

3) Levels of Testing (X) Special Tests

3.1 Unit Testing:

Unit Testing is a level of software testing where individual units / components of a software are tested. The purpose is to validate that each unit of software performs as designed. Unit tests focus on functionality and reliability, and the entry and exit criteria can be same for each module, or specific to particular module.

In unit testing, we require drivers and stubs. The driver acts as a calling unit and the stub acts as called unit.

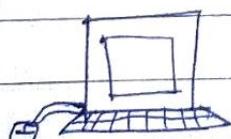


Drivers:

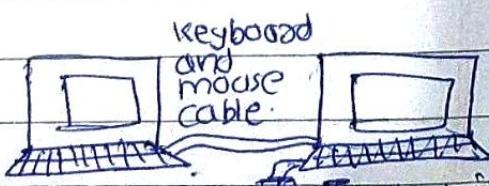
Drivers are tools used to control and operate the software being tested.

With some hardware modification and few software tools, we can replace keyboard and mouse of a system being tested with additional computer acts as a test driver.

We can write simple program on this driver computer that automatically generates the appropriate keystrokes and movements to test the software.



Normal
System Configuration



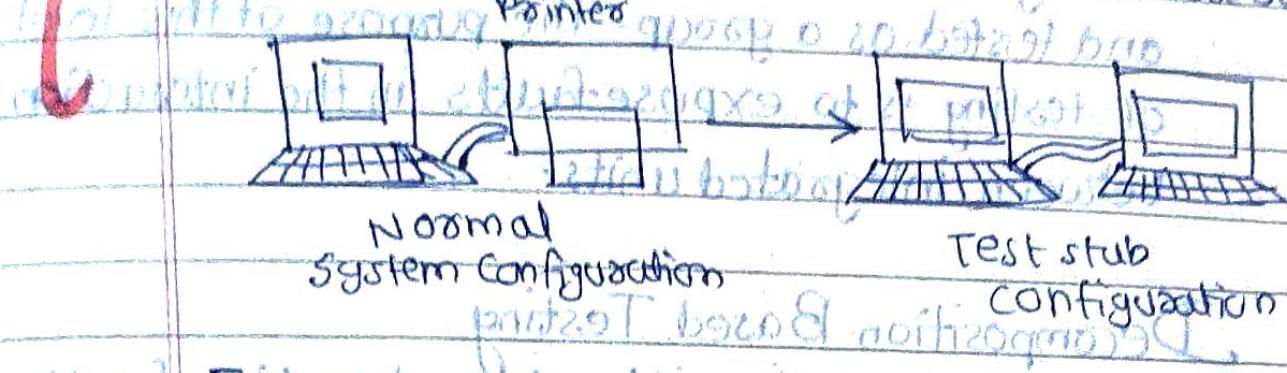
Test driver configuration

Fig:-

A computer can act as a driver & test tool to replace the keyboard and mouse of a system being tested.

~~Stubs~~:

Stubs have just opposite to drivers. They instead receive and respond to data that the software sends.



For example, you are testing software that needs to send data to a printer. You can do this in this case if you replace the printer with another computer that was running stub software that could read and interpret data.

Advantages of unit testing:

- ① Writing unit tests helps to produce high quality code in many levels.
- ② The availability of tests detects the introduction of bugs whenever the programmer adds new features or refactors code.
- ③ The act of writing tests encourages the programmer to write code in small chunks that can be tested independently.

(Collaborate to prod fig)

Helping hand 960 21st Jan 2019 9th 11A

3.2

Integration Testing :-

Integration testing is a level of software testing where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units.

Decomposition Based Testing

In decomposition based testing functions of an application / project are separated from each other. A functional hierarchy is created for that application / project mentioning the sequence of execution of functions. These separated units can be executed sequentially or simultaneously.

