

Top Down Integration Testing

Defn We will add the modules incrementally or one by one to test the data flow in similar order.

Execution It will be executed on the structure or procedure oriented program.

Observation In this the observation of test output is more complicated.

Work will work on major to minor components.

Complexity Simple.

Managed from main module to sub module.

Adv In this, early prototype is possible.

Fault localization is easier.

Bottom Up Integration Testing

Lower level modules are tested with higher level modules until all the modules have been tested successfully.

Object oriented program

more accessible

will work on minor to significant components.

Complex & Highly Data Intensive

performed from sub module to main module.

We do not need to wait for the dev of all the modules as it saves time. Identification of defects is easy.

Bi Directional Integration Testing.

It is a vertical incremental testing strategy that tests the bottom layers & top layers and tests the integrated system in the computer software development process.

Using stubs it tests the user interface in its f Using drivers it tests the very lowest level functions

It is a kind of integration testing process that combines top down & bottom up testing

Process

- ① Bottom up testing starts from middle layer & goes upward to the top layer. For a very big system, bottom up approach starts at a subsystem level & goes upwards.
- ② Top down testing starts from middle layer & goes downward. For a very big system, top down approach, starts at a subsystem level and goes downwards.
- ③ Big Band App is followed for middle layer. From this layer, BUA goes upwards and TDA goes downwards.

Advantages

1. This app is useful for very large projects having several projects.
2. Both top down & bottom up starts at the start of schedule.
3. It needs more resources & big teams for performing both methods of testing at a time or one after the other.

Dis Adv

1. It is very high cost of testing.
2. It cannot be used for smaller systems.

System Integration Testing

It is the overall testing of whole system which is composed of many sub-systems.

The main objective of SIT is to ensure that all the software module dependencies are functioning properly & that data integrity is preserved b/w distinct modules of the whole system.

It also verifies a software system's coexistence with others & tests the interface b/w modules of the modules of soft appl.

In this type of testing modules are first tested individually & then combined to make a system.

Why do SIT

- ① It helps to detect defect early
- ② Earlier feedback on the acceptability of the individual module will be available
- ③ Correct Data Flow
- ④ Correct Control Flow
- ⑤ Correct Timing
- ⑥ Correct Memory Usage
- ⑦ Correct with software requirements

Scenario Testing

It is a software testing technique that makes best use of scenarios. Scenarios help a complex system to test better where in the scenarios are to be credible which are easy to evaluate.

Methods

- ① System Scenarios
- ② Use Case & Role Based Scenarios

Strategies to create good Scenarios

- Enumerate possible users, their actions & their objectives
- Evaluate users with hacker's mindset & list possible scenarios of system abuse

- List the system events & how does the system handle such requests.
- List benefits & create end to end tasks to check them.
- Read about similar systems & their behavior
- Studying Complaints about competitor's products and their predecessor.

