

Software testing :- Part of Software development Process to identify defect and identify the defects in the software.

Project : developed based on the requirement of a specific customer.

Product :- developed for multiple customer's based on the market requirement's.

Error :- human action that produces an incorrect result.

Bug/defect :- Deviation from the expected behavior of the system.

Failure :- Deviation identified by the user while using the system is called a failure.

SDLC ; Software Development Life Cycle Process used to design and develop softwares.

Requirement analysis → Design → Develop → Testing → Deploying
[maintenance]

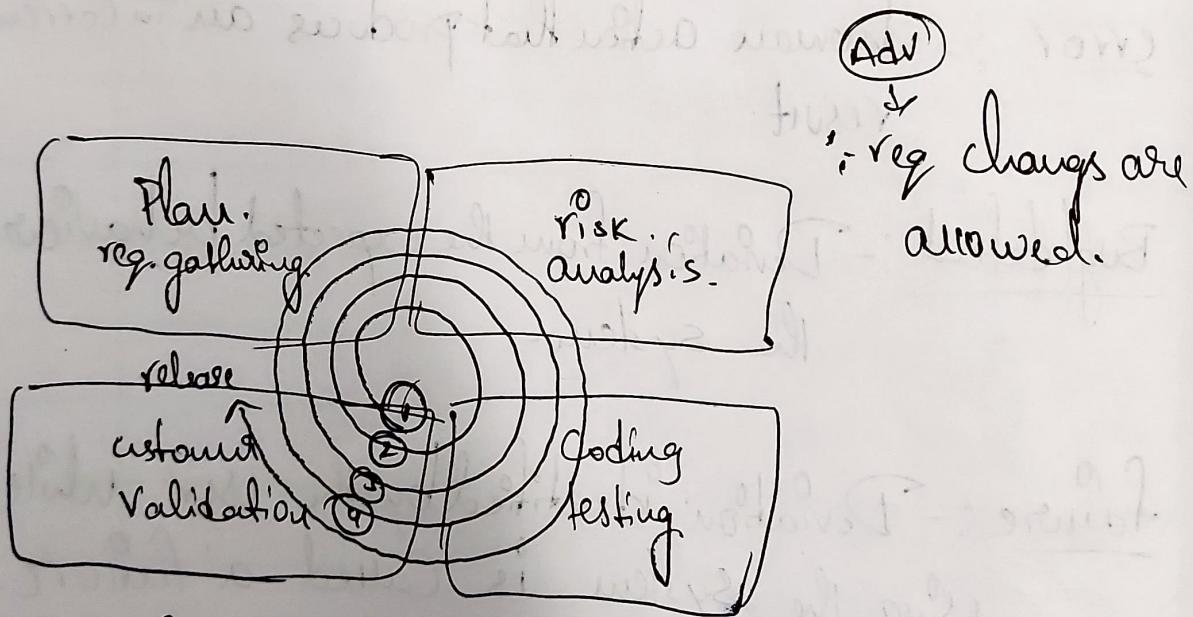
Waterfall model

Adv and disadv.



- ⇒ Requirement changes won't be allowed, so bugs are rare in this model.
- ⇒ Preferred for smaller projects.
- ⇒ any defect is present it'll be continued in the later phases. (disadv)