This testing is carried out using the following main strategies and they are:-

1] Non-incremental integration testing

2] Incremental integration

3] Top down testing

4] Bottom up testing

5] Bi-directional integration testing

1] Non-incremental integration testing

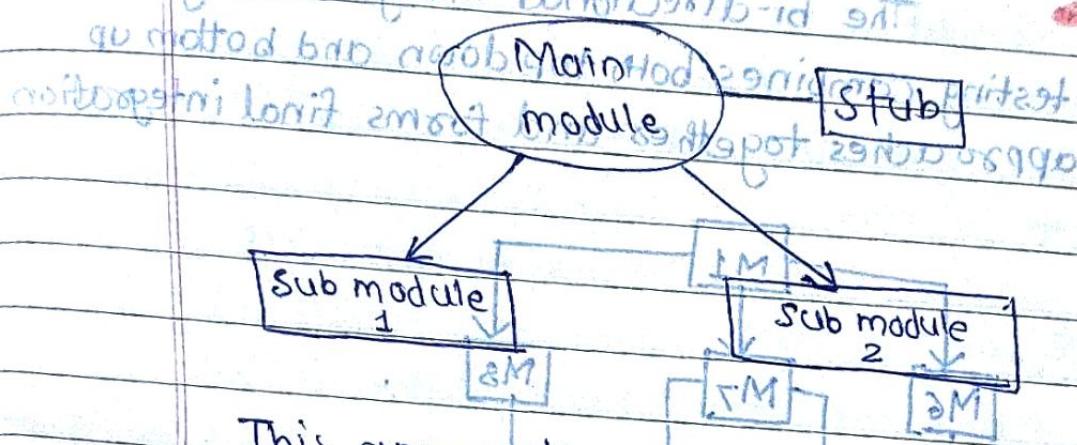
(Big bang or umbrella)

All the software units are assembled into an entire program. This assembly is then tested as a whole from the beginning. In this, causes of the defects are not easily isolated and corrected.

~~Ques 2] Incremental integration testing :-~~

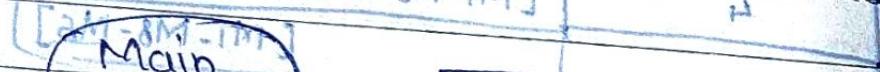
The program is constucted and tested in small increments by adding a minimum number of components at each interval. Therefore the errors are easier to isolate and correct, and the interfaces are more likely to be tested completely.

~~3] Top-down incremental integration testing :-~~



This approach requires the highest level modules to be tested and integrated first. The testing integration testing starts with the highest level control module, or main program, with all its subordinates replaced by stubs.

