24/01/23

Unit 1.

Evolution of Software Testing

In the early days of software development, software testing was considered only a debugging process for removing errors after the development of software.

We can divide the evolution of software testing into the following phases.

1) Debugging Oriented Phase

This phase is early period of testing at the time testing is unknown program basics are written and then tested by the programmer until they are sure that all the bugs were removed. The term used for testing was check out on getting the system to run.

2) Demonstration Oriented Phase

Debugging continued in this phase. In 1957 the purpose of check out is realised that it is not only to run the software but also to demonstrate the correctness according to mention requirements thus the scope of check out to program increased to program runs of program correctness

3) Destruction oriented phase

Here change the view of testing from "testing to show the absence of errors" to "testing is to find more and more errors". The importance of early testing was realised in this phase.

4) Evolution Oriented phase

This phase stresses on the quality of software product such that it can be evaluated at every stage of development. If the bugs were identified at any early stage of development it was cheeper to debug them as compared to bugs found in implementation or post implementation phases.

5) Prevention phase

The evolution model stressed on the concept of bug prevention as compared to earlier concept of bug detection with idea of early detection of bugs in earlier phases so that we can prevent the bugs in the implementation on further phases.

6) Process Oriented phase

In this phase testing was established as a complete process rather than a single phase (performed after coding) in the software development life cycle (SDLC) The testing process start as soon as the requirements for the project are specify and it runs parallel to SDLC

★ Software Testing 1.0

Software testing was just considered a single phase to be performed after coding of software in SDLC No test organisation was present a few testing tools were present but there was limited due to high cost. There was no quality goal

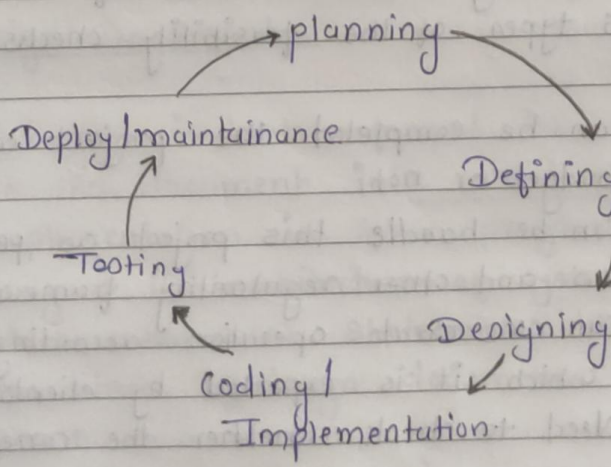
★ Software Testing 2.0

In this phase software testing gained importance in SDLC and the concept of early testing also started. Testing was involving in the direction of planning the resources many testing tools were also available in this phase.

★ Software Testing 3.0

In this phase software testing is being revolved in the form of a phases which is based on the strategic effort. This means that their should be a processes which give us a road map of overall testing process management is actively involved in this phase. It should be driven by quality goals.

★



SDLC cycle

31/01/23

Vidyalekhan
DATE
PAGE

SDLC stand for Software development life cycle. It describes the sequence of phases or steps to develop any software.

In simple words "Entire life time of software from beginning to ending."

1) Planning

→ It involves analysis of the problem and it is most important at fundamental stage.

→ If a software company any customer comes with the problem and it is analyzed that it is feasible or not to solve the customers problem.

→ Here estimation of rough idea of the resource requirements as well as the estimated time for completion is given.

→ It is performed by Senior members of the team. There are mainly 5 types of the feasibility checks.

i) Economic - Can be completed the project within the budget or not

ii) Legal - Can be handle this project as per the cyber law and other regularity frameworks

iii) Operation feasibility - Can we create operation according to which it is expected by clients?

iv) Technical - Need to check whether the current computer system can support the software.

v) Schedule - Decide that the project can be completed within the given schedule or not.

2) Defining / Analysis

→ The aim is to understand the exact requirements of the customer. It consists of two distinct activities

i) Requirement Gathering and Analysis.

⇒ The goal is to collect all the relevant information from customer regarding the product to be developed. So from (customer regarding) that the incompleteness and inconsistancy are removed. (ignore)

ii) Requirement Specification and documentation.