Risks

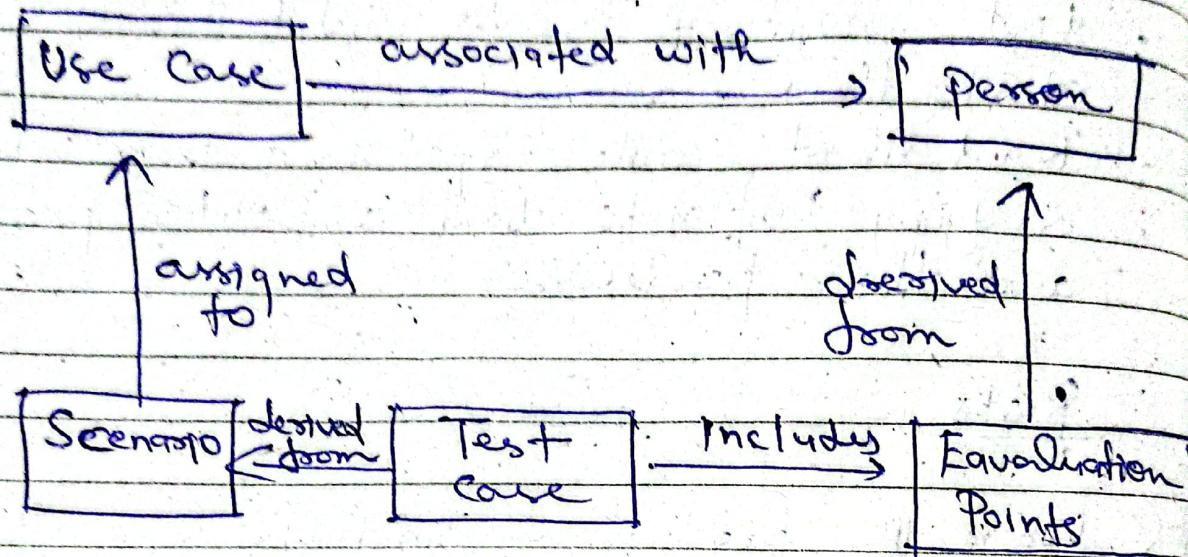
- ① When the product is unstable scenario testing becomes complicated
- ② Scenario Testing are not designed for test coverage.
- ③ Scenario tests are often heavily documented & used time again & again.

Characteristics

- ① Story
- ② Motivating
- ③ Credible
- ④ Complex
- ⑤ Easy to evaluate.

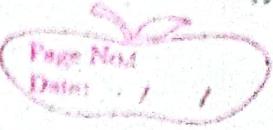
Scenario testing helps testers to know how the software will exactly work when end user will use it.

Scenario Testing Process



Defect Bash: colocated testing session performed by a group of people

- ① Defect Bash is an ad hoc testing.
- ② It is performed by different roles in an organization to test the product together at the same time such as all the developers, testers, program managers, researchers, designers.
- ③ It is not based on written test case.
- ④ What required to be tested ~~is~~ is up to individual's decision & creativity.



Functional Testing

- ① It verifies the operations of actions of an app.
- ② It is based on requirements of customer.
- ③ It helps to enhance the beh. of app.
- ④ Easy to execute manually.
- ⑤ It tests what product does.
- ⑥ Based on business requirement.

Examples

Unit Testing
Smoke Testing
Integration Testing
Regression Testing

Non Functional Test

- It verifies the behavior of an app.
- It is based on expectations of customer.
- It helps to improve the performance of ap.
- Hard to execute manually.
- It tests how product does.
- Based on performance requirement.

Performance Test

Load

Stress

Scalability

Deployment Testing

It refers to test installation process for developed software.

It is also known as Installation Testing or Implementation Testing.

The testing process may involve full, partial, or upgrades install/uninstall processes.

Objectives

- To identify end user problems when they will initially try to use the software.
- To measure the scalability & stability of system.
- To measure the performance rates such as discovery rate, data collection rate.
- To identify issues pro-actively that would occur in the deployment site.

Common Scenarios

- What are the common protocol used.
- How many type of proxies going to be supported.
- Is there will be a VPN.
- Is there any IP Filtering.
- Is there any User Authentication.

Beta Testing

Beta Testing is performed by real users of the software app in a real environment.

Beta Testing is one of the types of User Acceptance Testing.

It helps in minimization of products failure risks & it provides increased quality of the product.

It is the last test before shipping a product to the customers.

One of the major advantage is direct feedback from customers.

Characteristics

- ① It is performed by clients or users who are not employees of company.
- ② Reliability, Security & Robustness are checked during Beta Testing.
- ③ It commonly uses black box testing.
- ④ It doesn't require a lab or testing envr.

