Introduction to Software Testing Methods

STUDENTS-HUB.com
Uploaded By: anonymous

My Favorite Quote on Software Testing

The stuff we call "software" is not like anything that human society is used to thinking about. Software is something like a machine, and something like mathematics, and something like language, and something like thought, and art, and information.... but software is not in fact any of those other things. The protean quality of software is one of the great sources of its fascination. It also makes software very powerful, very subtle, very unpredictable, and very risky.

Some software is bad and buggy. Some is "robust," even "bulletproof." The best software is that which has been tested by thousands of users under thousands of different conditions, over years. It is then known as "stable." This does NOT mean that the software is now flawless, free of bugs. It generally means that there are plenty of bugs in it, but the bugs are well-identified and fairly well understood.

There is simply no way to assure that software is free of flaws. Though software is mathematical in nature, it cannot by "proven" like a mathematical theorem; software is more like language, with inherent ambiguities, with different definitions, different assumptions, different levels of meaning that can conflict.

STUDENTS-HUB.com

Uploaded By: anonymous

Why?

- **Software development involves:**
  - ambiguity
  - assumptions
  - and flawed human communication
- **Each change made to a piece of software, each new piece of functionality, each attempt to fix a defect, introduces the possibility of error**
- **Software testing reduces this risk**

Developers often overlook fundamental ambiguities in requirements in order to complete the project; or they fail to recognize them when they see them.

STUDENTS-HUB.com
Uploaded By: anonymous

Introduction

- The procedures that will be presented in this course can help developers and managers better understand the value of comprehensive software testing.
- Provide guidelines to achieve testing goals
- Software should be consistent and present no surprises to users.

Software testing is a process, or a series of processes, designed to make sure computer code does what it was designed to do and, conversely, that it does not do anything unintended

STUDENTS-HUB.com
Uploaded By: anonymous

Introduction..2

- a well-known rule of thumb that in a typical programming project approximately 27 **- 50%**of the **elapsed time** and more than **24 - 50%** of the **total cost** were expended in testing the program or system being developed.

- Even though new programming languages, tools, IDEs have been developed, but

"Software testing remains among the dark arts of software development", (Mayers, Sandler 2011)

STUDENTS-HUB.com

Uploaded By: anonymous

Developers vs Testers

- Designers and developers approach software with an optimism based on the assumption that the changes they make are the correct solution to a particular problem.
- The tester takes nothing at face value.
- The tester always asks the question “why?”
- They seek to drive out certainty where there is none.
- They seek to illuminate the darker part of the projects with the light of inquiry.

STUDENTS-HUB.com
Uploaded By: anonymous

Developers vs Testers Summary

- **Developers**
  - Act with the intention of finding solutions.
  - Assume that the solution they provided is correct.
- **Testers**
  - Approach with skepticism and ask questions.
  - Ensure that the solutions actually work and meet the intended purpose.
  - Thoroughly test every part of the system.
- **Goal**
  - The difference in roles ensures software quality.
  - Both developers and testers contribute to discovering and fixing potential issues, ensuring the success of the project.

STUDENTS-HUB.com
Uploaded By: anonymous

Test Cases and Test Scenarios

- **Test Case**: “A test case is a set of conditions or variables under which a tester will determine whether an application, software system or one of its features is working as it was originally established for it to do.”
- **Test Scenario:** “The exhaustive testing is not possible due to large number of data combinations and large number of possible paths in the software. Scenario testing is to make sure that end to end functionality of application under test is working as expected. Also check if the all business flows are working as expected.”
- **In summary:**
  - **Test Case = Detailed**, step-by-step test with specific expected results.
  - **Test Scenario = High-level**, general overview of what functionality to test.