~~4] Bottom up incremental integration testing~~



~~[S1M - S2M - S3M - S4M] [S1M1 - S1M2 - S2M1 - S2M2]~~

~~Main~~

~~Drives~~

~~Sub module 1~~

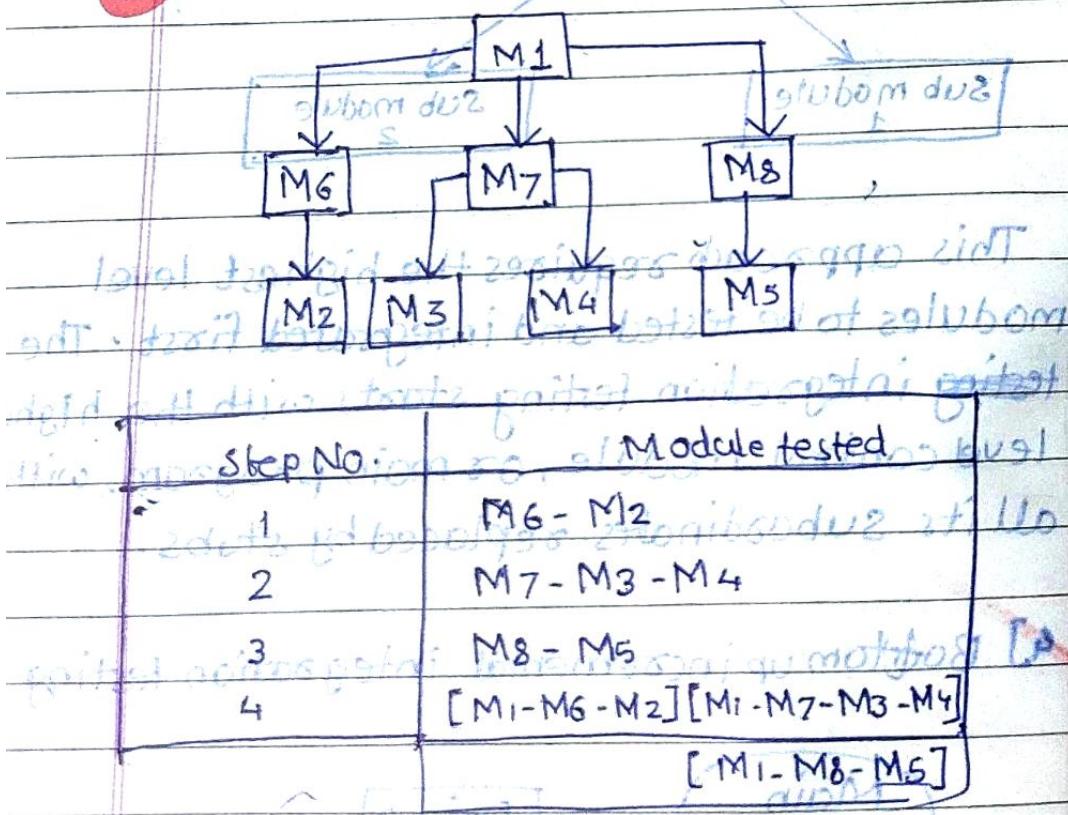
~~Sub module 2~~

The bottom up approach requires the lowest level units to be tested and integrated first.

The lowest level submodules are integrated and tested, and then the successively superior level components are added and tested, transmitting the hierarchy from bottom side to upward side.

~~4~~ 5] Bi-directional integration testing:

The bi-directional integration type of testing combines both top down and bottom up approaches together and forms final integration.



In this case, drivers work in bottom up fashion and stubs work in top-down (downward) way. After this, functionality of integrated modules is checked and stubs and drivers are removed.

3.3 System Testing :-

C System Testing concentrates on testing the complete system with variety of techniques and methods. This is done to compare program against its original objectives.

It tests all components and modules that are new, changed, affected by changes or needed to form the complete application.

This testing is classified into 4 divisions :-

1] Usability Testing (Ease of use or not) A Low Priority

2] Functional Testing (Functionality correct or not)

3] Performance Testing (Speed of processing) Medium priority

4] Security Testing (To break the security of system) High Priority.

Recovery Testing :-

It is a testing aimed at verifying system's ability to recover from varying degrees of failures. During this test, a test engineer validates that whether application build can bring recover from abnormal situations or not.

Recovery testing is basically done in order to check how fast and better the application can recover against any type of crash or hardware failure.

Ex:- During process power failure, network disconnection, server down, database disconnected

✓ Security Testing :-

Security testing attempts to verify that protection mechanisms built into a system will protect it from improper penetration.

During security testing, password cracking, unauthorised entry into software, network security are all taken into concern. To conduct these tests, highly skilled persons who have security domain knowledge.

This test is divided into three subtests:-

* Authorisation :- Verifies author's identity to check he is a authorised user or not.

* Access Control :- Test the rights given to the user to do the system tasks.

* Encryption / Decryption :- Encryption is to convert actual data into a secret code which may not be understandable to others and decryption involves converting the secret data into actual data.

✓ Performance Testing :-

Performance Testing is done to validate whether the output of particular program is given within a particular time period.

One of the main objective of performance testing is to maintain a website with low response time, high throughput and low utilization.

QUESTION

ANSWER

* Response Time:- It is the delay experienced when a request is made to server and its bandwidth. Server's response to the client is received.

* Throughput:- It refers to the number of bits or Client requests processed within a certain unit of time. Typically, the unit of measurement is requests per second or Pages per second.

This performance test is classified into below sub tests:-

- I] Load Testing III] Data Volume Testing
- II] Stress Testing IV] Storage Testing

~~I] Load Testing:-~~ It is defined as the testing to determine whether the system is capable of handling anticipated number of users. The objective of load testing is to check whether the system can perform well for specified load.

~~II] Stress Testing :-~~ Stress testing is conducted for the number of concurrent users beyond the specified limit. The objective is to identify the maximum number of users the system can handle before breaking down.

(Total 10 Ques.)

III] Data volume testing:-

Volume Testing is to find weaknesses in the system with respect to its handling of a large amount of data during short time period. Tester conducts this test to find maximum size of allowable maintainable data by our application build.

IV] Storage testing:-

Execution of our application under huge amounts of resources to estimates storage limitations to be handled by unit. Application is called as storage testing.

~~Usability Testing~~

~~Usability testing is a way to see how easy to use something is by testing it with real users. It is a gold standard by which you can determine if the design of an application meets the need of intended users and allows them to work productively.~~

~~It consists of following main aspects :-~~

~~* Ease of use (understandable to end users to operate)~~

~~* Look and feel (Pleasantness or attractiveness screen)~~

~~* Speed in interface (Less no. of events to complete task)~~

Compatibility Testing :-

It is a type of software testing to check whether your software is capable of running on different hardware, operating systems, applications, network environments or mobile devices. It is a non-functional test which primarily focuses upon the application's suitable performance in presence of an in relation to other programs.

