

Chapter - 1

Introduction to Software Testing

1.1 Software Testing Background

Mistake (Errors):

- An error is an incorrect step in a program, an improper data definition (its type and size) or an incorrect result produced by a step in a program.

Bug:

- When people make mistakes while coding we refer them as bugs.
- It is generated during construction stage.

Defect:

- A defect is pre-existing condition in a finished product.
- It can be due to an error or the operating environment in which the software is being used.
- A defect can originate in one of the three stages (specification; design or development.)

Fault:

- A fault occurs when the operation of the software encounters a defect.
- A fault may result in failure if fault tolerance is not built into the system.

Fault tolerance:

- It is the set of mechanisms built into a product that provide a corrective action or an alternative action.

So that failure doesn't occur result.

failure:

- failure is the result of a product encountering a fault when in operation where a fault tolerance feature is not present.

There are 3 types of defects:

1. Critical defect

- Critical defects cause failures.
- They remain in the product until corrective action is taken to eliminate them.
- These defects must be fixed immediately.

2. Major defect:

- These are the defects for which a fault tolerance feature is built into the product.
- They don't need to be fixed immediately, but they should be corrected at the earliest possible time.

3. Minor defect:

- These are the defects that don't cause failure and no alternative action is necessary.

Example:

- * Wrong spelling on a screen.
- * Poor color contrast on a screen.

4. Test Suite:

- It is a collection of test cases for finding bugs.

Test log:

It is a chronological record of all relevant details about the execution of a test.

Test Report:

It is a document describing the conduct and results of testing carried out for a system.

which is the largest bug creator? why?

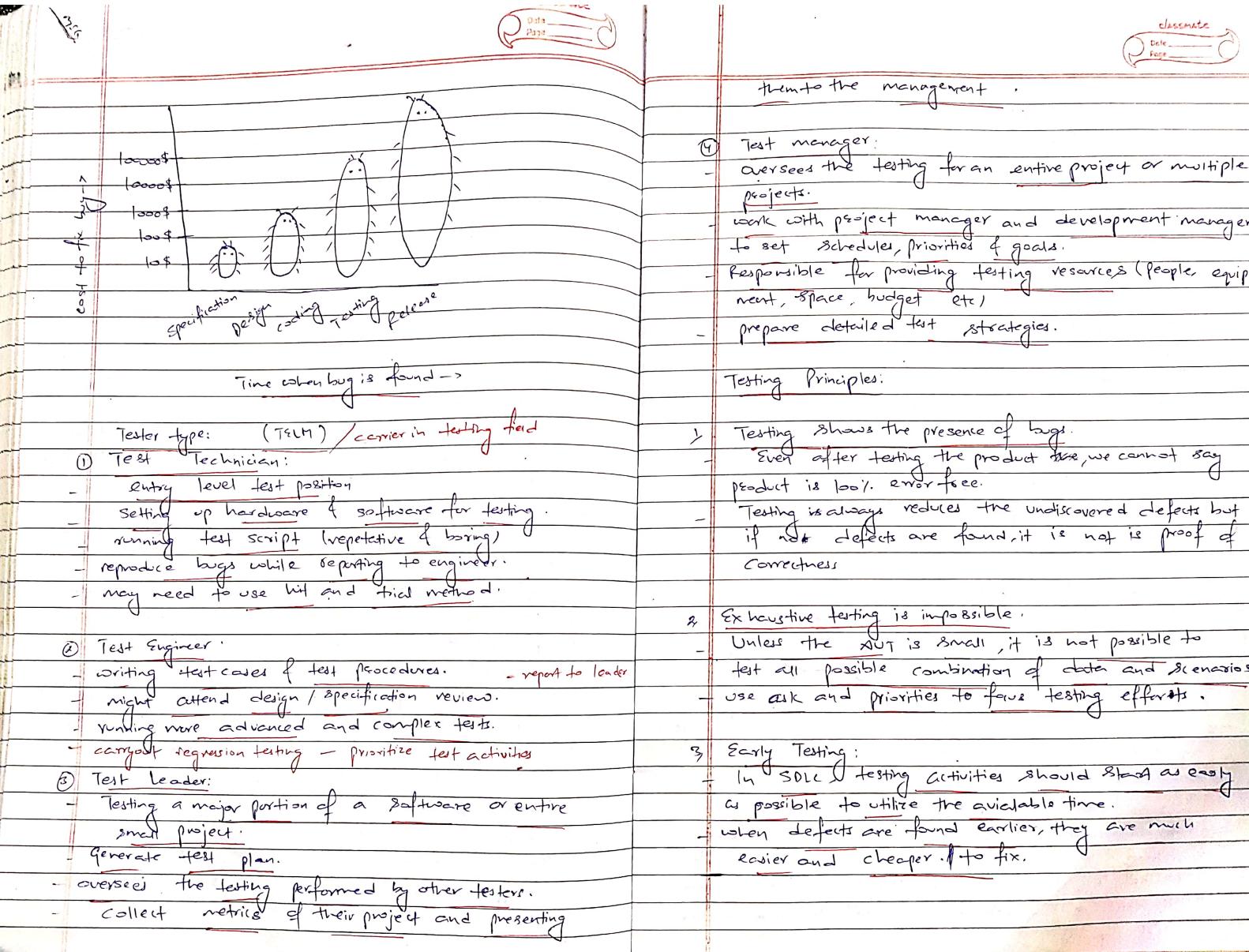
Specification:

- Specification is the agreement between client and the development team.
- It defines the detail about what the product will be, how it will act, what it will do, and what it won't do.
- -

Has it created bugs:

- Specification is the earliest phase of the development. When error occurs in this phase, the error will propagate and create other errors throughout the development cycle in an increasing rate.
- The error in specification occurs due to following reasons:

- ① Client cannot express their requirement to the development team to the fullest.
- ② The development team may not fully understand the requirement of client.
- ③ Technology and market demand constantly changes.
- ④ Inappropriate estimation of time, budget schedule and manpower.



4) Defect clustering:
This is the application of "Pareto Principle" to software testing.

The Pareto principle states that:
"Approximately 80% of the problems are found in 20% of the module."

5) Pesticide Paradox:
If same kinds of test are repeated again and again, eventually the same set of test cases will be unable to find any more defects.
So, to overcome this, test cases must be reviewed and updated as time passes.

6) Testing is context dependent:
Different kinds of SW are tested differently.
life threatening problems should be tested first than entertainment related issues.

7) Absence of error fallacy (エラー)
- If testing does not find errors, then it doesn't mean that the software is ready to be shipped.
- other factors are also needed to be considered before shipping.
- If the system built is unusable and does not fulfill user needs, then finding and fixing defects won't help.

8) keep software development static during testing
- Don't continue development of the software during testing
15) To be effective, software testing should be conducted by independent testers who are not involved in the development

13) Software testing should be planned prior to start of testing.
S/W testing should start with the smallest classmate unit of S/W and progress gradually towards the entire system.

9) ABC Test Parity (ABC of organization for testing)

10) Use experienced person for testing.

11) Use both positive and negative approach for testing.
Customer requirements should be the basis for all testing.

Testing
Testing is the process of evaluating a system by manual or automated means to verify that it satisfied requirements

- IEEE
The process of evaluating a program as system with the intent of finding errors.

- Myers

Importance of testing (why Testing?)

1. To point out defects and errors.
2. To ensure customer's reliability and their satisfaction in the application.
3. To ensure all product functionalities work as per the specification.
4. To ensure that the application should not result into any failures because it can be very expensive to fix it in the future or in the later stages of development.
5. To increase the name and fame of the company and stay long in the business.
6. To ensure that the product delivered does not have defect.
7. To improve the quality.

- 8) To ensure something that works when one person is using, also works when hundreds of people are using it.
- 9) To ensure the product will be compatible with lots of different devices, browsers and operating systems.

testing Testing? How to do testing? → 2 ways

i) Demonstration oriented (live way)

- * to show software works.
- * to satisfy the customer.
- * Software is demonstrated in normal flow.
- ii) branches may not be tested.
- * developers focus on this approach.

ii) Destruction oriented (live way)

- * Sociistic approach
- * to show software doesn't work.
- * tester focuses on this approach.
- * developer mainly focus on this approach.

Who performs testing?

- * every team member and not only test team.
- * Start with beginning till of SDLC still end.
- * a tester should have knowledge in two areas.
- i) Software testing techniques
- ii) Application Under Test (AUT)

When? (to stop)

- * testing is potentially endless
- * can't test till all the defects are unearthed & removed.
- * at some point, we have to stop the testing and ship the software.