STUDENTS-HUB.com
Uploaded By: anonymous

Test Case Elements

ITEM	DESCRIPTION
Title	A unique and descriptive title for the test case
Priority	The relative importance of the test case (critical, nice-to-have,etc.)
Status	For live systems, an indicator of the state of the test case. Typical states could include : Design – test case is still being designed Ready – test case is complete, ready to run Running – test case is being executed Pass – test case passed successfully Failed – test case failed Error – test case is in error and needs to be rewritten
Initial configuration	The state of the program before the actions in the "steps" are to be followed. All too often this is omitted and the reader must guess or intuit the correct pre-requisites for conducting the test case.
Software Configuration	The software configuration for which this test is valid. It could include the version and release of software-under-test as well as any relevant hardware or software platform details (e.g. WinXP vs Win95)
Steps	An ordered series of steps to conduct during the test case, these must be detailed and specific. How detailed depends on the level of scripting required and the experience of the tester involved.
Expected behaviour	What was expected of the software, upon completion of the steps? What is expected of the software. Allows the test case to be validated with out recourse to the tester who wrote it.

STUDENTS-HUB.com

Uploaded By: anonymous

Test Case Elements..2

- In addition to these basic elements, agile test cases may also include additional information such as:
  - **Priority**: The priority of the test case, which can be used to determine which test cases to execute first.
  - **Severity**: The severity of the defect that the test case is intended to detect.
  - **Owner**: The person responsible for executing the test case.
  - **Notes**: Any additional information about the test case, such as known issues or workarounds.