⇒ After all the ambiguities, inconsistancies and incompleteness have been resolved and all the requirements are properly understood. The requirement specification can start. RECS is very important activity and it is systematically organised into a software requirements specification (SRS).

3) Design Document

→ The goal is to transformed the requirement specified in the SRS document into structure i.e. suitable for implementation in some programming languages. We designed the architecture of system using SRS document which means we skeleton a blue print of the project two kinds of design documents are developed in this phase

i) HLD (High Level Design) - It's a complete architecture diagram

ii) LLD (Low Level Design) - Complete details and leasting of error messages are given

4) Coding

→ Developer start building the entire system by coding code using the chosen programming language the system first is developed in small programs called as units / models.

→ In coding phase the tasks are divided into units / modules and are assigned to various developers.

→ It is the longest phase of the SDLC process

→ Developers of all levels including seniors / juniors / freshers are involved in this phase.

5) Testing

→ It is important part of project as it helps to improved the quality of the project. After the development of the software they can be many defects in the system or may not be working as they are expected

6) Software Maintainance

→ Is the modification of the software product after the delivery to correct defaults, to improve the performance all other attributes or to adapt to the product to the modified environment. It is the backbone of software success.

1/2/2023

Vidyalekhan™
DATE
PAGE

★ Verification & Validation

Verification	Validation
1) It includes checking documents, design, codes and programs.	1) It includes testing and validating the actual product
2) Verification is the static testing	2) Validation is the dynamic testing
3) It does not include the execution of the code.	3) It includes the execution of the code.
4) Methods used in verification are reviews, walkthroughs, inspections and Desk checking	4) Method in validation are black box testing, white box testing and non-functional testing
5) It checks whether the software confirms to specification or not.	5) It checks whether the software meets the requirements and expectations of a customer or not.
6) It can find bugs in the early stage of the development	6) It can only find the bugs that could not be found by the verification process.
7) The goal of the verification is application and software architecture & specification.	7) The goal of validation is the actual product
8) Quality assurance team does verification.	8) Validation is executed on software code with the help of testing team

9) It comes before validation.	9) It comes after verification.
10) It consist of checking of documents / files and is performed by human.	10) It consist of execution of program and is performed by computers.

3/02/23

★ SW Testing Process.

Step 1 :- Asseas development plan and status.

This initiative may be prerequisite to put together verification, validation and testing plan wanting to evaluate implemented Software solution. during the step tester challenge completeness and conectness of the event plan based on the extensiveness and completeness of project plan tester can estimate quantity of resources they are going to go to test implemented software solution

Step 2 :- Develop the test plan

Forming plan for testing will follow an equivalent pattern as any Software planning process. The structure of all the plans should be an equivalent but content will vary supported degree of risk testers perceive as related to software being developed

Step3 :- Test Software Requirements

Incomplete, Inaccurate or Inconsistant requirements cause most of the S/w failures. The inability to get requirements right during requirement gathering phase can also increase cost of implimentation significantly. Testers through verification must determine that requirement are accurate, complete and they do not conflict with another.

Step 4 :- Test the S/w Design

This step test both external and internal design primarily through verification. The testers are concerned that planning will acheive objectives of requirements, also because design effective and efficient all designated hardware

Step 5 :- Build phase Testing

The method chosen to build the software from internal design document will determine type and extensiveness of testers needed as the construction becomes more automated, less testing are going to be required during this phase. However, if the software is made using waterfall process its subjective error and failures will be verified it Experience has shown that its significantly cheeper to spot defects during development phase rather than through dynamic testing during test execution step.

Step 6 :- Execute and Record Result

This involves testing of code during dynamic state. This approach, methods and tools layed out in test plan. and are going to be validated that are executable code that meets actually stated s/w requirements and therefore the structural ~~design~~ specification of design.

Step 7 :- Acceptance Test

Acceptance testing unables users to take applicability and usefullness of s/w in performing their day-to-day job functions. It is test what the user believes s/w should perform as against what documented requirements state s/w should perform

Step 8 :- Test Reporting

Test reporting is continuous process. It may be both oral and written. It is important defects and concerns be reported to the appropriate parties as ~~power to~~ possible, so that corrections can be made at the lowest possible cost.