* question is when?

* testing is trade-off between time, budget and quality

of approaches:

* Pessimistic: whenever some or any of the allocated resources are exhausted. (finished)

* Optimistic:

when either reliability meets the requirements or benefit from continuing testing can't justify testing cost.

d.4) (test plan is done while doing project planning)

1. Test Planning And Control

→ Test Planning has following measure tasks:

- i) To determine the scope and risk of testing.
- ii) To determine objectives of testing.
- iii) To determine test approaches.
- iv) To implement test policy.
- v) To determine the required test resources like people, test environment etc.
- vi) To schedule analysis and design - test implementation, execution and evaluation.
- vii) To determine the exit criteria of testing.

Test control has the following measures tasks:

- i) To measure and analyse the results of testing.
- ii) To monitor and document test progress, test coverage and exit criteria.
- iii) To provide information on testing.
- iv) To initiate corrective actions.

- 2, Test Analysis And Design:
- Test Analysis And Design has following major tasks:
 - i) To review Test basic) Test basic is the information that is needed in order to start the test analysis.
 - ii) To design test case.
 - iii) To identify required infrastructures and tools for testing.
 - iv, To identify the test condition.

3, Test Implementation And Execution:

- ⇒ During test implementation and execution, we take the test conditions into test cases.
- * Test implementation has the following major tasks:
 - i) To develop and prioritize test cases and create test data for those tasks.
 - * To create test suites from the test cases for efficient test execution.
 - * To verify and implement the test environment.
- ⇒ Test execution has the following major tasks.
- * To execute test cases and test suites following the test procedures.
 - * To compare actual results with expected results.
 - * To report differences between actual and expected results.
 - * To reexecute the test that previously failed in order to confirm or fix.
 - * To develop test log.

- 4, Evaluating Exit Criteria And Reporting
- After execution and running test for some time, exit criteria comes into picture. when;
- i) Maximum test cases are run with certain pass percentage.
 - ii) Bug rate falls below the certain level.
 - iii) When achieved the deadlines (Budget quality on time).
- ⇒ Evaluating exit criteria has following major tasks.
- i) To determine whether exit criteria has fulfilled or not.
 - ii) To determine whether the exit criteria specified should be changed when more test is required.
 - iii) To write test summary report for stakeholders.

5) Test Closure Activities:

- Test can be closed when :
- i) All information has been gathered which are needed for testing.
 - ii) When the project is canceled.
 - iii) When Some target is achieved.
- Test closure activities has the following major tasks:
- * To finalize and archived testware for later use.
 - * To handover the testware to maintenance organization.
 - * To evaluate how the testing went and learn lesson for future project.

3.5 The Psychology of testing:

- There are different perspective of testing from different people.
- ii) Comparison of the mindset of the tester and developer
 - The developers are waiting positively to solve the problems during the development process to make the product according to the user specification.
 - However while testing, tester are looking for the defects and failures in the product.
- Thus building the software requires a different mindset from testing the software.

2. The balance between Self testing and independent testing

- The tester can be a programmer and also the programmer can be a tester; although they often are separate roles.
- The programmer can test their component which they build.
- The developer always test their own code before giving it to any one.
- However we all know that it is difficult to find our own mistakes. So, developers depend on others to help test their work.
- The other person might be some other developer from the same team or test team specialist or professional tester.
- Giving application to the testing specialist or professional testers allows an independent test of the system.

field range etc) may lead to errors in code.

- Also faulty compilers, debuggers etc may invite errors and make difficult to debug them

iii) Buggy third-party tools

- During Software development, we require many third party tools like class libraries, editors, debuggers, plug-ins, API etc.
- A bug in such tool may cause bugs on SW being developed.

iv) Lack of skilled testing.

- lack of seriousness for testing and scarcity of skilled testers may increase the risk of errors

v) Last minute change:

- last minute change can cause error in SW.

Mistake: (Error)

A human error that produces an incorrect result

Bug:

When people make mistakes while coding, we refer them as bugs

Fault: An incorrect step, process or data definition in a program. (resulting from an error)

Failure: A failure occurs when a fault executes.

[Inability of system to perform its required function within specified limits].

Test: Testing is concerned with errors, faults & failures

- It is the act. of exercising software with test cases.

- It has 2 different goals.