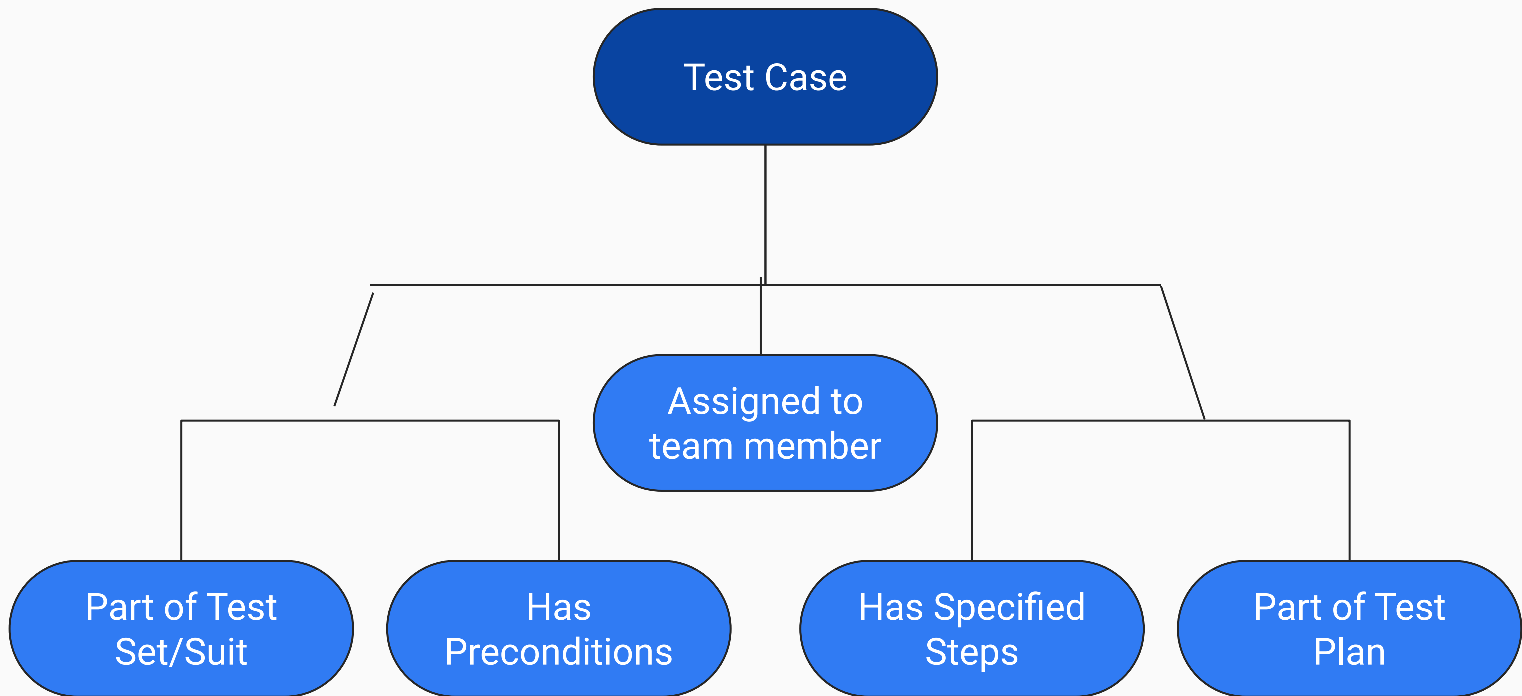
STUDENTS-HUB.com
Uploaded By: anonymous

Example Test Case

Test case ID: TC001 Test case description: Verify that a user can log in to the system.	
Pre-conditions:	* The user has a valid user account. * The system is up and running.
Test steps:	1. Go to the login page. 2. Enter the user's username and password. 3. Click the "Login" button.
Test data:	* Username: testuser * Password: password123
Expected result:	* The user is logged in to the system and redirected to the home page.
Actual result:	* The user is logged in to the system and redirected to the home page.
Status	Pass

STUDENTS-HUB.com
Uploaded By: anonymous

Test Cases..cont

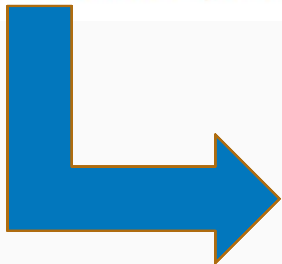


STUDENTS-HUB.com
Uploaded By: anonymous

Let’s start by self assessment exercise

Suppose that we have a program that we want to write test cases for:

The program reads three integer values from an input dialog. The three values represent the lengths of the sides of a triangle. The program displays a message that states whether the triangle is scalene, isosceles, or equilateral.



Scalene: no two sides are equal
Isosceles: has two equal sides
Equilateral: all three sides are equal

STUDENTS-HUB.com
Uploaded By: anonymous

Suggested Test Cases

1. Do you have a test case that represents a *valid* scalene triangle? (Note that test cases such as 1, 2, 3 and 2, 5, 10 do not warrant a yes answer because a triangle having these dimensions is not valid.)
2. Do you have a test case that represents a valid equilateral triangle?
3. Do you have a test case that represents a valid isosceles triangle? (Note that a test case representing 2, 2, 4 would not count because it is not a valid triangle.)
4. Do you have at least three test cases that represent valid isosceles triangles such that you have tried all three permutations of two equal sides (such as, 3, 3, 4; 3, 4, 3; and 4, 3, 3)?
5. Do you have a test case in which one side has a zero value?
6. Do you have a test case in which one side has a negative value?
7. Do you have a test case with three integers greater than zero such that the sum of two of the numbers is equal to the third? (That is, if the program said that 1, 2, 3 represents a scalene triangle, it would contain a bug.)

STUDENTS-HUB.com Uploaded By: anonymous

Suggested Test Cases..2

8. Do you have at least three test cases in category 7 such that you have tried all three permutations where the length of one side is equal to the sum of the lengths of the other two sides (e.g., 1, 2, 3; 1, 3, 2; and 3, 1, 2)?
9. Do you have a test case with three integers greater than zero such that the sum of two of the numbers is less than the third (such as 1, 2, 4 or 12, 15, 30)?
10. Do you have at least three test cases in category 9 such that you have tried all three permutations (e.g., 1, 2, 4; 1, 4, 2; and 4, 1, 2)?
11. Do you have a test case in which all sides are zero (0, 0, 0)?
12. Do you have at least one test case specifying noninteger values (such as 2.5, 3.5, 5.5)?
13. Do you have at least one test case specifying the wrong number of values (two rather than three integers, for example)?
14. For each test case did you specify the expected output from the program in addition to the input values?