Step 9 :- The Software Installation

Once test team has confirmed that Software is prepared for production use power to execute that software during production environment should be tested. This test interface to operating Software

related software and operating procedures

Step 10 :- Test Software changes.

While this is often shown as step 10, within context of performing maintainance often software is implemented, concept is additionally applicable to changes throughout implementation process. whenever requirements & changes test plan must change; and unpait of it that change on slw system must be tested and evaluated.

Step 11 :- Evaluate Test Effectiveness

Test improvement can best be acheived by evaluating effectiveness of testing at top of every software test argument, while this assesment is primarily performed by testers, it should involve developers, uses of software and quality assurance professionals of function exits units the IT organisation.

7/02/23

★ Terminologies in Testing.

1) Error

The problem in code leads to errors, which means that a mistake can occur due to developers coding error as the developer misunderstood the requirements or the requirement was not define correctly. The developers used the terms errors.

2) Fault

The fault may occur in software because it has not added the code for fault tolerance, making an application act up. Following reasons

1) Lack of resources
2) An invalid step.
3) Inappropriate data defination can lead to a fault to occur in a program.

3) Failure

Many defects lead to the s/w failures which means that a loss a specifies a fatal issue in s/w or in its module., which makes the system unresponsive or broken.

In other words, we can say if the end user detects an issue in the product than that particular issue is called as failure.

eg In a bank application if the amount transfer module is not working for end users when the end user tries to tranfer money, Submit button is not working. Hence this is failure

4) Verification

Verification is the process of checking that a s/w acheive its goal without any bugs. It is the process to ensure whether the product that is developed is right or not. It verifies whether the developed product fulfills the requirements that we

have. Verification is static testing.
Verification means "are we the building the product right?"

5) Validation

Is a process of checking whether the slw product is upto the mark or in other words product has high level requirements. It checks what we are developing is the right product. It is the validation of actual & expected product. Validation is dynamic testing.
Validation means are we building the right product?

8/02/23

★ Test Cases

Test case is a set of action executed to verify a perticular feature or functionality of your software application. A test case contains test steps, test data, precondition, postcondition developed for specific test scenario to verify any requirement.
The test case includes specific variables or conditions using which a testing engineer can compare expected & actual results to determine whether a software product is functioning as per the requirement of the customer.
For a test scenario checks login functionality there are many possible test cases

Test case 1 :- checks results on entering valid ID and password.

Test case 2 :- check results on entering invalid user ID and password.

Test case 3 :- check response when user ID is empty and login button is pressed, and many more.

The format of a standard test case

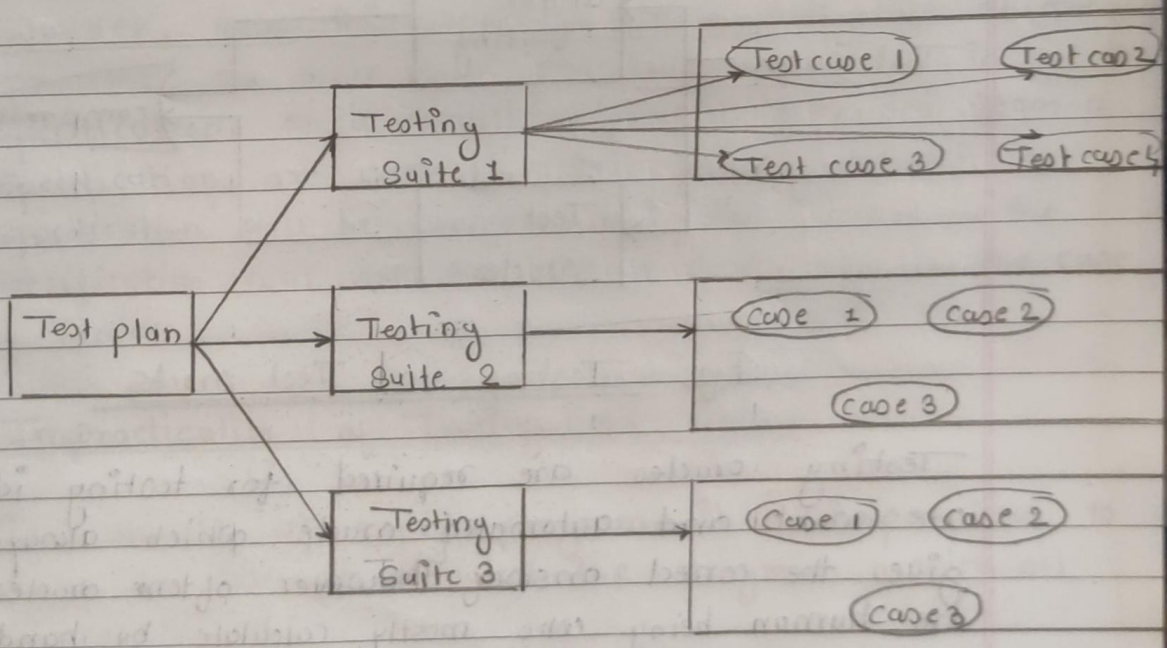
Test case ID	Test case Description	Test steps	Test Data	Expected Results	Actual Results	Pass/ Fail
TU01	Check the customer login with valid data	1. Go to the site http://demoguru.com 2. Enter User ID 3. Enter password 4. Click Submit	User ID = guru99 Password = Pass99	User should login in to an application	As expected	Pass
TU02	check the customer login with Invalid data	〃	User ID = guru991 Password: Pass99	No login	As expected	Pass
TU03	check the customer login with Empty ID	〃	User ID [Empty] Password Pass99	No login	As expected	Pass