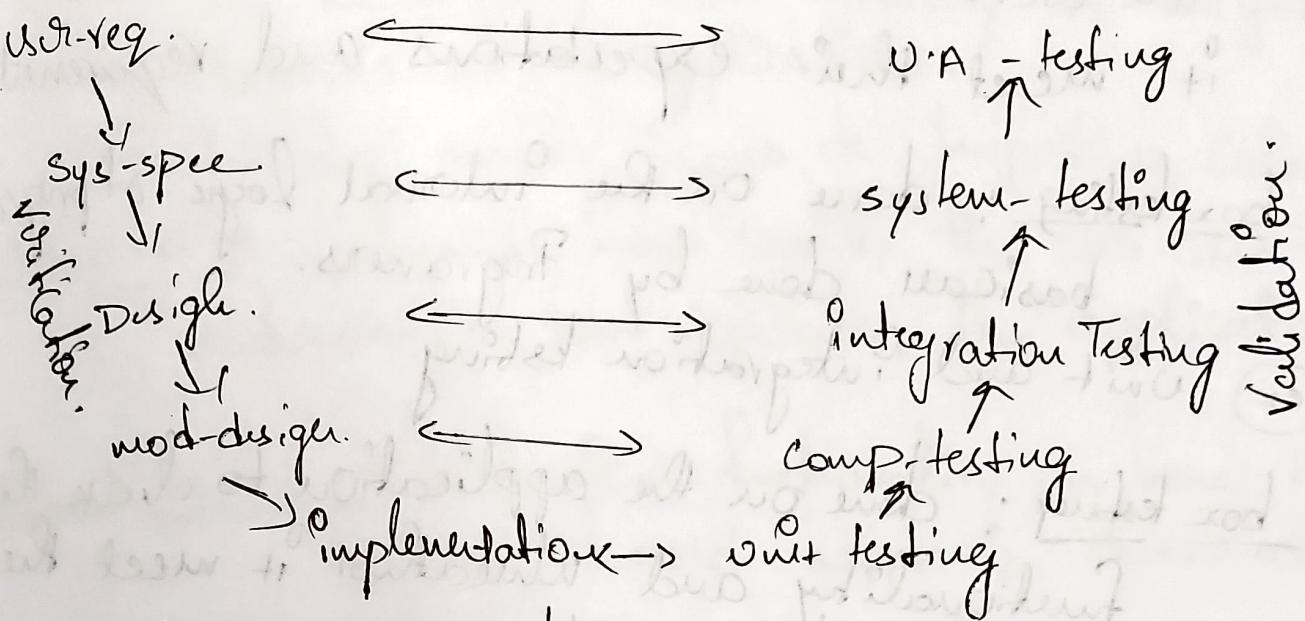
Spiral model [vision control + model]



- :- it is a iterative type model and a new build of software is released in every cycle.
- :- after the char analysis and customer validation the software is released.

V-model

ISU



:- testing done in every phase

Static testing :- testing through form of review and inspection

Dynamic testing :- testing the software by giving input and observing the result.

Dynamic testing techniques

① Unit testing :- testing the functionality of the individual components (or) functions that they work correctly in isolation. done on single module.

② Integration testing :- checking how diff modules integrate with each other and ensure a seamless integration. done on 2 (or) more modules.

③ System testing :- checking the complete functionality of the system to ensure it meets the requirements.

VAT: - Real user's test the software and check if it meet their expectations and requirement's

white box testing: done on the internal logic of program basically done by Programmers.

(c) Unit and Integration testing

black box testing: done on the application to check the functionality and whether it meet the requirements or not.

(c) System and VAT testing.

Grey box testing: Combination of white and black box testing.

(c) database testing.

System testing

i) User interface testing (GUI)

ii) Usability testing

iii) Functional testing

iv) Non-functional testing.

GUI

USA

func → obj prop

non

error handling

calculator

Cookie and session

database testing

file testing

GUI Testing: Graphical user interface testing focuses on verifying the functionality and usability of the graphic elements of an application.

(ex) buttons, menus, icons etc., hyperlinks, colors etc.
usability testing: checks how easily the user can understand and operate the application.

functional testing: behavior, functionality or how the system should work is checked.

① Object prop Testing: checks the functionality of visual elements and their behavior.

(ex) submit button is displayed / clickable()

② database testing: ensures the system completely interacts with the database

(ex) user details stored and retrieved when logged in & fine
error handling testing: checks the app throws appropriate error message when needed

(ex) when input is not given in mandatory field must give an error message.

③ Calculation testing: testing all the numeric accuracy of the operations and result is correct or not

(ex) total price of items in a site is correctly calculated

Link testing: checks the appropriate functionality of the links.

- (ex) after clicking link it leads to the correct link address.

Cookies and session testing :- checks the session-related function such as log-in and re-entering details after ending a session.

- (ex) log in to account, close browser, re-open and confirming the user is still logged in due to handing of sessions and cookies.

functional Testing non-functional testing

- i) checks the software is functioning as intended and meet the requirement's
- i) checks how software works under diff conditions and aspects beyond requirements.
- ii) ensures the features are working correctly on individual basis
- ii) checks how well the software works like speed, security, user experience

Regression testing :- testing the application is working properly even after adding the new build and it doesn't impact the existing functionality of a software.

