Testing - s/w appln --->

-to check with - client reqs
-Expected result=Actual Result
-identify the defect --> defect free product
-quality of the product -customer perspective

Flow of testing-
Manual testing

Developers--->appln---> testers---> Test Cases (ER=AR)---> If ER <> AR ---> rectify (developers)

Test Cases-
Excel sheet
Test Management tools

Automation testing -

Developers--->appln---> testers--->tools---> Test Script (ER=AR)---> If ER <> AR ---> rectify (developers)

Tools-
1.Functional testing tool---> behavior of the appln will be tested . functional charcateristics of the appln .

1. QTP (10)/UFT (11) ---> HP
2. Selenium ----> thoughtWorks -open source tool
3. Silk test ---> Borland
4. Rational functional tester ---> IBM

2.Non-Functional Testing tool ---> testing to check the surrounding of the appln to work.

1. LR- HP
2. OPENSTA - Cyrano
3. Silk prerformer --->Borland
4. Rational performance tester ---> IBM
5. Jmeter

3.Test Management tool - managing the STLC phases

Req gathering & analysis
Test Plan
Test Development
Test Environment
Test Execution
Defect Tracking
Test Closure

1. QC(10)/ALM (11) --> HP
2. Bugzilla
3. JIRA
4. Test Director

Types of testing-

1. Static - verification - are we building the product right

checking during the development of appln ---> whether are we going in the correct direction

Techniques

1. Reviews

2. Dynamic -validation -are we building the right product

after the development of appln we check whether it is developed according to client expectation

1.Structural based testing - behavior of the code -white box testing

Techniques

1.Statement coverage
2.decision coverage
3.Condition coverage
4.cylomatic complexity

2.Functional based testing - behavior the appln will be tested ---> black box testing

1.Equivalence Partioning
2.Boundary value analysis
3.Decision tabe testing
4.State transition testing

3.Experienced based testing - based on the experince of the tester

1.Error guessing
2.Exploration testing

Principles of testing -

1.Testing shows presence of defect
2.Exhaustive testing is impossible
3.as early as SDLC
4.Defect clustering
5.Testing is contect is dependant
6.absence of error fallacy
7.Pesticide paradox

STLC-

Req gathering & analysis
Test Plan -roadmap for the testing activity

Test Managers

Word doc
TestMgmt tool

Test Scope
1.What to be tested

In scope
-features to be tested

Out Scope
-features not to be tested

Test Schedule -
2.When to test

Roles , Responsibilites & skills needed
3.Who-

Test Environment
4.Where - tools, H/W ,S/w required for testing

5.How - Manual or Automation testing

6. Suspension & resumption -

8. Test Deliverables

Test plan
Test Case design
Test case execution
Defect tracking sheet
Requirement traceability matrix

9.Risk analysis

- possibilities of defect may happen to the appln

-prevent the defect
-reduce / overcome the defect

Test Development- Preparation done to check ER=AR

*Test Scenario - Testers
Manual- excel
Automation tool- Test mgmt tools

-high level of testing
-overall testing
-end to end testing

login page

cancellation in IRCTC-

* user cancellation
* waiting list ticket and cancelled after chart prepared

* train itself cancelled

Types- Scenario & Cases

- + ve - what the system should do for the the user
- - Ve - what the system shouldnt do for the users
- Alternative - other way of achieving the positive scenario / Cases
- Exception - conditions will lead valid to invalid

*Test case Design
- ER= AR

-Field level validation
-Form level validation

- username accepts min 4 to max 8
- amazon - 1000 - 5000 - del -100

Test Environment -setup required to test

Test Execution -

Test Case despendency

if a particluar step is failed
can we execute the next step- condition to be checked

Yes - Failed status
Continue the next step of same TC

No - Failed status
Remaining step -Blocked status
Continue to execution for next test case

Defect Tracking

when ER <> AR - test case

communicating the defects to the developers

testers & developers

testers creates the defect tracking sheet

Manual -Excel
Automation- Test Mgmt tools

-severity - impact of the defect
- technical aspects
- type of defect

Types of severity

1.blocker / showstopper - further the execution is not possible .

2. critical /major - major req or critical req has defect

High

3.minor - minor req is not working

- medium

4.trivial - cosmetic mistakes

5.enhancement- suggestions

- Low

-priority - importance to which defect has to be resolved first

- bussiness

Logo & theme
S - L
P - H

-defect life cycle -

1. Unconfirmed - Tester - tester is not sure whether it is a defect
   - triage meeting (TM)

  Vote -
  valid bug - NEW - TM
  invalid bug - REJECTED - TM

2. When a NEW defect

  can we resolve in current version - OPEN - TM
  we need time to work on it (later version) - DEFERRED - TM

3.ASSIGN - TM - assign to developers

4.FIXED - Developers - resolved by developers but need to be verified by testers

5.Retesting - test again to check whether defect is resolved
   Regression testing -

6. CLOSED - Testers -defect is resolved
   RE-OPEN - Testers - defect not resolved --> ASSIGN to developers

Test Closure - RTM - last phase of STLC

- test mgmt tools
- charts

- reports- line ,bar ,pie
- metrics (measurements)