This entire table may be created in word, excel, or any test management tool.

★ Testing Suite.

Test suite is a container that has a set of test cases which helps the testers in execution and reporting the test execution status. It can take any of the 3 steps namely active, Inprogress and completed.

A test case can be added to multiple test suites and test plans after creating a test plan test suits are created based on the cycle or based on the scope. It can contain any type of test which functional or non-functional



Test Suite Diagram

10/02/23

Vidyalekhan
DATE
PAGE

★ Test Oracles

Test oracle is a mechanism diff^n from program itself that can be used to test the accuracy of a programs output for test cases conceptually we can consider a testing a process in which test cases are given for testing & the program under test. The output of the two then compared to determine whether the program behaved correctly for the test cases. This is shown in the diagram below.

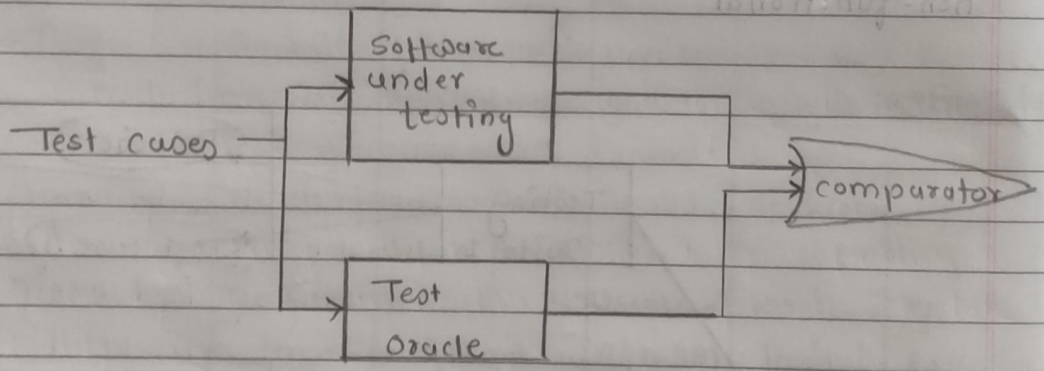


Fig Testing and Test oracle

Testing oracles are required for testing ideally we want ~~and~~ automated oracle which always gives the correct answer however often oracles are human being who mostly calculate by hand what the output of the program should be as it is oftenvery difficult to determine whether the behaviour corresponds to the ~~exact~~ expected behaviour, are "human dieties" may make mistakes

Consequently, when there is a disturbancy betn the ~~proj~~ program and result we must verify the result produced by the oracle before declaring that there is a defect in the result.

The human oracles typically use the programs specifications to decide what the correct behaviour of the program should be to help oracle determine the correct behaviour it is important that the behaviour of the system or the component is explicitely specified and the specification itself be error free.