* to find failures or,

* to demonstrate correct execution

Test case:

A test case has an identity and is associated with program behaviour. It has a set of inputs & a list of expected outputs.

Test Suite:

a collection of test cases for finding bugs.

Test log:

a chronological record of all relevant details about the execution of a test.

Test report:

a document describing the conduct and results of testing carried out for a system.

Specification:

an agreement among various parties especially client & development team. It defines the product they are creating, detailing what it will be, how it will act, & what it won't do.

Specification is largest bug creator

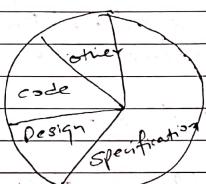


fig: causes for bugs.

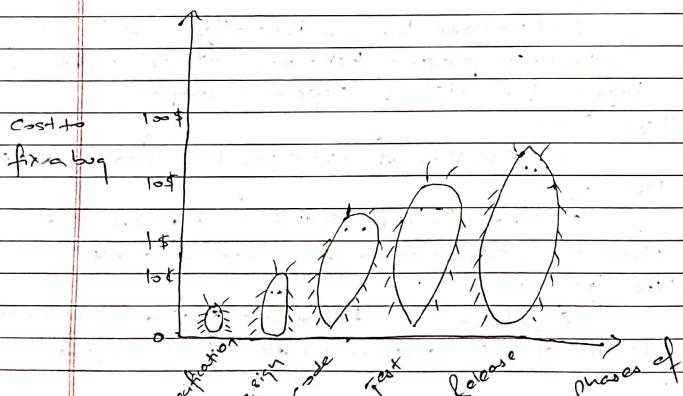


fig: cost of bug increases dramatically over time.

Testers type:

- ① Test technician
- ② Test Engineer
- ③ Test lead
- ④ Test Manager

① Test Technicians:

- Entry level test position
- setting up l/w/o, s/w configuration, running test, script/test automation, working with projects to reproduce bugs.
- same work can be repetitive and boring.

② S/w Test Engineer:

- running more advanced and complex tests.
- writing own test cases & test procedures
- reporting bugs & repeats.

③ S/w test lead:

- testing a major portion of a s/w project or entire small project.
- generate test plan for their areas & oversee the testing performed by other testers.
- involved in collecting metrics for their product and presenting them to management.

④ S/w test manager:

- oversees the testing for an entire project or multiple projects.
- test lead (er) reports to them.
- work with P.M (Project Manager) & development managers to set schedules

priorities and goal.

- responsible for providing appropriate testing resources (people, equipment, space) for their projects.
- prepare detailed test strategies

What is testing?

Defn:

- Process of evaluating a system by manual or automated means to verify that it satisfies specified requirements.

- IEEE 83a

"Process of executing a program or system with the intent of finding errors."

- Myers

- Testing ensures the correctness of product.
- It helps to enhance the quality of product.
- Quality product always enhance the customer confidence in using the product thus increasing business economics.

Why Testing?

- To find errors.
- To ensure that s/w delivered does not have defects.
- To ensure all product functionalities work as per the specification.
- Post-release removal of defects is most expensive.
- To improve quality.
- buggy s/w can be hazardous to

data, property and life.

How to do Testing?

- Demonstration oriented (+ve) way
 - * to show s/w works
 - * to satisfy the customer
 - * s/w is demonstrated in normal flow.
 - * all branches may not be tested.

- Destruction oriented (-ve) way.

- * Random approach.
- * to show s/w doesn't work
- * testing focuses on this approach.

Who carry out testing?

- every team member & not only test team.
- Starts with begining till end.
- A tester should have knowledge in 2 areas
 - ① s/w testing techniques
 - ② AUT (Application Under Test)

When? (to stop)

- testing is potentially endless.
- can't test till all defects are unclear & removed.
- at some point, we have to stop testing and ship the s/w.
- question is when?
- testing is trade-off between time, budget and quality.

2 approaches:

* pessimistic

whenever some or any of the allocated resources are exhausted

* optimistic:

when either reliability meets the requirement or benefit from continuing testing can't justify testing cost.

imp

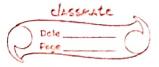
Testing Principles

~~Testing~~ Testing shows the presence of bugs.

- testing can show the defects are present but cannot prove that there are no defects.
- even after testing the product, we cannot say product is 100% error free.
- testing always reduces no of undiscovered defects but even if no defects are found, it is not a proof of correctness.
- thus, important to design test cases which find as many defects as possible.