Backward and Forward compatibility :-

Backward compatibility testing ensures new version of the product to continue to work with the older product.

For example, if a newer software version cannot save files that can be read by the older version, it is not backward compatible with older version.

A forward compatible system is expected to gracefully handle input which is intended for a newer version by ignoring the unknowns and selecting the known subset of data that the system is capable of handling.

An example of forward compatibility is a web browser ignoring the HTML tags that it does not yet recognize or handling them in a standard way.

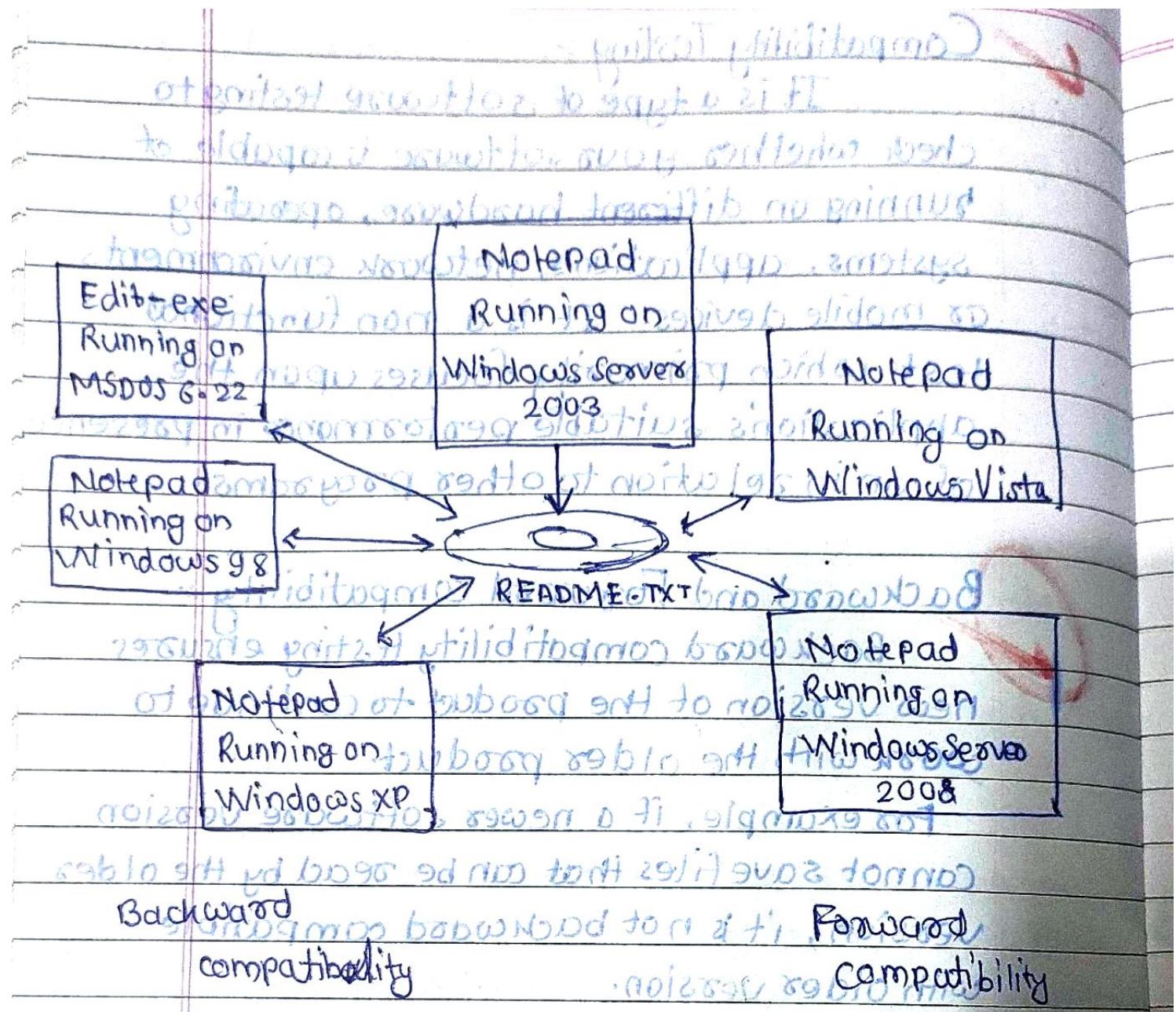


Fig- Backward & Forward compatibility defines what version will work with your software data files.