Re-testing :- testing the functionality of the system after fixing every new bug from developer and reducing the fixed bugs so it doesn't affect the functionality of the system.

In summary :- Regression testing checks the bug issue is fixed and resolved. doesn't impact negative existing functions. Re-testing checks that fixed bug hasn't caused any new issues in the application.

Smoke testing : [Build Verification]

:- very basic functionality of the software is checked. like visibility of client, enabling of screens etc. ex) app is installed? Once installed screens are visible?

Sanity testing :-

Checking the normal functionalities like login
ex) clicking sign up redirects to proper sign up form.
ex) clicking sign up doesn't direct to sign in page etc.

Exploratory Testing

:- when no documt of req is given, test the software with all the possible scenarios. done only when there is no req given. testers have knowledge about application.

AdHoc Testing :-

:- testing randomly with no-test cases and no-req document, only aim to break the system. done by test engineer's. testers have knowledge of application.

Monkey Testing :-

Randomly testing the software with no test cases and req doc, with only aim to break the system. tester donot have any knowledge on the application.

End-to-End Testing :-

Testing that evaluates the entire application flow from start to finish. ensure that they work seamlessly.

Globalization: checks if the app can function seamlessly function in diff regions, cultures, accommodating various languages, date formats and currencies etc.

Localization: Here's the app's adapted to specific localities and cultural preferences (or) regional requirements. and regional language

N A T C C
T P T C D
E S E
T

Software Testing Life-cycle

Req Analysis: ✓

Test Case: step by step validation to check the validation of the system

Test planning: ✓

Test scenario :- A Possible area to be tested. (what to test)

Test case development: ✓

RTM: req Traceability Matrix is a document that maps all the test-cases and checks the requirements are covered.

Environment set-up: (OA Environment)

Test Execution: ✓

Test Cycle Closure: ✓

Defect :- Deviation from expected functionality of System.

Defect Severity

:- blocker [ex app crashed]

:- critical [ex basic functionality is not working]

:- major [ex unexpected behavior but app's functional]

:- minor [ex won't cause any breakdown in system]

Defect Priority

P₁ [High] Immediate resolving of issue req

P₂ [Medium] can wait until new build

P₃ [Low] can fix if later release.

Agile :- Iterative and Incremental approach.. agile allows the process to repeat again and again also allows the new build's on existing software.

- Small build's will be released in this process and releases will be very fast.

Scrum : is a agile frame-work to build software
↳ based on agile Principles.

Terminology

- ① User-story :- A module in Software
- ② Epic :- Collection of user-stories
- ③ sprint :- Period of time to finish user-stories.
- ④ design input
Equivalence class partition. $\rightarrow 0-100 - (50)$ (Pass)
- boundary value analysis. $\rightarrow \begin{matrix} 18 & 36 \\ 17 & 37 \end{matrix} + 1 (25)$ (Pass)

Decision table testing \rightarrow based on conditions

A.C	v	v	v
P.E	v	v	x
ES	v	x	x

state transition \rightarrow entered wrong password for 2 times
blank account

error guessing : \rightarrow checking for error by not giving any inputs

WebDriver: Comp of Selenium and a Java Interface which also work as an API

Setting up chrome driver

WebDriverManager.chromedriver().setup();

Launch url

driver.get("url");

tag name

driver.findElement(By.tagName("tagname"));

close browser

driver.close();

link text

driver.findElement(By.linkText("linktext"));

css-selector

→ tag id : ex # smallSearchInput

. → tag class : ex .searchbox-text

[] → tag Attribute : ex [name="q"]

[] → tag class attribute : ex searchbox-text [name="q"]

Alert

Alert alert = driver.switchTo().alert();

Alert.alert().get();

Alert.alert().sendKeys("string") ;

Alert.alert().accept(); close with OK button

Alert.alert().dismiss(); close with Cancel button

Auto-suggest dropdown

driver. findElements(By.xpath("")); → store in a list of webelements.

```
for (int i=0; i < list.size(); i++) {
```

```
    String s = list.get(i).getText();
```

```
    if (s.equals("option")) {
```

```
        list.get(i).click();
    } break;
```

headless-testing: Testing the process without running any UI by running the test cases in the backend

```
chromeoptions option = new ChromeOptions();
option.setHeadless(true);
```

```
webdriver driver = WebdriverManager.ChromeDriver();
Capabilities option = option.create();
```