STUDENTS-HUB.com Uploaded By: anonymous

Exercise Discussion

- The point of this exercise is to illustrate that the testing of even a trivial program is not an easy task.
- Given this is true, think of testing of 100K statement program of air traffic control system!
- A typical GYM desktop program has 15K statements
- A Payroll system can have much more than that

STUDENTS-HUB.com
Uploaded By: anonymous

Psychology and Economics of SW Testing

- It is true that SW testing is technical task, but also involves important considerations of:
  - **Economics and**
  - **Human psychology**
- It is true that we want to test every single permutation of a program.
- But in most cases, this is simply impossible

STUDENTS-HUB.com
Uploaded By: anonymous

Tester’s Psychology

- **According to research:**
  - **The tester needs a proper attitude/vision to successfully test a software product.**
  - **Tester’s attitude is more important than the process/tools applied.**

- One of the primary causes of poor application testing is the fact that most programmers begin with a **false** definition of the term:
  - “Testing is the process of demonstrating that errors are not present.”
  - “The purpose of testing is to show that a program performs its intended functions correctly.”
  - “Testing is the process of establishing confidence that a program does what it is supposed to do.”

STUDENTS-HUB.com
Uploaded By: anonymous

Tester’s Psychology..2

- **These definitions are upside down, because**
  - When you test a program, you want to add some value to it.
  - Adding value through testing means raising the quality or reliability of the program.
  - Raising the reliability of the program means finding and removing errors.
  - Therefore, don’t test a program to show that it works; rather, start with the assumption that the program contains errors (a valid assumption for almost any program) and then test the program to find as many of the errors as possible

STUDENTS-HUB.com
Uploaded By: anonymous

Testing is the process of executing a program with the intent of finding errors.

Human beings tend to be highly goal-oriented, and establishing the proper goal has an important psychological effect on them. If our goal is to demonstrate that a program has no errors, then we will be steered subconsciously toward this goal; that is, we tend to select test data that have a low probability of causing the program to fail. On the other hand, if our goal is to demonstrate that a program has errors, our test data will have a higher probability of finding errors. The latter approach will add more value to the program than the former.

All text quoted from Myers, Sandler 2011

STUDENTS-HUB.com
Uploaded By: anonymous

Tester's Psychology..4

- We also need to clarify the term successful and unsuccessful:
- Most project managers refer to a test case that did not find an error a "successful test run," whereas a test that discovers a new error is usually called "unsuccessful."

A test case that finds a new error can hardly be considered unsuccessful; rather, it has proven to be a valuable investment. An unsuccessful test case is one that causes a program to produce the correct result without finding any errors.

STUDENTS-HUB.com
Uploaded By: anonymous

Economics of Testing

- After having agreed on definition for SW testing, we need to be clear that **it is not possible to find all errors in the program.**
- There are two prevalent testing strategies:
    - Black-Box Testing: also known as input driven and input/output testing, no knowledge about internal program structure.
    - White-Box Testing: testing is based on complete knowledge about internal code and its structure and conditions

STUDENTS-HUB.com
Uploaded By: anonymous

Verification and Validation

- **Verification** is the process of evaluating whether the software conforms to specifications and requirements during the development phase. It asks, "**Are we building the product right?**“
  - **Purpose**: Ensures that the product is being developed according to the design specifications and standards.
  - **Focus**: Internal processes such as code reviews, inspections, walkthroughs, and unit testing.
  - **Timing**: It happens during the development cycle, typically at every stage, like when you complete a module or a design phase.
  - **Method**: Static testing (e.g., code reviews, design reviews) where you do not execute the code.
  - **Output**: Documentation, reports, and artifacts to ensure that each stage of the process produces the correct outcome.

STUDENTS-HUB.com
Uploaded By: anonymous

Example of Verification

Example of Verification: Suppose you're working on a mobile app. During the design and coding stages, a code review checks whether the code adheres to design documents, coding standards, and requirements like proper input validation. Even if this part is perfect, the app might still not function as intended (e.g., users' needs might not be met).