Types of Beta Testing

- ① Traditional Beta Testing
- ② Public
- ③ Technical
- ④ Focused
- ⑤ Post release

Tools

- ① TestFairy
- ② CenterCode
- ③ ToyMyUI
- ④ UserTesting
- ⑤ TestRail
- ⑥ UserSnap
- ⑦ Zephyrus
- ⑧ TestFlight

Advantages

- ① It reduces product failure risks.
- ② It helps in improving product quality.
- ③ Cost effective compared to similar data gathering methods.
- ④ It increases customer satisfaction.

PIS Advantages

- ① Sometimes it is complex to follow the errors or bugs because testing environment varies from user to user.
- ② There is chance of having duplication of errors or bugs.
- ③ The dev team & testing team have no control over this real time testing environment.
- ④ This testing is time consuming process.

Scalability Testing

is a type of non functional testing in which the performance of a software application, system, network or process is tested in terms of its capability to scale up or scale down the number of user request load or other such performance attributes.

Objective

To determine how the app scales with increasing workload.

To determine user limit for the software product.

To determine client side degradation & end user experience under load.

To determine client server side degradation.

Scalability Testing Attributes

- ① Response Time
- ② Throughput
- ③ Performance measurement with number of users
- ④ Threshold load
- ⑤ CPU Usage
- ⑥ Memory Usage
- ⑦ Network Usage

Adv.

- ① Define a process that is repeatable for executing scalability test
- ② It provides more accessibility to the product
- ③ It detects issues with web page loading and other performance issues.
- ④ It finds and fixes the issues earlier in the product which saves lot of time
- ⑤ It ensures the end user experience under load.
- ⑥ It provides customer satisfaction

Disadv.

- ① Sometimes it fails to find the functional errors, issues in product
- ② Some automation tools for scalability testing are costlier which increases the budget of tool
- ③ Team members involved in this testing should have high level of testing skills

Reliability Testing

X is a testing that relates to test the ability of a software to function & given environmental conditions that helps in finding issues in software design and functionality.

It is defined as a type of software testing that determines whether the software can perform a failure free operation for a specific period of time in a specific environment.

Objective

To find the number of failures occurring in a specific period of time.

To discover main cause of failure.

Types

- ① Feature : Each fun should be properly executed.
- ② Regression : is basically performed whenever any new functionality is added.
- ③ Load.

↳ to determine whether app. is supporting the required load without getting breakdown.

Measurement of Reliability Testing

① Mean Time Between Failures (MTBF)

② Mean Time To Failure (MTTF)

The time b/w two consecutive failures

③ Mean Time To Repair (MTTR)

The time taken to fix the failures

$$\boxed{MTBF = MTTF + MTTR}$$

Stress Testing

It determines the robustness of software by testing beyond the limits of normal op.

It is particularly important for critical soft but is used for all types of software.

It emphasizes robustness, availability, error handling under a heavy load.

It also verifies the stability & reliability of the system.

It is perform to ensure that the system would not crash under such situations.

It is also known as Endurance Testing or Torture testing.

Tools

Jmeter

Load Runner

Stress Tester

Neo load

Advantages

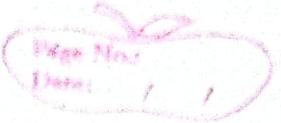
- ① determines the behavior of system after failure & ensure that the system recovers quickly.
- ② ensures that system failure doesn't cause security issues.
- ③ makes the system work in normal as well as abnormal cond

Disadv

takes a longer time to complete and it is a complicated process

Good scripting knowledge for implementing the script test cases

There is need for external resources to implement Stress testing



Types of Stress Testing

① Server-Client Stress Testing

testing is carried out all clients from server across

② Product Stress Testing

It concentrates on discovering defects related to data locking & blocking, network issues

③ Transaction Stress Testing

is performed on one or more transactions b/w two or more appl

④ Systematic Stress Testing

is used to perform tests across multiple systems running on same server

⑤ Analytical Stress Testing

is carried out to find defects in unusual scenarios like a large number of users logged at the same time

Acceptance Testing

It is a method of software testing where a system is tested for acceptability.