  - how many test case created / tester = for a week
  - how many test case executed
  - how many test case are in open status
  - age of a defect for a severity type

what we have planned and what we have executed

test plan ---

RTM - requirement traceability matrix

- roadmap for requirements
-it will ensure whether the req is covered to all the phases of STLC

- forward RTM - from SRS to defects

ensure whether the req is covered to all the phases of STLC

-Backward RTM - from defects to SRS

identify the root cause

Techniques -

* Static -
Techniques

1. Reviews

* Dynamic

1.Structural based testing - behavior of the code -white box testing

Techniques

1.Statement coverage

- atleast each of the stt to be executed once

eg- int a,b,c
    c=a+b
    print c

coverage = (total no of lines executed / total no of lines) * 100 =3/3 * 100 = 100%

```
int a,b
if a >b
print a
else
print b
endif
```

= 3/6 = 50 %

2.decision coverage - atleast each of the decision to be covered once

coverage= total no of decision executed/ total no of decision * 100 = 1/2 * 100 = 50 %

3.condition coverage

```
if ( date= 1 ) (month= Jan ) ( Time= 12)
  print "happy new year" - T
else
print "happy day"        -F
```

D	M	T	o/p
T	T	T	T
T	F	T	F
T	T	F	F

4.cylomatic complexity =basis path testing

Flow chart

edges = 8
nodes = 7
part = 1

edges - nodes + 2p
8 -7 +2(1)
1+2 = 3

```
if a=250
if A>B
print A=B
else
print B=C
end if
endif
print A
```

2.Functional based testing - behavior the appln will be tested ---> black box testing

1.Equivalence Partioning - divide the large set of data to a subset of date

=1 valid class
= 2 invalid class

min 4 to max 8 chars

amazon 1000 to 5000 - del - 50

less than 1000 1000 - 5000 above 5000
500(invalid) 2500 (valid) (invalid) 6000

less than 24 24-40 above 40

1.boundary value is not checked
2.tester to tester will vary with inputs

2.Boundary value analysis (BVA)

divide the large set of data to a subset of date

n,n+1,n-1

n=1000 n=5000
n-1= 999 n+1=5001
n+1= 1001 n-1=4999

n=10 n=21
n-1=9 n-1=20
n+1=11 n+1=22

9,10,21,22

10,11,20,21

3.Decision tabe testing -

Rules
Condition
Action

IRCTC appln - search for ticket

from
To
Date
No of tickets

Login
username

password

4.State transition testing -

login

valid details - login
invalid deatils- error msg - 1 st

valid details - login
invalid deatils- error msg - 2nd

valid details - login
invalid deatils- error msg - 3rd try - account locked

3.Experienced based testing - based on the experince of the tester

1.Error guessing

Date - numbers
format dd/mm/yyyy
d- 31
m -12
yy - 4 digits
DOB- past dates
booking - present and future dates
leap year feb -29

2.Exploration testing -understand the appln

Reviews

Rework - working onthe areas where suggestions was given

Follow-up - checking whether the status of dadlines are completed

informal review - within the team discussion happened will be called as informal

No Plan, no documenatation

technical

NO MOM

Semi-formal

author presenting

reviewer is attend the meeting - suggestions if required

Scribe -record the meeting

Plan & documenatation is done

MOM is prepared

Formal

Moderator - conclusion - go or no go

Plan & documenatation

MOM will be prepared

Levels of testing

UT - CT component testing

-developers

-white box testing

-functionality of the code

-single module will be tested in terms of code

IT - CIT

-testers

-gray box testing

-data flow from one module to another

Appraoches-
Big bang -
Increemntal

System testing-

testers

black box testing

as a whole the entiire appln will be tested

Functional & non -functional testing

SIT - S1 --- > S2

testers

black box testing

AT

alpha & beta

-client

-black box

-SRS

Testing types

Build verification testing

before execution

smoke testing-overall testing

sanity -in depth - critical test cases

adhoc

compatability

SCM- change of req (CR) ---change request --->CCB ---> any loss

cost

time

change request is accepted ---> Configurable item (CI)

not accepted --> change request is closed

version control

past	present	future
	CI	closed
CI	Closed- CI-1	
	windows XP	vista
XP	vista	W7

CI
-test plan
-test scenario
-test cases

NCI
-MOM
-weekly report

Introduction to Automation testing-

- appln---> testesd---> tools --> ER=AR (test Scripts)

IF ER<> AR --> Developer (rectification)

Types of automation tools

1.Functional testing tools
2.Non functional testing
3.Test Management tools

Tool ---> Robot (instruction need to be given)Human intervenetion

Advantages of Automation testing-

1.Reusability
2.reduntancy
3.avoid time consuming
4.efficiency
5.accurate
6.reliable
7.robust

Selenium -

Features of Selenium-
1.Functional testing tool
2.Regression testing
3.Compatability testing
4.Open source tool
5.Official website---> seleniumhq.org (s/w)
6.Selenium can tested using most of the programming languages
7.It can test only web based applications
8.Selenium--->Suite of products
9.ThoughtWorks