STUDENTS-HUB.com

Uploaded By: anonymous

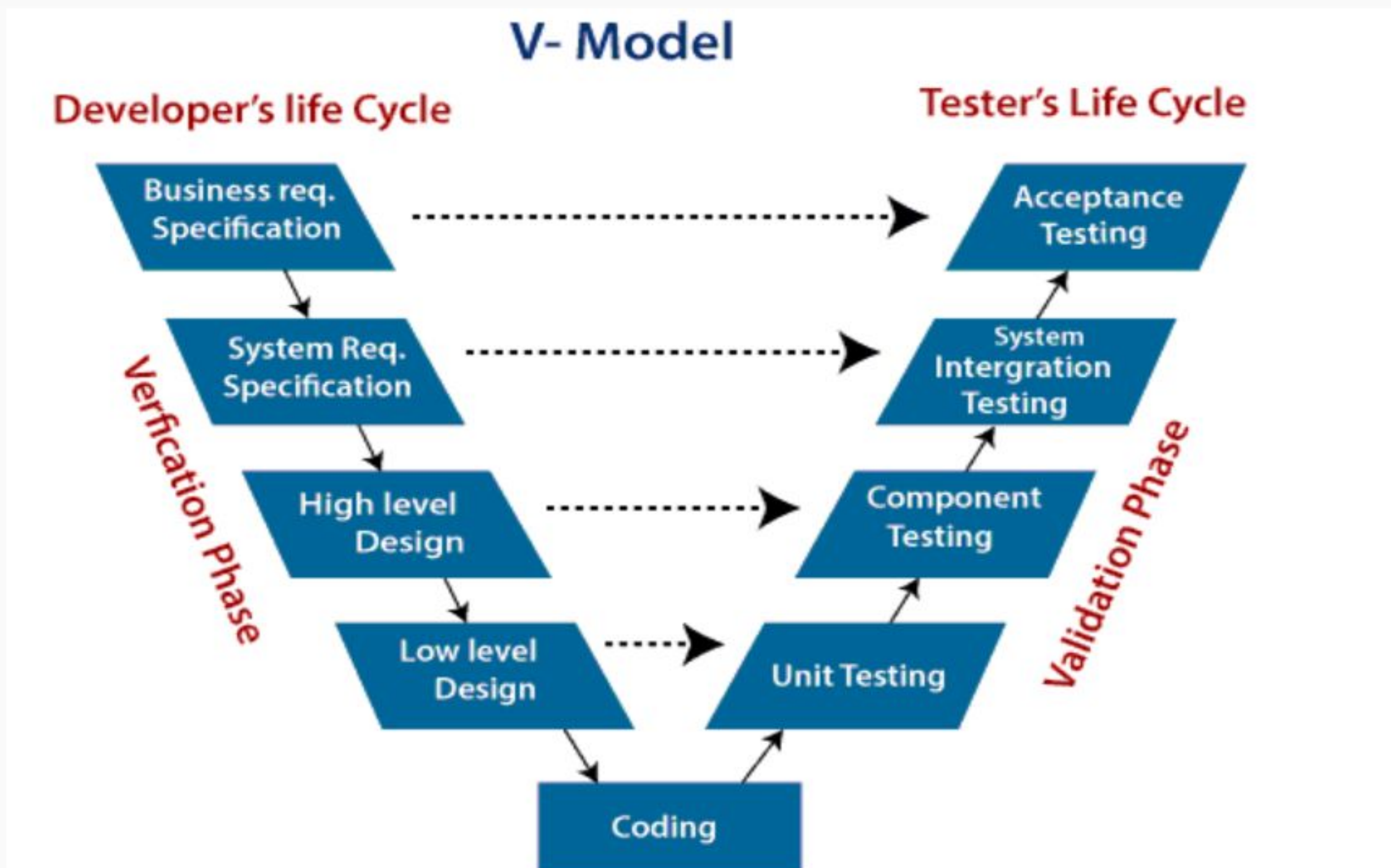
Verification and Validation

- **Validation** is the process of evaluating whether the final software product meets the business needs and expectations of the user. It answers the question, "**Are we building the right product?**"
  - **Purpose**: Ensures that the product actually meets the user's requirements and performs as expected in real-world conditions.
  - **Focus**: The external behavior and the functionality of the system.
  - **Timing**: Usually performed after the development process is complete or during the final testing phases.
  - **Method**: Dynamic testing (e.g., functional testing, integration testing, user acceptance testing) where the code is executed.
  - **Output**: A validated software product that users can accept and that functions correctly in their environment.