There are some systems where oracles are automatically generated ~~whic~~ with such oracles we are assured that the output of the oracle confirms to specification. However, even this approach does not solve all are problems as their is a possibility of errors in specification. as a result a device generated from a specification are correct, the result of the specification will be correct and the result if the specification will not be correct / reliable in the case of errors.

- **Impracticality of Testing all Data.**

1. For almost ~~all~~ every program It is impossible to make an attempt to test the program with all sets of inputs.

2. The correctness of the output for a perticular test input is determined using a testing oracle.

3. A testing oracle is a mechanism that can be used

for determining whether a test has passed / failed.

- **Impracticality of Testing all Paths :**

1. For almost every program it is impossible to attempt to test all executable paths through the project because of the number of combinations i.e. "Combination Explosion."

2. Developing an algorithm for this purpose is also not possible.

15/02/23

★ **Introduction : Purpose, Productivity and quality in S/w introduction**

An engineering displine, almost by defination is driven by a practical parameters of cost, schedule & quality.

A solution that takes anomous resolven and a many years may not be acceptable, similarly a poor quality solution even at low cost may not be of much use like all engineer decipline s/w engineering is driven by the three major factors

a) Cost
b) Schedule
c) Quality

1) Cost -
The cost of developing assistant, if the cost of resources used for the system which in the case of s/w is dominated by manpower cost as development is largely labour intensive. Hence, the cost of s/w project is often major in the terms of person-month i.e. the cost is consider to be the total no. of person-month spent in the month period on the project.

2) Schedule -
Schedule is an important factor in many projects. Buisness trends are decdating that the time to market of a product should be reduced. that is cycle time from concept to delivery should be small for s/w these means that is need to be develop factor

3) Productivity -
In the terms of output (KLOC) per person month can adquately capture both cost and scheduler concent if productivity is higher it should be clear that the cost in the terms of person month will be lower. (The same work can now be done will fewer person months)
Similarly, if the productivity is higher the potential of developing the s/w in shorter time improved time of higher productivity will finish a job in easser time then a same size time with lower productivity.
In other words, productivity is the key in all buisness & a desired a high quality productivity.
The other measure factor driving any production discipline is quality.

According to the quality model adopted by this the standard compressed of 6 main attribute.

i) Functionality
→ The capability to provide function which meets the statedled the improve needs coher s/w is used

ii) Reliability
→ The capability to maintain a specified level of performance.

iii) ~~Usability~~ Efficiency
→ The capability to provide performance reliable to the amount of resources used

iv) Usability
→ The capability to be understood, learned and used

v) Maintainability
→ The capability to be modified for the purpose of making corrections, improvements or adaption

vi) Portability
→ The capability to be adapted for different specified environment without applying actions or means other than those provided for the purpose in the product.

20/02/23

Vidyalekhan

DATE
PAGE

★ Testing Vs Debugging

Testing	Debugging
1) Testing is the process to find bugs and errors	1) Debugging is the process of correcting the bugs found during testing.
2) It is the process to identify the failure of implemented code	2) It is the process to give absolution to code failure
3) Testing is the display of errors.	3) Debugging is a deductive process
4) Testing is done by the tester.	4) Debugging is done by either programmer or the developer
5) There is no need of design knowledge in the testing process	5) Debugging can't be done without proper design knowledge
6) Testing can be done by insiders as well as outsiders	6) Debugging ~~can't~~ is be done ~~without~~ insiders. An outsiders can't be debugging
7) Testing can be manual or automated	7) Debugging is always manual Debugging can't be automated
8) It is based on the different testing levels i.e. unit testing, integration testing, System testing, etc.	8) Debugging is based on different types of bugs.
9) Testing is a stage of the s/w development life cycle(SDLC)	9) Debugging is not an aspect of s/w development life cycle it occurs as a consequence of testing
10) Testing is composed of validation & verification of s/w	10) while debugging process seeks to match symptoms with cause by that it leads to error correction

11) Testing is initiated after the code is written	11) Debugging commences with the execution of a test case
12) Testing process based on various levels of testing - system testing, integration testing, unit testing, etc	12) Debugging process based on various types of bugs is present in a system