3.4 Acceptance Testing: <http://www.guide2qa.com>

~~REDO~~ Acceptance Testing is a level of software testing where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and assess whether it is acceptable for delivery.

Acceptance testing allows customers to ensure that the system needs meets their business requirements.

The types of acceptance testing are:-

(a) User Acceptance test: It focuses mainly on the functionality thereby validating the fitness for use of the system by the business user.

(b) Operational Acceptance test: It validates whether the system meets requirements for operation. It may include backup/restore, disaster recovery, maintenance tasks and periodic check of security vulnerabilities.

(c) Contract Acceptance test: It is performed against the contract's acceptance criteria for producing custom developed software. Acceptance should be formally defined when contract is agreed.

(d) Compliance Acceptance test: It is performed against the regulations which must be adhered to, such as governmental, legal, safety regulations.

Results of these tests will allow both the customers and the developers to be confident that the system will work as intended.

- Acceptance criteria :- It is defined on basis of following attributes :-
- * Functional correctness and completeness
 - * Data integrity
 - * Data conversion
 - * Usability
 - * Performance
 - * Timeliness
 - * Confidentiality and Availability
 - * Installability and Upgradability
 - * Scalability
 - * Documentation

Alpha Testing :-

It happens at the development site just before the rollout of the application to the customers. Alpha tests are conducted without replicating the live environment where the application would be installed and running. The test is conducted at the developer's own site only. In SDLC, depending on the functionalities the number of alpha phases required is stated in project plan.

Basic installation, in-installation tests, the completed core functionalities are tested.

Beta Testing :-

Beta testing comes after the alpha testing and can be considered a form of external user acceptance testing. Version of the software, known as beta versions, are released to limited audience outside the programming team. The software is released to groups of people so that further testing can ensure the product has few faults or bugs.

Advantages of Beta Testing

- * Get error free final product
- * Product is validated in error free environment
- * Save product cost due to product is tested before launch
- * Better customer satisfaction
- * Support team knows problem before launch.

Difference Point	Alpha testing	Beta testing
Conducted for	Software applications which is intended for specific customers	software product intended for market requirements
Conducted by	People other than development team inside the organisation	Outside people or customers.
Conducted at	Development site	Customer site

3.5 Special tests :-

The developer makes executable file by combining all these files. These executable file is known as software build.

~~Smoke Testing :-~~

Smoke testing is done to check the stability of the build received for software testing. It is performed after software build to ascertain that the critical functionalities of the program is working fine.

In smoke testing, the test cases chosen cover the most important functionality or component of the system. The objective is not to perform exhaustive testing, but to verify that the critical functionalities of the system are working fine.

For example, A typical smoke test would be -

Verify the application launches correctly,
check the GUI is responsive, etc.

(Read the example from Textbook) —

~~Sanity Testing :-~~

Sanity testing is a surface level testing where QA engineer verifies that all the menus, functions, commands available in the product and project are working fine.

The goal of sanity testing is to determine that the proposed functionality works roughly as expected. If sanity test fails, the build is rejected to save time and cost involved in running a more rigorous testing suite.

Introducing Sanity testing is performed when development team needs to know quick state of the product after they have done changes in code.

~~-~~ ~~sanity testing is not rigorous~~

~~(Read features and examples from Textbook)~~

~~print~~ ~~Smoke Testing instead of Sanity Testing~~

- | | |
|--|---|
| 1) It is performed to confirm that the critical functionalities of the program are working fine. | Sanity testing is done to check the new functionality and bugs have been fixed. |
| 2) The objective is to verify the stability of the system. | The objective is to verify the functionality of the system. |
| 3) This testing is performed by developers or testers. | This test is usually performed by testers. |
| 4) It is usually undocumented or scripted. | It is not usually documented or scripted and is unscripted. |
| 5) It exercises entire system from end to end. | It exercises only the particular component of the entire system. |

~~Introducing~~

~~Regression Testing :-~~

It is defined as a type of software testing to confirm that a recent program or code change has not adversely affected the existing features. Regression testing is nothing but a full or partial selection of already executed test cases which are re-executed to ensure that existing functionalities work fine.

Entry criteria for regression testing :-

- * The defect is repeatable and has been properly documented
- * A change control or defect tracking record was opened to identify and track the regression testing effort.
- * A regression test specific to the defect has been created, reviewed and accepted.

Exit criteria for regression testing :-

- * Results of the test show no negative impact allowed to the application.