It is last phase of software testing performed after System testing

Unit Testing

Integrating Testing

System Testing

Acceptance Test

Use

- ① To find the defects missed during functional testing phase
- ② How well product is developed
- ③ A product is what actually the customers need
- ④ Feedback helps in improving the product performance & user experience

Regression Testing

It is the process of testing the modified parts of code and the parts that might get affected due to modifications to ensure that no errors have been introduced in the software after the modifications have been done.

Process

Firstly whenever we make some changes to the source code for any reasons like adding new functionality; optimization etc.

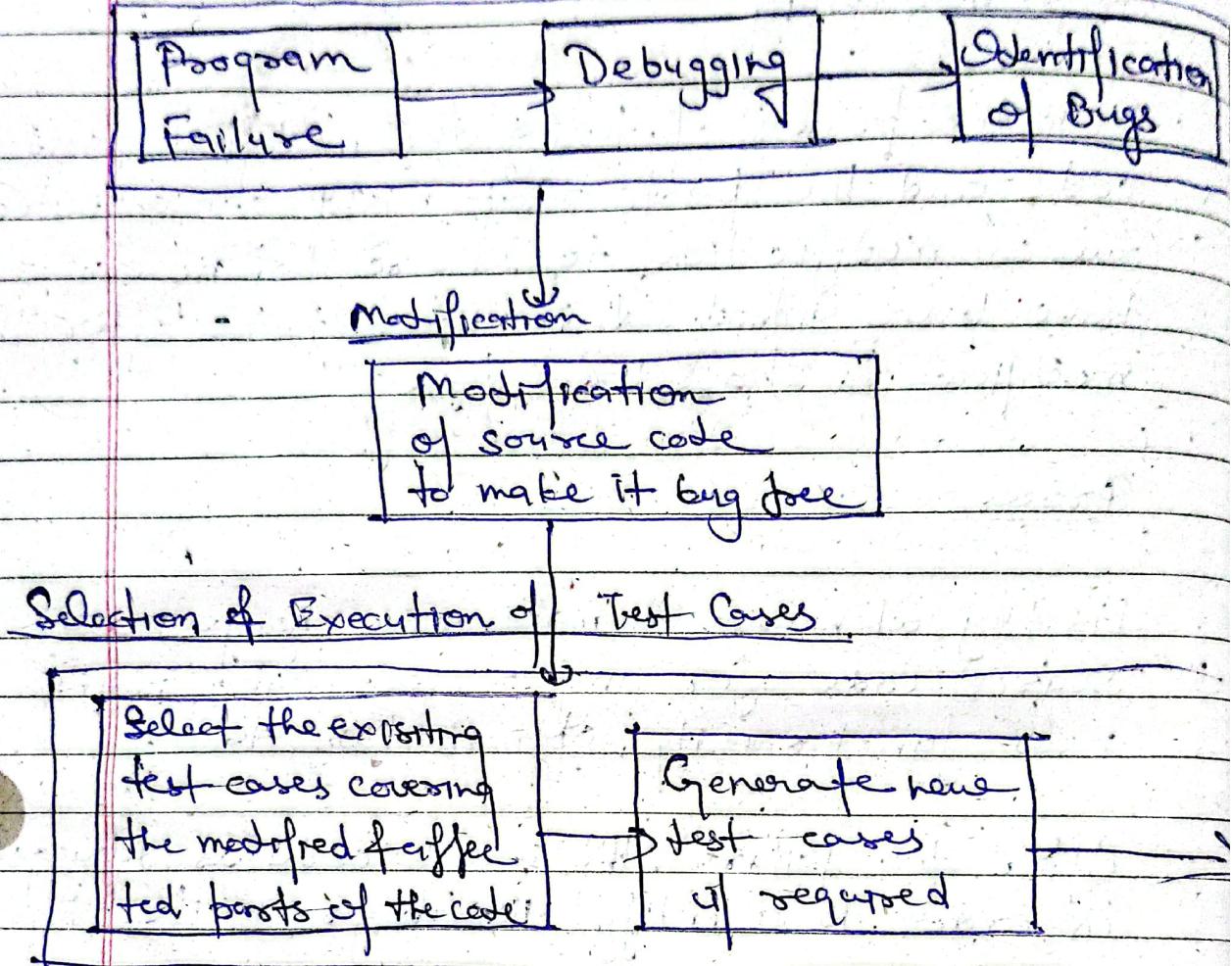
After the failure, the source code is debugged in order to identify the bugs in the program.

