Software Testing - It is a process that assures quality by verifying and vaildating the given requirements.

It is not a single activity..It is rather a process

We plan about

What to test

How to test

Execute the test plan

Reports and Results

What plays a key role in success of a business and who are the main promoters of any business?

Customer Satisfaction..Customers – word of mouth

We need to find defects of application in early stages – that’s is why testing plays a key role.

What happens if some defects are found in **production** or **live** environment?

You might lose your customers and it is very costly to fix.

40 – 100 times more than the cost it takes to fix a defect early

Add to cart :

When user adds a product its hould be added to his list

He can still continue shopping and keep adding to the cart.

public void addToCart(int productId){

sumOfItems(productId);

}

function int sumOfItems(int productId){

int listOfItems;

if(productNum is valid){

listOfItems++;

}

return listOfItems;

}

When we write a script/program to test another program – that is whitebox testing/unit testing….

Public void testSumOfItems(){

expectedOutput = 1;

actualValue = sumOfItems(56789);

if(expected == actualValue){

then requirement is satisfied…

}

}

Verification: White box testing – we actually have knowledge on the code and we are trying test the code as per the requirements.Developers take care of verification.

Validation: black box testing – where we try to validate that application is functioning as per the requirement.

Testing Team or Qa team take care of validation

Ecommerce website – 25000 different products..

You are now testing the search box..

Is it possible for you to test all the products? do you write 25000 test cases???????

NOOOOO………….

We need to think about quality test cases but not quantity..

We can never do all possible permutations and combinations in testing….

Positive test cases and negative test cases

int sum(int a , int b){

return a+b;

}

4, 5 – 9

40, 55 – 95

0,0

-5, 7 = 2

-9, -10 = -19

0, null = null

null, null

Test Strategy – **Test strategy** is a high level document which defines the approach for software**testing**. It is basically derived from the Business Requirement document. **Test strategy** is developed by project manager or business analyst.

high level document which talks about requirements, resources,test methodology, tools…

It says what methodology/ criteria/rules we follow for testing…

It is given by BA/ ProductOwner/ProjectManager

Eg: traffic department defining traffic rules to be followed by every citizen

Test Plan – A test plan documents the strategy that will be used to verify and ensure that a product or system meets its design specifications and other requirements. A test plan is usually prepared by or with significant input from [test engineers](https://en.wikipedia.org/wiki/Test_engineer).

It is given by Testlead/QaManager along with inputs from test engineers.

how to test, what to test, who will test, allocation of resources, entry and exit criteria…tools..environment…

Eg: We try to achieve/follow the rules by traffic signals, signboards, no parking, u turn boards

Test Case –

Steps to be followed to test a requirement

Actual result

ExpectedResult

Test execution (manual or automated)– gives us actual result which we compare with expected result…

The faster we have product ready to end users –the faster we get the profits /return on investments…

We need quality within limited time span…..

SDLC:

Requirements/analysis

Design- high/low level

Development

TEsting

Maintenenace

Testing life cycle :

Plan Testcases

Execute test cases

Find defects

Assign to developer

Additionaltestcases/actual testcases

Regression testing(check for entire functionalty along with defect testing)

An issue found can always give further ideas/ways to think of other issues….

Defect – first name should be actually alphanumeric but not only numeric..

Actual -Its accepting all numbers

Now what happens if there are all special characters

What happens if it has invalid special characters….

Static testing – in early stages you try to verify the requirement doc /user story whether it is all good to go or it has some misinterpretations..

When you find some issue in docs you report it to BA/PO/PM and try to fix it in requirement stage only…

You are not actually testing the application but testing the requirement..

Dynamic testing – where you test application against requirement..

Types of Testing :

Functionality (verify against given requirements) – 80%

Usability Testing –

Happy Customers

Look and feel

Proper font family and proper font size

Font colors

Alignment of html elements(text boxes,links, plaintexts,dropdowns,radio buttons,scrolling…..)

Layouts and templates

Cosmetic – typos

Links, navigations, connections

Ajax calls- something that load dynamically without page being refreshed is ajax call..we are trying to reduce the space required/cut down the no of pages …

Responsive UI design – it is adapting to the screen width and resolution

Single page applications- loaded only once and later the data required only is fetched by ajax calls…

We are trying to check resources like speed and memory/space..

