly implemented specifications are related to the specifications from what the customer wants. This may be termed as 'misinterpretation of specifications'.

- missing specifications are the specifications that are present in requirement statements but not available in final product.

- Features not supported by specification but present in the product represent something extra.

(iii) Root causes of Defects :-

- Wrong requirements given by user/customer can be a basic cause of defect.

- Business analyst/system analyst interprets customer needs wrongly can be another major cause of defect.

- System design architect does not understand the requirements correctly & architecture is wrong.

- Data entry caused by the users while using a product.

- Errors in testing - false call / failure to detect an existing in the product.

- Mistake in error correction, where defect is introduced while correcting some identified defect.

* Defect, Error, OR Mistake In Software :- (Ref. Page No 101)

- The problems with software work product may be put under different categories on the basis of who has found it and when it has been found.

Page No.:

	Mistake	Error	Defect
(i)	An issue identified while reviewing own documents or peer review may be termed 'mistake'.	An issue identified internally or in unit testing may be termed 'error'.	An issue identified in black box testing or by customer is termed defect.
(ii)	Very low cost of finding mistakes & can be fixed immed-iately.	slighty more cost of finding an error & needs sometime for fixing.	Most costly & needs longer time for fixing defects.
(iii)	Most of the time, problems and resolut-ions are not documented properly.	Sometimes, problems & resolutions are documented but may not be used for process improvements.	problems & resolution are officially documented & used for process improvements.

* Developing Test Strategy :- (Ref. Page No 102)

-Test planning includes developing a strategy about about how test team will perform testing. Some key components of testing strategy are as follows:

- Test factors required in particular phase of development.
- Test phase corresponding to development phase.

Process of developing test strategy goes through the following stages:

(i) Select and Rank Test factors for the Given application
- The test team must identify critical success factors / quality factors / test factors for the software product under testing.

(ii) Identify System Development Phases & Related Test factors :-
The critical success factors may have varying importance as per development life cycle phase.

(iii) Identify Associated Risks with each selected Test factor in case If it IS NO achieved :-
Trade offs may lead to few risks of development and testing the software.

(iv) Identify phase in which Risks of Not meeting a Test factor Need to be Addressed :-
The risks may be tackled in diff^n ways during development life cycle phase.

* Developing Testing methodolies (Test plan) (Ref. Page No. 102)

- Developing test practices is the job of project level test mananager / test lead. Diff^n projects may need diff^n tactics as per type of product / customer. Designing & defining of test methodology may take the following route.

i) Acquire & Study Test Strategy AS Defined Earlier
- Test Strategy IS developed by a test team familar with business risks associated with software usage.

(ii) Determining the type of development project being executed :-

- Agile methodology of development has small iterations of development & heavy regression testing.
- Iterative method of development has the continuously changing requirements and all other various technologies & code readability.
- Spiral development where new things are added in system again & again.

(iii) Determining the type of software system being made

- Type of software system defines how data processing will be performed by the software. It may involve
  - Determine project scope.
  - Changes to existing system such as bug fixing, enhancement & porting.

(iv) Identify Tactical Risks Related to Development.

- Risks may be introduced in software due to its nature, type of customer, type of developing organisation
  - Structural Risks
  - Technical Risks
  - Size Risks

(v) Determine When Testing Must Occure During Life cycle :-

- Testing phases starting from proposal, contract or requirement testing till acceptance testing & their integration testing.

* Testing Process :- (Ref Page No 105)

Testing is a process made of many milestones Testers need to achieve them, one by one to achieve the final goal of testing. Each milestone forms a basis on which next stage is built. They may vary from organisation to organisation, project to project following are few milestones commonly used by many organisations

i) Defining Test Policy
ii) Defining Test Strategy
iii) Preparing Test Plan
iv) Establishing Test objectives to be Achieved
v) Designing Test Scenarios & Test cases
vi) Writing & Reviewing Test cases
vii) Creation of Test Bed
viii) Executing Test cases
ix) Test Result
x) Test Result Analysis
xi) Performing Retesting when defects are Resolved
xii) Root cause Analysis & corrective/preventive Actions.

* Attitude Towards Testing :- (Ref Page No 107) (common People Issues)

- Attitude of development team & senior management or project management towards test team is a very imp respect to build morale of test team. It may be initiated from test policy & may be percolated down to test strategy design & test planning.

- New members of development team are not accustomed or project to view testing as a discovery process where defects are found in the product.
- We take pride on what we developed or we wish to prove that it is right or it is not my fault are very common responses.
- Conflict between developer & tester can create differences between project teams & test teams.