Regression Testing Techniques :-

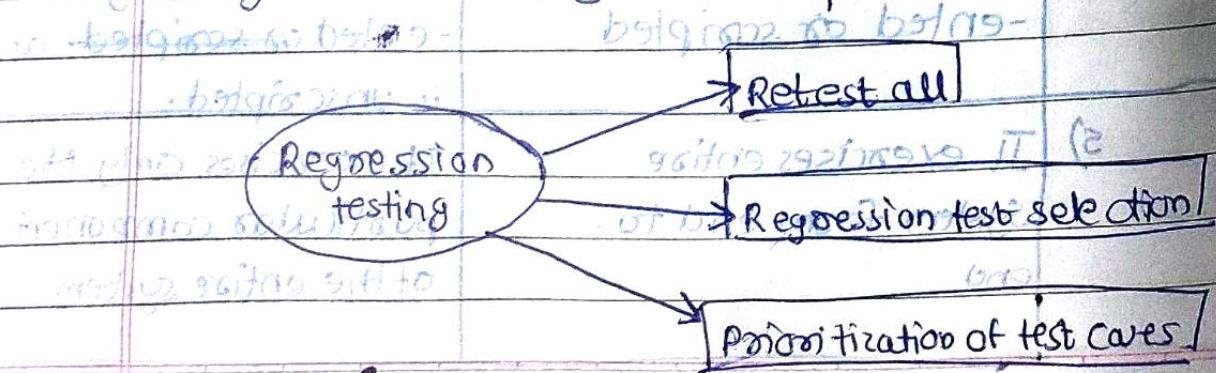


Fig :- Regression testing techniques

I] Retest all (TDD) ~~functionalities~~ identified

In this method all the tests in the existing test bucket or suite should be executed. This is very expensive as it requires huge time and resources. It is time consuming and cost based.

II] Regression test selection :-

Instead of re-executing the entire test suite, it is better to select part of test suite to be run.

Test cases selected can be categorised as :-

* Reusable test cases can be used in succeeding regression cycle.

* Obsolete test cases can't be used in succeeding regression cycle.

III] Prioritization of Test cases

Prioritize the test cases depending on business impact, critical & frequently used functionalities.

Regression Testing Tools

Following are the most important tools used for both functional and regression testing.

QUnit, Selenium, JUnit, TestNG, etc.

QUnit is a unit testing framework for web applications.

Selenium is a tool for automating web browsers.

TestNG

~~Notes~~

I] Quick Test Professional (QTP) :-

HP Quick Test Professional is automated software designed to automate functional and regression test cases. It uses VBScript language for automation. It is a data driven, keyword based tool.

II] Rational Functional Tester (RFT) :-

IBM's Rational Functional Tester is a Java tool used to automate the test cases of software applications. This is used for automating regression test cases and it also integrates with Rational Test Manager.

III] Selenium :-

This is an open source tool used for automating web applications. It can be used for browser based regression testing.

GUI Testing :-

GUI testing is a process to test application's user interface and to detect if application is functionally correct. It involves carrying set of tasks and comparing the result of the same with the expected output and ability to repeat same set of tasks multiple times with different data input and same level of accuracy.

~~Approach of GUI Testing :-~~

- a) Manual Based Testing :- Under this approach, graphical screens are checked manually by testers in conformance to the requirements stated in business requirement document. It is often error-prone and there are chances of most of the test scenarios to be left out. [done & significant]

- b) Automated Based Testing :- GUI testing can be done using automation tools. This is done in two parts i.e. Record Test and Playback Test. During Record, test steps are captured into the automation tool. During Playback, the recorded steps are executed on the application under test.
Ex of tools: QTP [done & significant]

~~Object Oriented Application Testing:~~

~~In object oriented sys~~
Object Oriented Testing is a collection of testing techniques to verify and validate an object oriented software.

Q3

OOT problems (what are the OOT problems)

① The understanding problem:- The complex relationships in OOT programs made it difficult to understand.

② Dependency problem:- The OOT programs are highly dependent on other features. This gives rise to dependency problem.

[This topic is not IMP for exam]

[so read it just from T.B]

2d 102 part 1 UN - part 291 b9208 bottom left

grobizint - about software engineering

>body> The testing levels :-

① The Algorithm Level

② The class level, based on

③ The Cluster level, algorithm

④ The System level

Class and inheritance testing

[This topic is not imp for exam]

(part 2) unit testing [Read it from T.B]

2d 102 b9208 bottom right

Unit Testing in OOT (unit testing)

In object oriented testing, unit testing

is also called individual classes testing. It is an activity that is performed in early development stages in order to detect errors in software and to gain confidence in the correctness of the software if no errors are found.