Example of Validation

Example of Validation: For the same mobile app, after the development is done, you might run User Acceptance Testing (UAT) with actual users. During UAT, you verify if the app's features align with what the users need. If users find that the app lacks an essential feature they expected, it has failed validation, even though the code is flawless.

V- Model In Software Development



Black-Box Testing

- Finding all errors using this strategy means running the program against all possible correct and incorrect input.

- This is totally **impossible**.

- **But, we can maximize the errors found using a finite number of test cases**

STUDENTS-HUB.com
Uploaded By: anonymous

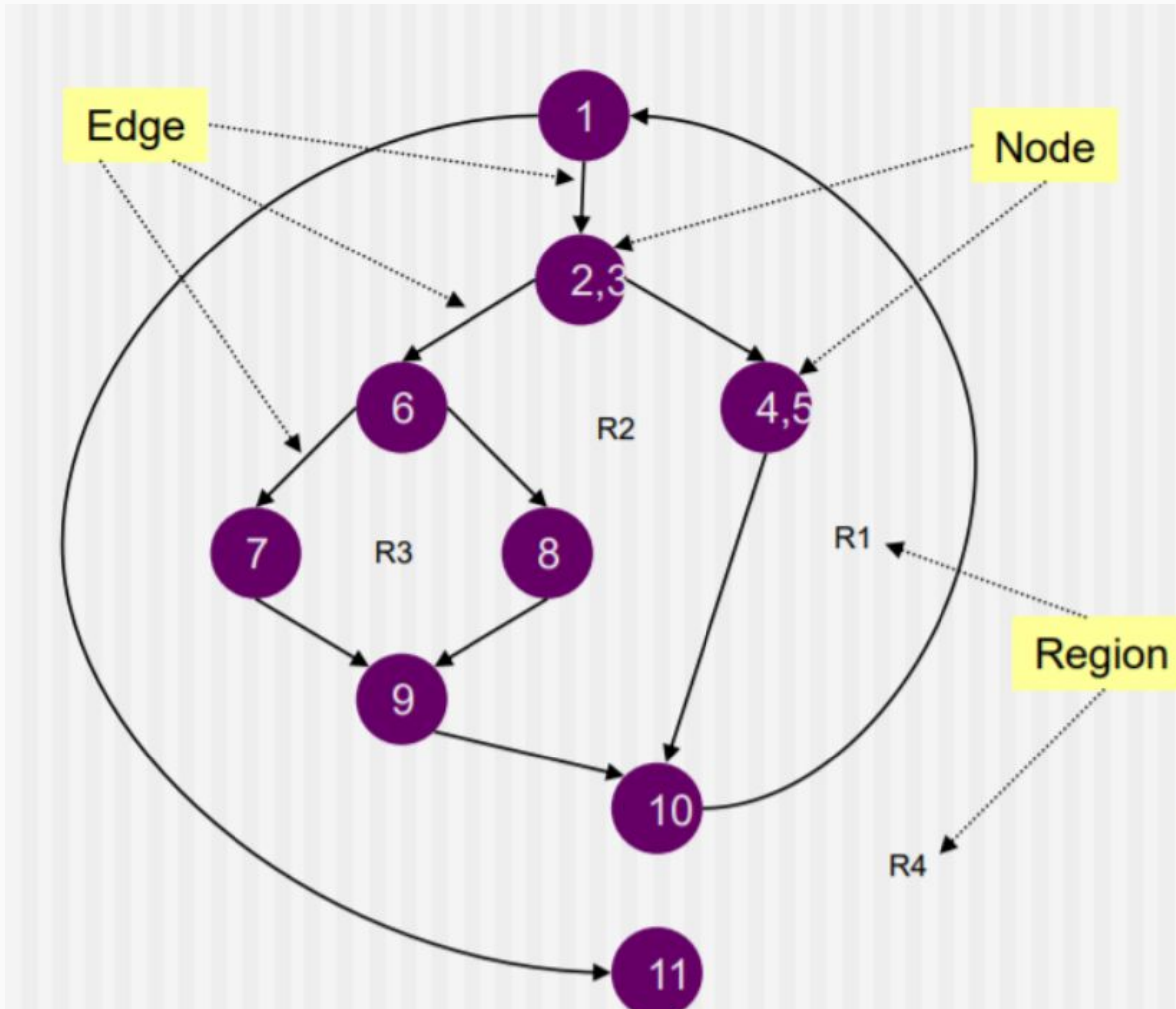
White-Box Testing

- Derives test data by examining the internal program code and **structure** (i.e, program logic)
- For instance, one tester may assume that causing every statement to execute in a program may be the correct answer. (i.e, exhastive path testing of a program)
- But this is totally untrue as well:
  - The number of unique logic paths in a program can be astronomically large!

STUDENTS-HUB.com

Uploaded By: anonymous

White-Box Testing..2



Number of unique paths = 100 trillions

Exhaustive path testing is not possible

STUDENTS-HUB.com
Uploaded By: anonymous

White-Box Testing..3

Example:

Let's take a simple if-else and loop program and create a control flow graph from it.

```
public void exampleMethod(int x) {
    int y = 0;
    if (x > 0) {      // Decision point: Node 2
        y = x + 1;    // Action: Node 3
    } else {          // Decision point: Node 4
        y = x - 1;    // Action: Node 5
    }
    for (int i = 0; i < x; i++) {   // Loop start: Node 6
        y += i;       // Action: Node 7