Selenium components
1.Selenium IDE

Features of SEleenium IDE
- Add-on to firefox (fully supported) & chrome (cant use other browser to test the appln)
- Script to be in HTML & javscript (we cant use any other scripting or programming languages)
- record option is there (only)
- In- built test results
- test script & test suite can be created & executed
- by default we donthave conditional & looping stts (.js file)
- Alertbox cannot be handled
- datadriven & database testing cant be performed

SCenario-

my atos syntel login page - single username & password
23,000 employees Us& PSWD

single script- 23,000 data

parameterization-passing multiple values to a single script

framework- Datadriven testing

table- & source view - experts

script can be generated in two ways-

1. record based - initially level of tester
- tester instructs ---> tool will generate the scripts

2. descriptive programming - tester needs to write script

what will script contain -

1.which element/object to test - how to identify the element ---> with the help of attributes or properties - target
2.what action to be performed - command
3.value to be entered (if cases) -value

target locator type=locator value

locator types-
1.id
2.name
3.css - styles

worst case

tagname[attribute='value']

4.xpath

absolute - /html/body/form/input[4]
relative - //tagname[@attribute='value']

//input[@name='Gender'][2]

5.link = text of the link

6.DOM - document objuect model

document.forms[0].elements[0]
document.forms[0].elements[1]

Selenese - commands in selenium

1.action - activity perform on elements - open,click,wait,type,select,echo

2.acessors - stores value which to be used in the appln

store,storetext,storetitle =${a}

3.assertion - verify ER=AR

assert , asserttitle,assertext -if ER<>AR , it will abort its execution
verify , verifytitle,verifytext -if ER<>AR , will continue the execution

Syncronisation - match the speed of appln with the speed of tool

30 secs

wait
thread.sleep

clickandWait -

Waitfortitle -

2.Selenium RC-

- Selenium 1 remote control -server

Features -
1.we can test using any browsers
2.we can use any operating system
3.we can use any scripting & programming language
4.Jar file (add any editor - eclipse)
5.we can test for datadriven testing & database validation
6.it supports alert box
7.we can have conditional and looping stts in the script

Disadvantage-
1.no direct communication b/w tester & appln

Architecture of RC-

Script (java) --->RC server (client libraries + HTML / Javascript)---> Appln(browser)

Selenium Webdriver-

Features-
1.we can test using any browsers
2.we can use any operating system
3.we can use any scripting & programming language
4.Jar file (add any editor - eclipse)
5.we can test for datadriven testing & database validation
6.it supports alert box
7.we can have conditional and looping stts in the script
8.direct communication b/w tester & appln

Script - java (jar file)
(Junit) ---> appln (browser - chrome- Driver server)

Scripte-1

System.setProperty("webdriver.chrome.driver",path);

locator types-

1.id
2.name
3.cssSelector
4.xpath
5.linktext- exact text should be known, partiallinktext -
6.classname
7.tagname

methods-
textbox- sendKeys
radio, checkbox , link -click
button- submit ,click

close the browser

driver.close();- current tab will be closed

driver.quit(); - close the entire browser

chrome- chrome
firefox-gecko
internetexplorer-ie

drop down listbox-

Select methods-

1.selectbyvisibletext
2.selectbyvalue
3.selectbyindex
4.ismultiple - boolean true or false

5.deselectbyvisibletext
6.deselectbyvalue
7.deselectbyindex
8.deselectall

alertbox
driver.switchTo().

ok- accept()
cancel-dismiss()
message- getText()

frames

wait stts-

1.implicitWait- throughout the script a single wait stt is included for all the elements tested

2.explicitWait- wait stt to a certain element alone

WebDriverWait var1=new WebDriverWait(d1,10);

var1.until(ExpectedConditions.visibilityOfElementLocated(By.name("sel")));

var1.until(ExpectedConditions.alertIsPreseframeToBeAvailableAndSwitchToIt(locator)nt());

var1.until(ExpectedConditions.elementToBeClickable(locator));

var1.until(ExpectedConditions.elementToBeSelected(element)

var1.until(ExpectedConditions.frameToBeAvailableAndSwitchToIt(locator)

3.fluentWait -

1.wait stt
2.peak or frequency
3.ignoreexceptions

Wait var_name = new FluentWait(driver)
牋牋.withTimeout(30, SECONDS)
牋牋.pollingEvery(5, SECONDS)
牋牋.ignoring(NoSuchElementException.class);
牋 wait.until(ExpectedConditions.(providethemethod))

navigate commands

- driver.navigate().forward()
- driver.navigate().back()
- driver.navigate().refresh()
- driver.navigate().to(url-parameter)

get commands-

d1.get(url)
d1.getText()- gets the inner html value

d1.getPageSource()
d1.getCurrenturl()
d1.getTitle()

max & min window -

d1.manage().window().maximize();
Dimension min=new Dimension(300,250);

d1.manage().window().setSize(min);

AutoIt- third party tool - open source tool

the interface of window based in web based appln.

3 ---> .au3

different windows
mouse movements
WebElement
TestNg
4.Selenium grid