In the context of object orientation, a particular class, which is part of the application to be developed is considered to be a single unit for testing with boundaries between self and others.

Usually the test of a single class involves the usage of other classes too. For instance, classes that appear in the signatures of the class under test (CUT) are required for the test of this class.

(Read examples from T.B.)

Integration testing in OOT :-

In object oriented testing, Integration testing is also called inter-class testing. The major testing focus here is their interfaces, integrated functions, and integrated behaviors.

When unit testing is completed then only we can perform integration testing. It's biggest problem is to determine how long to spend to test this phase as it takes almost the whole testing phase.

The most popular integration testing strategies:

- * 1] Top/down
- * 2] Bottom up
- * 3] Execution based integration test
- * 4] Value based integration test
- * 5] Function based integration test

desc of next page

From last page desc.
↓

Page No. 52

- 3] → Tracing the execution of an interaction.
- 4] → Executes the interaction between components with certain values.
- 5] → Tests the correct provision of functionality through the component's collaboration.

State based testing

A state based testing of system can be modelled as a state machine or a finite state machine or state transition diagram to represent the change of states of the program under test. In finite state machine, there are only two key elements: state & transition.

* State - corresponds to certain value of the certain attributes of program

* Transition - corresponds to methods

which belongs to particular module

Any module or program starts with a start state

and exit at the end state.

and it starts from start state and ends at end state.

— (Read in detail from IIT-B) —

Initial point moving from SIT

Steps for state based testing followed by tester:

go motto [S ★]

* Create a state transition diagram for each relevant logic with logical states [L1 L2 L3]

* Transform the state transition diagrams into a transition trees, whereby root of the tree is the start state.

* Create a state based test for the class under test, whereby each state corresponds to a branch of the transition tree. The test case then sends the object under test a series of test stimuli that are intended to walk the object under test down the corresponding branch of the transition tree.

Limitations of State based techniques:-

* It requires correct state models of relevant classes in order to develop the test cases.

* State based testing requires complete state models. It is not adequate to use partial state model of a class without including inheritance.

The uniform modelling language (UML) gives a standard way to create a system's structure, database schema, covering business process, system classes and reusable components.

The modelling diagrams used in UML are :-

1] Use case diagrams :- It captures requirements of the system.

2] Class diagrams :- It depicts a static view of the model.

3] Object diagrams :- It uses a subset of the elements of a class in order to emphasize the relationship between instances of classes.

- 4] Sequence diagrams:- It is a form of interaction diagram which shows object as lifelines running down the page.
- 5] Collaboration diagrams:- It describes interaction between objects among classes and associations.
- 6] State machine diagrams:- It models the behaviour of a single object, specifying the sequence of events that object goes through during its lifetime in response to events.
- 7] Activity diagrams:- It is used to display sequence of activities of a system.
- 8] Component diagrams:- It illustrates the pieces (units) of software, embedded controllers, etc. which will make up the system.
- 9] Deployment diagrams:- It provides the model of run-time architecture of the system.
- ~~System Testing~~: It is the last phase and comes just before the product is delivered to the customer. System testing ensures that the program matches the final specification that was drawn at the beginning of the project.
- This phase is not concerned how the system works rather than it is more concerned with the results produced.
- System Testing should take place in the setup which accurately reacts like a system in which the product will be deployed.

~~Client Server Testing~~

~~Client server computing is a style of computing involving multiple processors, one of which is typically a workstation and across which a single business transaction is completed. These are tiered architectures. In these architectures there is distribution of presentation services, application code, data across clients and servers. A networking protocol is used for client-server communication.~~

~~Two main types of architecture~~

~~* 2-tiered architecture:- It is also known as client centric architecture. Nearly all processing happens at the client, and client accesses the database directly without using any middleware.~~

~~(缺点) In this architecture, direct access to the database makes it simpler to verify the test results. The disadvantage is the limit of the scalability and difficulty for maintenance.~~

~~* 3-tiered architecture:- In this application is divided into a presentation tier, a middle tier and a data tier. The middle tier is composed of one or more application servers distributed across one or more physical machines.~~

The following are some specific considerations needing to be addressed in a client/server testing plan:

- i) Must include consideration of different hardware and software platforms on which the system will be used.
- ii) Must take into account network and database server performance with which mainframe systems does not have to deal.
- iii) Has to consider the replication of data and processes across networked servers.
- iv) Client/server interface and client's interface with servers.
- v) Server functionality.
- vi) Reliability and performance of the network.

