Software Testing

What is Software Testing?

Whether the application is functioning or not.

Main objective is to find ‘Defects’ – difference between actual result and expected result.

Expected Result:

Name:

OK

Customer req: application with a field to enter the name. if I enter any name and click on Ok button then it should display a welcome message else error message should be displayed.

Valid: welcome message, Invalid: error message

Test above module:

Name:

Cancel

OK

After execution:

Valid: Abc ------ Welcome message

Invalid: abc@123 ----- error message.

In above module, functional point of view there is no defect.

But there is a defect – UI point of view – UI defect.

**Testing Principles**

There are total seven principles in testing

1. Exhaustive testing is impossible.

Testing each and every combination of input is impossible.

Example:

Name: \_\_\_\_\_\_\_

Valid | Invalid

-------- |------------

Abc | @RJ

Komal| 123komal

IET | 4abc

To test “name” field, we can have infinite set of values.

As a tester I can’t test each n every input value for the name field.

To overcome this problem, testers were introduced with some techniques to derive the inputs for testing.

1. Testing only shows presence of defects not its absence.

Example:

Name: \_\_\_\_\_\_\_

Valid | Invalid

-------- |------------

Abc | @RJ

Komal| 123komal

IET | 4abc

Above values are derived using the testing techniques.

Tester will the values for testing the name field.

For valid values, the application displayed a welcome message.

For invalid values, the application displayed an error message.

Tester assumes that if for the derived values my application works as expected then for other values too from that partition it will behave the same.

According to our assumption, let’s say tester tried 10000000 values from each partition using the techniques.

Customer can try any random input in the name field which may have been tested or may not have.

Customer tried “Z2A” as an input in Name field.

Customer got a welcome message. Customer will report it as a failure.

Question. Can any software be 100% defect free?

Answer: No because exhaustive testing is impossible.

1. Defect Clustering

Defects are always present in groups in our application because of the connectivity between the modules.

We have to find the root cause of the defect, remove it.

Search for Pareto Principle

1. Pesticide Paradox

We should not repeat the techniques used for testing on the application multiple times. We should update the test cases, test related methods regularly.

Example:

Hit Spray: since 20+ years

Current price of Hit Spray is Rs. 150

Today,

Hit Spray | Hit Spray(20 yrs ago)

Rs. 150 | Rs. 10

We will prefer Rs. 10 hit over Rs. 150 hit spray.

1. Testing is context dependent.

Context: Domain

Depending on the domain of the application we may have changed the efforts or focus areas while testing.

Gaming Application | Internet Banking Application

Graphics | Security

Animation | Functionality

UI – color |

Speed

Functionality

1. Absence of Error fallacy.

Finding and fixing defects won’t satisfy the customer.

We must focus on the other attributes of the application as well to satisfy the customer. Attributes like performance, speed, UI. Compatibility, Security, etc.

1. Early Testing

Start testing activities as early as possible.

Right from requirements phase we should start the testing activities i.e. verification of documents.

Early testing doesn’t necessarily say’s that testers should be involved from first phase.

Configuration Management:

Change control board or Version control board.

Helps in managing the changes requested by the customer.

Customer is requesting for adding new requirements: Change Request: 10 CR

Business Analyst (BA) will interact with customer and will gather the change requests. BA performs a process called as “Impact Assessment” – will the change request affect the existing application or not? If yes, which modules will be impacted.

CCB – change control board team members analyses the IA report submitted by the BA and decides which change requests to accept and which to deny.

Accepted CR: 7,

Denied CR: 3

CMT – Configuration Management Team implements the changes approved by CCB team. Implementing the change requests is nothing but adding the new requirements into the existing project.

Basically the further activity i.e. requirement analysis, req docs, designing – docs, coding, testing, is performed.

All the documents will be upgraded I.e. the version of the documents will be changed, build version will be changed.

PM – Project Manager forwards the approved CR to CMT.

After implementing all the CR CMT acknowledges it to PM who further forwards it to CCB members.

CMT

PM

CCB

BA

CR

Impact approve or deny implement the CR

BRS 1.1

Assessment

SRS 1.1

HLD 1.1, LLD 1.1

Source code 1.23, buildv1.5

Testcases 1.1