After identification of the bugs in source code appropriate modifications are made.

The appropriate test cases are selected from the already existing test suite which covers all the modified & affected parts of source code. We can add new test cases if required.

In the end regression testing is performed using selected cases.

Identification of Bugs



Techniques for the selection of Test Cases for Regression Testing

- Select All Test Cases

In this tech, all the test cases are selected from the already existing test suite. Most simple, safest tech. but not much off.

- Select Test Cases Randomly

In this tech test cases are selected randomly from existing test suite but it is only useful if all the test cases are equally good in fault

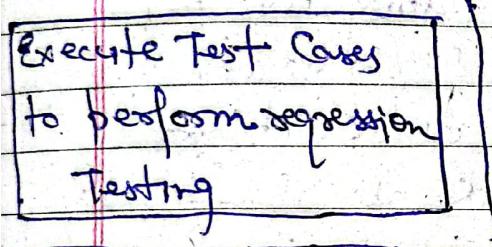
Detection capability which is very rare.

- Select modification covering test cases

In this techn only those test cases are selected which covers if tests the modified portions of the source code.

- Select higher priority test cases

Priority codes are assigned to each test case of test suite based upon bug detection capability, customer requirements.



Tools for Regression Testing

Selenium

WATIR (Web Application Testing in Ruby)

QTP (Quick Test Professional)

RFT (Rational Functional Tester)

Winrunner

Silk test

Adv

- Ensures that no new bugs has been introduced after adding new functionalities to the system

- It helps to maintain the quality of source code
- Can be easily automated by automated tools.

Disadv:

- It can be time & resource consuming if automated tools are not used.
- It is required even after small changes in the code

Ad hoc Testing

It is a type of software testing which is performed informally & randomly after the formal testing is completed to find out any loophole in the system.

It is also called as Random Testing or Monkey Testing.

It is not performed in an structured way so it is type of Unstructured Software Testing.

Ad hoc Testing has

- ① No Documentation
- ② No Test Cases
- ③ No Test Design

Types of Adhoc Testing

① Buddy Testing

where two bodies will be involved one is from developer team & one from tester team

② Pair Testing

where two bodies from the testing team can be involved to test the same module.

③ Monkey Testing

in which system is tested based on random inputs without any test cases.

Adv of Adhoc Testing

- The errors which can not be identified with written test cases, can be identified by Adhoc Testing.
- It can be performed within very limited time.
- Helps to create unique test cases.
- This testing can be performed anytime during SDLC.

DisAdv.