PerformanceTesting:

Load testing – max load you need to test for now

Stress testing – peak point where my application breaks

Volume – the data that is stored in database

SecurityTesting:

Authentication

Authorization

Integrity

Vulnerabilties – ways to hack – crosssite scripting, sql injection

Select \* from Users where username=””delete from Users”” and password=”Test123” or “1=1”

Pass some javascript code and hack the user details, cookies,

Different test environments:

Dev, test, stage, prod

Dev, sit, uat, prod

<http://api.dev2.talentworks.io/api/v1/common/levels>

class{

int[] studentIds = new int[50];

public void getIds(){

if(student.isValid){

studentsIds.add(student);

}

Establish a database connection:

Host

Port

url

DriverManager

Close the connection…

File management:

IO activities- read a file …data stream..close the connection..

Api- webservice calls…

Open and close connections…

}

Automation:

SIGN UP PAGE

Smoke Testing –

1.signup page should be loaded when u click u signup link..

2.Enter details

3.Click on create – you should success page..

Functional Testing –

Valid details – create arecord in db—again when u try to login back – it should take to account

Invalid – appropriate error msg validation- email is wronf format..pwd is wrong….

UsabiltyTesting – look and feel

RegressionTesting(test cases) – unit, functional, system…..

SystemTesting….

Automation:

Reduce cost for many resources

Save time

Increase of effciency of testing

QTP

Selenim IDE

Selenium RC

Appium

Jmeter

SoapUI

HttpClient

Robotium

Calabash

Js based:

Protractor

Jasmine

Nightwatch.js

Automation Fw: Structure to the scripts with reusability….avoiding redundant tasks…

Planning

Structure

Reusability

Logging

Reporting

Error handling

TestPlan:

How to do…

Scope(new requiremnets), resources(2),time,tools,

TestStrategy:

What you need to do?

IntegrationTesting

Entry criteria …exit criteria….

FunctionaltyTesting

Usabilty testing

PerformanceTesting

SecutiryTesting

Bug/Defect/issue – deviation from requirement

Error/

Priority 1 2 3 4

Severity low medium high

http – protocol that transfers data across web

maintain the state of http:

cookies

session Id

xml and Json: QA/UI

Xml – extensible markup language

It describes the data

Username

pwd

order id

Client- server architecture – request-response

Ordered-84784

Product num - 7899

product name-Watch

tracking status

shiiping address

xml

schema- what are valid wr.rto ur xml/rules to define xml tags

xml - uses tags and heavy weight as it concentrates on opening/closing/casesenstive

JSON- javascript object notation

Light weight than xml

No need to write individual tags

Key value pair..

U can have nested data in both xml and json

<student>

<id>56</id>

<name>geetha</name>

</student>

“student”:{ “id”:56, “name” : “geetha”}

Xpath and css selectors:

Xpath – It Is lang used to query the xml data

Absolute xpath : "/html/head/link"

Starts from your root element and traverses acrosss your child elements. It is lengthy and inefficient.

Eg: html/body/nav/p/div[2]/ul/li[1]/a

Relative xpath : “//link”

Relative xpath locates elemnt from anywhere in the document.It starts with //. It is shorter and efficient trhan absolute xpath.

.//\*[@id='navbar-collapse']/ul/li[1]/a

Diff ways to write/identify the elemnts:

1. Write xpath to find all the paragraph(<p>) elements in html page

//p

1. Write xpath to find the 3rd paragraph element in page

(//p)[3]

1. Locate last but one hyperlink (<a>) in html page

(//a)[last()],(//a)[last()-1]

1. All child elements of a given parent element

//body[@class='home-page']/\*

1. Element with specfic id attribute

//body[@id='home-page']

1. Elements with specfic class

//body[@class='home-page']/div[2]

7.identify element thru text

//div[text()=’Hello’]

.//\*[starts-with(@class, 'home')]

Css selectors:

Cleaner,faster and good support for all browsers..

Simple selectors:

Id attribute- #navbar-collapse

Class attribute- .contact

By tag name – img or p or div

Nested selectors:

* > - dierct child of element
* Space – will give all the childs

+ - sibblings

<div  id="modal-overlay-signup" class="modal grow modal-overlay modal-backdrop-body fade ng-scope">

<input type=”text” id=”training” class=”.wblText”>

Attribute selectors:

Input[placeholder\*=’Phone’]

div[type=’test’]

Pseudo selectors:

Div:empty

Structural pseudo selectors

Div:first-child

Form div:last-child

Write your own xpath & css selctors for locating all whiteboxqa.com header and footer elements