~~Client/Server testing in different layers:-~~

- a) Testing on the client side (GUI testing)
 - i) The complexity of GUI testing is due to:
 - i) Cross platform nature
 - ii) Event driven nature
 - iii) Using the mouse, as an alternate method of input, also raises some problems. It is necessary to assure that the application handles both mouse input and keyboard input correctly.
 - iv) Many traditional testing techniques can be utilised here such as Review techniques, Data validation techniques, scenario testing, decision logics, structured capture/playback.

b) Testing on Server side (Application testing) (iv)

(i) Client/Server/Loading tests:

It is necessary in order to determine the suitability of application servers, database servers and web servers performance. For these tests, the common objectives include:

- * Measuring the length of time to complete an entire task.

- * Describing which hardware/software configuration gives optimal performance.

- * Tuning database queries for optimal response.

- * Capturing Mean time to Failure as a measure of reliability.

- * Measuring system capacity to handle loads etc...

(ii) Volume testing:

Its purpose is to find weaknesses in the system with respect to its handling of large amount of data during extended time periods.

(iii) Stress testing:

Its purpose is to find defects of the system capacity of handling large numbers of transactions during peak periods.

(iv) Performance testing:

It is generally assessed in terms of response time and throughput under differing processing and configuration conditions.

(v)

Data testing: It involves activities like data recovery testing, data backup and restoring testing, Data security tests, Replicated data integrity testing.

c) Networked application testing

(i) Application response time: It is necessary to measure application response time while application is completing series of tasks.

(ii) Application functionality: It involves testing shared functionality across workstations, shared data and shared processes.

(iii) Configuration and sizing: It measures the response of specific system configurations.

(iv) Stress testing: It is to overload network resources such as routers or hubs.

(v) Performance testing: It can be used to determine how many network devices will be required to meet network's performance requirements.

(vi) Reliability testing: It involves running the network for 24-72 hrs under a medium to heavy load.

It is important that the network remain functional in the event of node failure.

d) Security Testing:

For internet based client server systems, security testing for webserver is important. The webserver is your LAN's window to the world.

Some risks are mentioned below that focuses on web server security testing.

- * The primary risk is errors in web servers side wrong configuration that would allow remote users to steal confidential information or executing commands on server host.
- * The secondary risk occurs on the browser side active content that crashes the browser, thereby damages your system and breaches privacy.
- * The third risk is data interception during copying data from your browser.

~~Web based testing :- test to run and~~

Web pages contain text, pictures, sounds, video and hyperlinks.

Following components should be considered while testing websites.

- * Check links:- check all outgoing links from all pages. Test all web pages inter links.

- * Check all forms of web pages:- check validation of web page fields.

- * Validate HTML:- for optimization purpose, it is most important.

- * Check internal text for typographic errors.

- * Check with other browsers.

Black box testing for website

The easiest way to start is by considering the webpage/website as a black box. We don't have any idea how it works, what is its specification. We just have the website in front of us.

① Text:

- * When testing the text, consider the audience, the terminology, the content and subject matter, the accuracy and spelling.
- * You should check the title for each page.
- * Also check the overlooked type of text.

② Hyperlinks:

- * Check to make sure that hyperlink jumps to correct destination and opens the correct window.

③ Graphics:

- * Test all graphics load and display properly on webpage.
- * Also test if graphics can be loaded properly on slow internet connection or not.
- * Make sure that there should not be too many graphics on the webpage.

④ Forms:

- * Check all components on the form, with respect to the functionality.

Page No.	61
Date	

Gray box testing:

With gray box testing, you straddle the line between white and black box testing. Webpages lend themselves nicely to gray box testing.

Using gray box testing test the web pages and HTML code of web pages. HTML is not a programming language, it is a markup language.

White box testing:

These are two types of web pages, static web page and dynamic web page. To perform white box testing, you don't necessarily need to become expert in these languages; only thing you need to know is able to read, understand the code, from test case embedded in what you see in the code.

Following are the several approaches effectively tested with white box approach:

- * Dynamic Content
- * Database driven web pages
- * Programmatically created web pages
- * Server performance and loading
- * Security.

Approaches followed in following

Approaches followed in following

Approaches followed in following

Approaches followed in following

Approaches followed in following