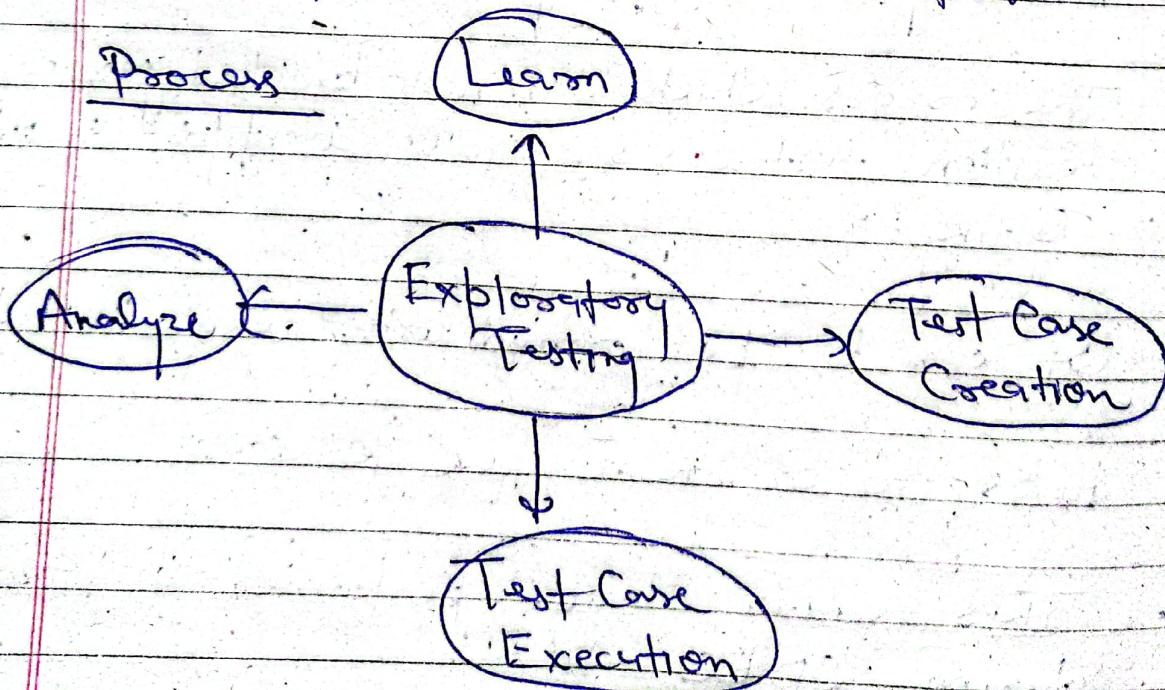
- Sometimes resolving errors is difficult as no written test cases f does one these.
- It does not provide any assurance that the error will be definitely identified.

Exploratory Software Testing

is a type of software testing in which the tester is free to select any possible methodology to test the software.

It is an unscripted approach for ST.

In this QDs use their personal learning, knowledge, skills & abilities to test.



Adv

- It takes no preparation as it is unscripted testing technique
- It finds critical defects very quickly.
- In exploratory testing testers use their knowledge, skills & exp to test

DisAdv

- In this once testing is performed it is not reviewed
- keeping track of tests performed is difficult
- It is not possible to repeat same test methodology.

Iterative Testing

It is a method in which a product is tested on user repeatedly

By making use of results, after each test the product is improved at different stages

The main motive to make the product foolproof and market ready before its launch

It is common

Defect Seeding

It is a practice in which defects are intentionally inserted into a program by one group for detection by another group.

The ration of no of seeded defects detected to the total no of defects provides a rough idea of the total no of unseeded defects.

Smoke Testing

It is a broad approach to test where all parts of app are tested.

measures stability of system by performing regression testing

- performed by both tester & developers

Whenever needed tester has to go into deep

Smoke Testing is undocumented

This is considered as subset of Acceptance Testing

Sanity Testing

It is narrow app to test where specific parts of app are tested

measures rationality

- performed by only testers.

does not need to go into deep of app

Not documented like Adhoc

subset of Regression Testing..

It is used to ^{test} end to end function of the app

It is used to test only modified or defect fixed functions

Slicing

Slicing is a technique used in software testing which takes a slice (groups of program statements) in the program for testing particular test conditions or causes that may affect a value at a particular point.

It can also be used for the purpose of debugging in order to find the bugs

Types
 → Static
 → Dynamic

Dynamic Slicing

A dynamic slice of program contains all the statements that actually affect the value of variable at any point for a particular execution of foo.

Dynamic Slices are generally smaller

Considers only a particular exec of program

Startie

Int n

cin>>n;

n=0;

y(n>10) $\xrightarrow{10}$

cout <<

else

cout <<

Int n

y(n>10)