* Test Methodologies / Approaches (Ref Page No 107)

The two major disciplines in testing are given below :-

① Black box testing :-

- Black box testing is an attesting methodology where product is tested as per software specification defined by business analyst.
- Black box testing mainly talks about the requirement specification give by customer or intended requirements as perceived by tester.
- This testing is with the view as if a user is testing the system

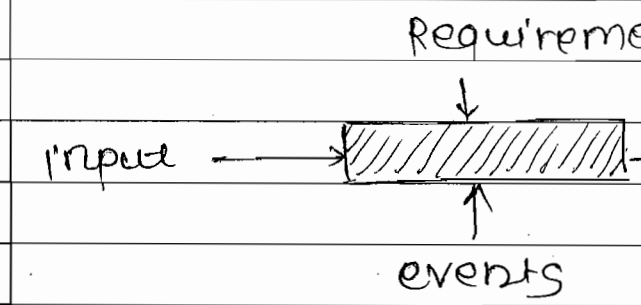

fig. Black box testing.

(ii) White Box Testing.

- It is a testing methodology where software is tested for the structure.
- This testing covers verification of work products as per structure, architecture, coding & standards & guidelines of software.
- It mainly deals with structure & design of the software product.

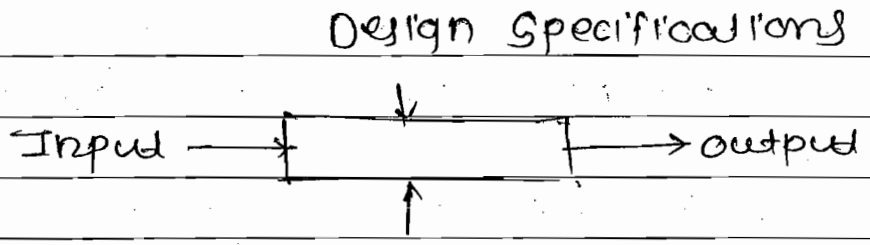

fig. White box testing.

(iii) Gray Box Testing :-

- This testing talks about combination of both approaches.
- Gray box testing is done on the basis of internal structures of software as defined by requirements, design, coding standards & guidelines

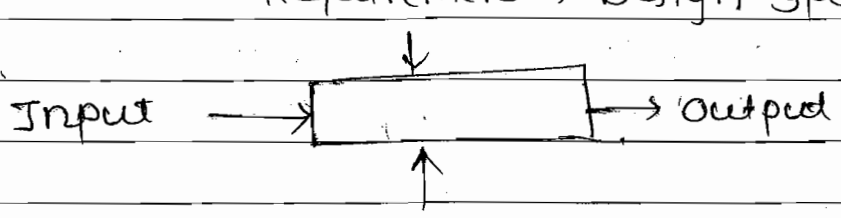

Fig Gray Box Testing.

Page No.:

* Raising Management Awarness for Testing.

(Ref Page No 11)

The management must be aware of the roles & responsibilities that testers are performing to achieve customer satisfaction by finding defects.

* Tester's Role :-

- While establishing a test function in an organisation, the management hay some objectivey to be achieved. Need to understand objectives & fulfill them.
- Calculating testing cost, effectiveness of testing & ensure that management understands the same.
- Demonstrate cost reduction & increase in effectiveness over a time span
- Highlights needs & benefits of training in team & development team on scrap reduce & skills so that testers can perform better.
- Collect & distribute information on testing to all them team members
- Get involved in budgeting.
- Testing needs people, money, time, training & other resources.
- The organigation may have to develop budget to procure all these aspects.

* Skills Required By Tester (Ref Page No 112)

- Testing needs a disciplined Approach.
- A tester is the person entrusted by an organisation to work as the devils agent.
- main purpose of testing is to demonstrate that defects are present, & point towards the weakey area in the software & processes used to build so that actions can be initiated in the direction.

(i) General Skills:-

- Written & verbal presentation skill
- Effective Listening skill
- Facilitation skill
- Software Devement, operations & Maintenance
- Continuous Education.

(ii) Testing Skills :-

- Concepts of testing
- Levels of testing
- Techniques for validation & verification
- Selection & Use of Testing Tools
- knowledge of Testing standards
- Risk Assessment & management
- Developing Test plan
- Defining Acceptance Criteria
- Checking of Testing Processes
- Execution of Test plan.
- Continuous Improvement of Testing Proces

* People challenges In Software Testing :-

- Testing is a process & must be improved continuously. People may need to analyse & take actions on the shortcomings
- Few expectations of software process improvement needs from testers are given below.

① Tester is responsible for improving testing process to ensure better products with less numbr of defects going to customer thus enhancing Customer satisfaction. All ~~defination~~ defects must be found & the confidence level must be built in the process that can give customer satisfaction. proper coverage as required by test plan must be achieved.

② Testing needs trained & skilled people who can deliver products with minimum defects to the stakeholders.

③ Tester have to improve their skill through continuous learning.

④ The tester needs a positive team attitute for creative destructive of software.

⑤ Testing is creative work & challenging task. feasible test scenarios & test cases & effective ways of looking for defects are essential to improve testing effectiveness.

⑥ programmers & testers work together to improve quality of software developed & delivered to customer.

⑦ Testing needs patience, ambition, creditability, capability & diligence on part of testers. Every defect must be seen from buisness perspective.

Page No.:

Book Referred:-

Textbook -

① M.G. Limaye "Software Testing Principles, Techniques and Tools", Tata Mcgraw Hill, ISBN : 9780701399090070139903.