    }
    System.out.println(y);  // Node 8 (End point)
}
```


STUDENTS-HUB.com
Uploaded By: anonymous

White-Box Testing..4

Control Flow Graph (CFG):

1. Node 1: Start of the method.
2. Node 2: if (x > 0) decision point (leads to Node 3 if true, Node 4 if false).
3. Node 3: y = x + 1 (executed if the condition is true).
4. Node 4: else branch (y = x - 1).
5. Node 5: Result of the else block.
6. Node 6: for loop decision point (for (int i = 0; i < x; i++)).
7. Node 7: y += i inside the loop.
8. Node 8: End of the method, print result.

Number of Paths:

- Path 1: 1 -> 2 -> 3 -> 6 -> 7 -> 8 (x > 0, loop iterates)
- Path 2: 1 -> 2 -> 3 -> 6 -> 8 (x > 0, loop does not iterate)
- Path 3: 1 -> 2 -> 4 -> 5 -> 6 -> 7 -> 8 (x <= 0, loop iterates)
- Path 4: 1 -> 2 -> 4 -> 5 -> 6 -> 8 (x <= 0, loop does not iterate)

STUDENTS-HUB.com
Uploaded By: anonymous

Software Testing Principles

1 - A necessary part of a test case is a definition of the expected output or result.

2- A programmer should avoid attempting to test his or her own program.

3- A programming organization should not test its own programs.

4- Any testing process should include a thorough inspection of the results of each test.

5- Test cases must be written for input conditions that are invalid and unexpected, as well as valid and expected

STUDENTS-HUB.com
Uploaded By: anonymous

Software Testing Principals..2

6- Examining a program to see if it does not do what it is supposed to do is only half the battle; the other half is seeing whether the program does what it is not supposed to do.

7-Avoid throwaway test cases unless the program is truly a throwaway program

Software Testing Principals..3

8- Do not plan a testing effort under the tacit assumption that no errors will be found.

