Software Quality Testing

Tasniya Ahmed
Lecurer
IIT
NSTU

Definition of Software Testing

- **Testing is the process of executing a program with the intent of finding errors.**(Myers)
- A successful test is one that uncovers an as-yet-undiscovered error. (Myers)
- Testing can show the presence of bugs but never their absence.(W.Dijkstra)
- Testing is a support function that helps developers look good by finding theirmistakes before anyone else does. (James Bach)
- ⍰ Execution of a work product with intent to find a defect**.**

Role and responsibilities of Testing

- **What are the roles and responsibilities of a Test Leader?**
- Involved in the planning, monitoring, and control of the testing activities and tasks.
- In collaboration with the other stakeholders, devise the test objectives, organizational test policies, test strategies and test plans.
- They estimate the testing to be done and negotiate with management to acquire the necessary resources.

Continue

- They recognize when test automation is appropriate and, if it is, they plan the effort, select the tools, and ensure training of the team. They may consult with other groups, programmers – to help them with their testing.
- They lead, guide and monitor the analysis, design, implementation and execution of the test cases, test procedures and test suites.

- They ensure proper configuration management of the testware produced and traceability of the tests to the test basis.
- As test execution comes near, they make sure the test environment is put into place before test execution and managed during test execution.
- They schedule the tests for execution and then they monitor, measure, control and report on the test progress, the product quality status and the test results,adapting the test plan.
- They write summary reports on test status.

- Sometimes test leaders wear different titles, such as test manager or test coordinator. Whoever is playing the role, expect them to plan, monitor and control the testing work.
- Along with the test leaders, testers should also be included from the beginning of the projects, although most of the time the project doesn’t need a full complement of testers until the test execution period. So, now we will see testers’ responsibilities.

Failure, Error, fault, Defect, Bug Terminology,

- **Failure:** the inability of a system or component to perform its required functions within specified performance requirements.

- **Fault:** An incorrect step, process, or data definition in a computer program.

- **Error:** A human action that produces an incorrect result.

- An **error** can be a **grammatical error** in one or more of the code lines, or a
- **logical error** in carrying out one or more of the client’s requirements.

- Not all software errors become **software faults.** in some cases, the software error can cause improper functioning of the software.
- In many other cases,erroneous code lines will not affect the functionality of the software as a whole.

- **A failure** is said to occur whenever the external behaviour of a system does not conform to that prescribed in the system specification.
- A software fault becomes a software failure only when it is “activated

What is the difference between software fault and software failure?

- Software failure occurs when the software does not do what the user expects to see. Software fault, on the other hand, is a hidden programming error.

  A software fault becomes a software failure only when the exact computation conditions are met, and the faulty portion of the code is executed on the CPU. This can occur during normal usage. Or, when the software is ported to a different hardware platform. Or, when the software is ported to a different complier. Or, when the software gets extended.

BUG

- **Definition:** A bug is defined in simple term as any error or mistake that leads to the failure of the product or software either due to the specification problem or due to communication problem, regarding what is developed and what had to be developed.

1. Bug Terminology: Failure, Error, Fault

- The various terms related to software failure with respect to the area of application are

listed as **Defect, Variance, Fault, Failure, Problem, Inconsistency, Error, Feature,**

- **Incident, Bug, and Anomaly.**
- i. ***Problem, error, and bug*** are probably the most generic terms used.
- ii. ***Anomaly, incident, and variance*** don’t sound quite so negative and infer more unintended operation than an all-out failure.

- iii. ***Fault, failure, and defect*** tend to imply a condition that’s really severe, may be even dangerous. It doesn’t sound right to call an incorrectly colored icon a fault.
- These words also tend to imply blame: **“It’s his fault that the software failed.**
- iv. As all the words sound the same they are distinguished based on the severity and the area in which the software failure has occurred.

- v. When a program is run the error that we get during execution is termed on the basis of runtime error, compile time error, computational error, and assignment error.

- vi. The error can be removed by debugging, if not resolved leads to a problem and if the problem becomes large leads to software failure.

- vii. A bug can be defined as the initiation of error or a problem due to which fault, failure, incident or an anomaly occurs.

- viii. The program to find the factorial of a number given below lists few errors, problem and failure in a program.

- **For example**

```
#include<stdio.h>
void main()
{
int i , fact, n;
printf(“enter the number “);
scanf(“%d”,&n);
for(i =1 ;i <=n;i++)
fact = fact * i;
printf (“the factorial of a number is ”%d”, fact);
}
```

- As in line number 4 the fact is not initialized to 1, so it takes garbage value and gives awrong output, **this is an example of a bug**.
- If fact is initialized to zero (fact = 0) than the output will be zero as anything multipliedby zero will give the output as zero.
- **This is a bug** which can be removed byinitializing fact = 1 during initializing.
- As the fact is declared as integer, for the number till 7! Will work perfectly. When the
- number entered is 8, the output is garbage value as the integer limit is from – 32767 to
- +32768, so in declaration change the initialization to long int fact;

Causes of software defects

- 1. Faulty requirements definition
- 2. Client-developer communication failures
- 3. Deliberate deviations from software requirements
- 4. Logical design errors
- 5. Coding errors
- 6. Non-compliance with documentation and coding instructions
- 7. Shortcomings of the testing process
- 8. User interface and procedure errors
- 9. Documentation errors

Objectives of Testing

- Software Testing has different objectives.
- The major **objectives of Software testing are as follows:**
- Finding defects which may get created by the programmer while developing the software.
- Gaining confidence in and providing information about the level of quality.
- To prevent defects.

- To make sure that the end result meets the business and user requirements.
- To ensure that it satisfies the BRS that is Business Requirement Specification and SRS that is System Requirement Specifications.
- To gain the confidence of the customers by providing them a quality product

Test Scenario: - its design aspect.

- **Test cases** are written using **test scenario**. It describes each transaction and expected result of each transaction included in test scenario.
- Test cases are executed through test data.
- Actors (Person, system, hardware, or software) are taking part in transaction.

- Transactions represent the interactions of the actors with system under testing.

- Test case characteristics: Accurate, Economical, Real, and reusable, Traceableto requirements, Appropriate, Self standing, Self-cleaning… etc.)

Test Case

- **How to write a good test case?**
- Test case is an important document from validation of system perspective.

- These must be written at each stage of validation from unit test till acceptance testing.

Basic principles of writing a good test case are:

- Test case must be testable.
- Tester should know what is to be done when to wait for system to do it.
- Inform tester each transaction displayed/ replied by the system on screen at each step. And wait for user response.
- Use simple conversational language for writing test case, which improves clarity and avoid communication losses.

- Use consistent names of fields must be used in place of generic names. Any change in field name must be incorporated in test cases.
- Tester should aware windows basics.
- Order of the test cases must follow business scenario. Avoid time wastage,

Common Mistakes in writing test cases:

- Making test cases too long and combining two or more test cases in single test should be avoided
- Incomplete, incorrect, and incoherent test cases can create confusion and frustrate testers.
- Steps should be made very clear in test case steps.
- Test case changes must be updated in software user interface.
- Define pass/fail criteria correctly, i.e. test are successful or not, there is a defect or not?

Essential activities in Testing:

- Maintain test strategy, test plan, test scenario and test cases in a location so that they are available to concerned people when required for test artifacts.

- Follow configuration management standards, and reasons for updates, Test cases used for testing must be reviewed before using them.