2) Exhaustive testing is impossible:

- unless the AUT is small (simple logical structure & limited input), it is not possible to test all possible combination of data & scenarios.
- testing everything including all possible combination of input & precondition is not possible.
- So instead of doing the exhaustive testing we can use risks and priorities to focus testing efforts.



- also, budget, time & quality come into consideration.

3) Early Testing

- In CDT, testing activities should start as early as possible & should be focused on defined objectives
- the sooner we start testing, the better we can utilize the available time.
- when defects are found earlier in the life cycle, they are much cheaper and easier to fix.
- It is cheap and easy to change an incorrect requirement than having to change a functionality in large system that is not working.

4) Defect clustering:

- small no. of modules contain most of the defects in the system.
- Most of the reported defects are related to small no. of modules in the system.
- This is the application of 'Pareto Principle' to S/W testing - approximately 80% of the problems are found in 20% of the module.

5) Pesticide Paradox:

- If same kinds of tests are repeated again and again eventually the same set of test cases will no longer be able to find any new bugs.
- To overcome this 'pesticide paradox', it is really important to review test cases need to be written to exercise different parts to find more defects.
- Anytime a fault is fixed, we need to do regression testing.

regression; bug fix gamepachi yester side effect laxa ki mai vannira test gache.

- 6) Testing is context-dependent:
 - different kinds of S/W are tested differently
 - ex: web apps, mobile apps, health-related S/W, space related S/W
 - all depends on type and nature of application.
 - life-threatening problems should be handled first then entertainment related issues.

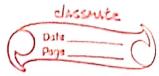
(3rd at 100% errors xaiya bhannu ৰাখো)

7) Absence of errors fallacy (mistaken belief)

- If testing didn't find any errors then it doesn't mean the S/W is ready to be shipped.
- other factors should be considered before shipping
- also if the system built is unusable & doesn't fulfill users' need then finding & fixing 'defects' won't help.

Reidherse
principle

- developer has to test own code
- experienced person choose genre (testing techniques)
 keep S/W static during testing of AUT
 (test Banque development happens)



Fundamental Test Process:

1) Planning and control

- Planning: determine scope & risks
- determine test approach
 - implement test policy
 - determine required test resources
 - determine exit criteria
 - create schedule

2) Control: measure & analyze results

- monitor and document progress
- make decisions
- initiate corrective action.

3) Analysis and Design:

1) review test basis

- identify test conditions
- design the tests.
- evaluate testability of requirement and system.
- design test environment set-up & identify infrastructure and tools.

3) Implementation & Execution:

- develop and prioritize our test cases by using techniques and create test data for those tests.

2) Create test suites.

- implement & verify environment.

Execution:

- run test suites following test procedures
- re-execute tests that failed earliest to confirm the fix.

- to log outcome
- to compare actual result with expected result.
- report discrepancies as incidents.

4) Evaluate exit criteria & reporting

- after execution & running tests for some time, exit criteria comes into picture when
- maximum test cases are run with certain pass percentage.
- bug rate falls below some level.
- when achieved deadline.

5) Test closure activities:

- testing can be closed when:
 - all information has been gathered which are given by testing.
 - a project is cancelled.
 - some target is achieved.

H last activities like:

- to check which planned deliverables are actually delivered.
- to finalize & achieve testware such as scripts, environment etc. for later use.
- to handover testware to (maintenance) organization.
- to evaluate how testing went and learn lessons for future projects.

Psychology of Testing:

- different perspective of different people.

- * comparison of the mindset of tester and developer
- * balance between self-testing & independent testing.
- * Degree of independence avoids author bias and is often more effective at finding defects and failures.
- * clear and courteous communication and feedback on defects between tester and developer.

Code of Ethics:

- a written set of guidelines issued by an organization to its workers and management to help them conduct their actions in accordance with its primary values & ethics standards.

* exercise honesty, diligence and objectivity in duty performance.

- * exhibit loyalty in all matters of organizations not engaged in acts or activities which are discreditable to the profession
- * refrain from entering any activities that may be in conflict with interest of their organization.
- * not accepting anything of value from an employee, client, etc. that would impair their professional judgement
- * continually strive for improvement
- * not leak (releas) sensitive information or news.

- * maintain / improve competency through continuing education.
- * co-operate for mutual professional benefit.