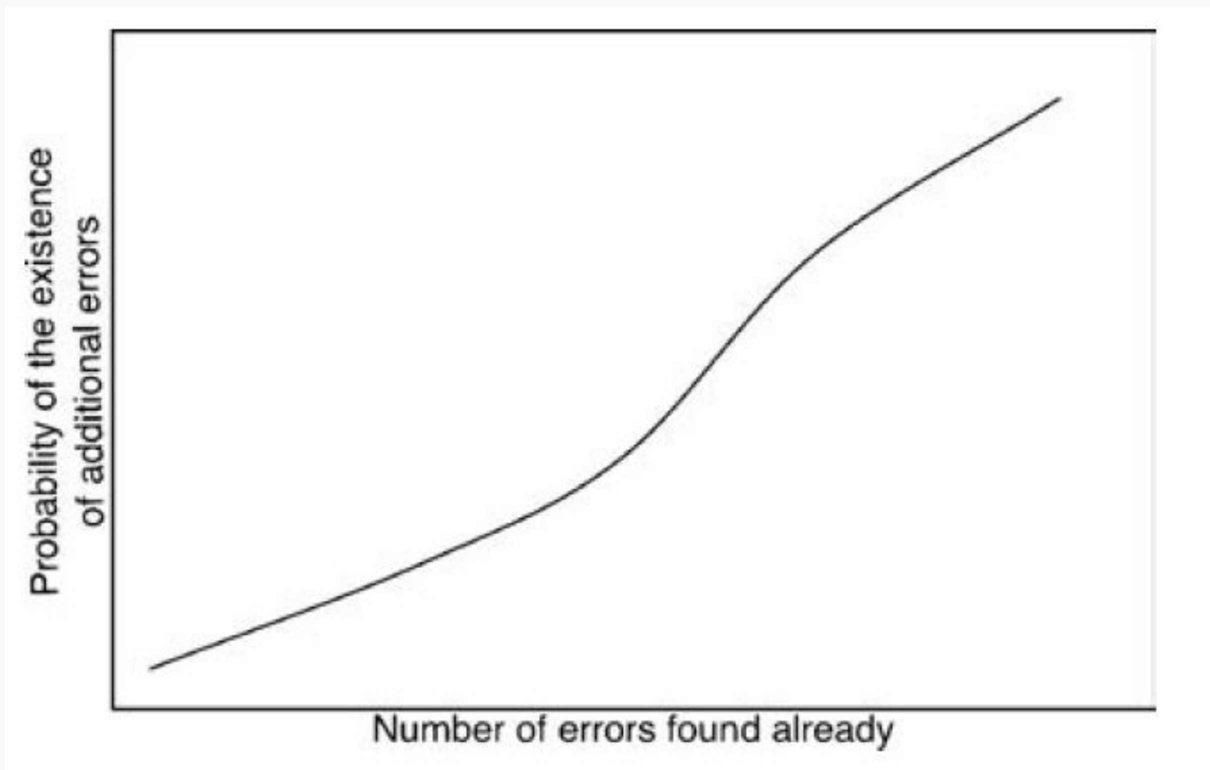
9- The probability of the existence of more errors in a section of a program is proportional to the number of errors already found in that section.

10-Testing is an extremely creative and intellectually challenging task.

STUDENTS-HUB.com

Uploaded By: anonymous

Probability of Number of errors



STUDENTS-HUB.com
Uploaded By: anonymous

Software Testing Principals..4

10 -Test Early, Test Often

“a bug found at design time costs ten times less to fix than one in coding and a hundred times less than one found after launch”

Ratio of costs in different phases 1:6:10:1000

STUDENTS-HUB.com Uploaded By: anonymous

Retesting vs Regression

- **Retesting:** You must retest fixes to ensure that issues have been resolved before development can progress.
- So, ***retesting*** *is the act of repeating a test to verify that a found defect has been correctly fixed.*
- ***Regression testing*** *on the other hand is the act of repeating other tests in 'parallel' areas to ensure* that the applied fix or a change of code has not introduced other errors or unexpected behavior.

STUDENTS-HUB.com
Uploaded By: anonymous