Screen shot

- ① Take Screenshot ts = (TakesScreenshot) driver → file screenshot
- ② WebElement ele = driver.findElement(By.xpath("")); → get element
- file src = ts.getScreenshotAs(OutputType.FILE);
- file Trg = new File("path"); → save file
- FileUtils.copyFile(src, trg);

get methods

get(url);
 get(title);
 get(currentUrl());
 getPageSource();
 getWindowHandle();

Conditional Concepts

is displayed()
 is enabled()
 is selected()

checkbox

driver.findElement(By.xpath("//input")).
 click();

dropdown with no-select

webElement ele = driver.findElement(
 By.name("opt1"));

List<WebElement> list = driver.findElements(By.xpath("//ul/li"));

```
for (int i=0; i<list.size(); i++) {
    String s = list.get(i).getText();
    if (s.equals("Option1"))
        list.get(i).click();
}
```

mouse Action

Action Act = new Actions(driver);

Act.ContextClick(webElement).Perform() → rightclick
 Act.DoubleClick(webElement).Perform() → doubleclick
 Act.DragAndDrop(src, trg).Perform()

Act.MoveToElement(element).MoveToElement(element).Perform() → hover
 WebElement element = driver.findElement(By.xpath("//input[@value='OK']"));
 Act.dragAndDropBy(element, 100, 60).Perform();

Creating action and
storing in variable

Action Act = new Action(driver)

Act.click()

Action Act = act;

Act.click();

Context click (below);

build;

Act.Perform();

hover;

Move to element;

drag and drop;

double click;

right click;

click();

move to element();

perform();

dragAndDropBy();

dragAndDrop();

doubleClick();

contextClick();

moveToElement();

perform();

Open a New Tab and Window

driver.switchTo().newWindow(WindowType.TAB);

driver.switchTo().newWindow(WindowType.Window);

Explicit wait

WebDriver wait myWait = new WebDriverWait(driver, Duration.ofSeconds(10));

WebElement ele = myWait.until(ExpectedConditions.

visibilityOfElementLocated(By.xpath("//input[@value='John']")));

Implicit wait

driver.manage().timeouts().implicitlyWait

(Duration.ofSeconds(10));

Navigation

navigate.to();

Navigate().back();

navigate().forward();

navigate().refresh();

JavaScript executor [mainly used during element click Interception]

javascriptExecutor.is = (javascriptExecutor) driver.

js.executeScript("arguments[0].click();", element);

js.executeScript("arguments[0].setAttribute('value','John');", inputbox);

Key board actions

Actions Act = new Action (deiv); ctrl+a

act. KeyDown (keys. Control). sendKeys("a"). KeyUp (keys. ctrl+c) perform
control) perform

act. KeyDown (keys. CONTROL). sendKeys("c"). KeyUp (keys. ctrl+c) perform
control) perform

act. KeyDown (keys. TAB). KeyUp (keys. TAB) perform
(01)

act. sendKeys (TAB). perform; X

/* @attribute = 'value'
*/
/* [Contains @name, value] */

scroll

certain pixel

js.createScript ("window.scrollBy(0,5000);");
scroll till element is present

js.createScript ("arguments[0].scrollIntoView(); element");
end of page

js.createScript ("window.scrollBy (0, document.body.scrollHeight);");

to initial position

js.createScript ("window.scrollBy (0, -document.body.scrollHeight);");

Static table

To read any value

L-19

for (int i=0; i < row; i++)
 {
 for (int j=0; j < col; j++)
 {
 string s = driver.findElement(By.xpath("//table/tr[" + i + "]/td[" + j + "]")).getattribute("value");
 System.out.println(s);
 }
 }

select row by xpath by (f9)
 row = ele.size()

select col by xpath th
 col = ele.size()

string s = driver.findElement(By.xpath("//table/tr[" + i + "]/td[" + j + "]")).getattribute("value");

- - - // tr ["+r+"] // td ["+j+"]));

getattribute();

Pagination table

for (int i=0; i <= 10; i++)

{
 if (total pages > 1)

{
 WebElement page = driver.findElement(By.xpath("//a[@class='next']"));

- - - // Text() = +pt+);

Page.click();

xpath: it's a xml path. It's a Syntax (or) language for finding any element using xml path expression.

select
from.
join.
where
orderby.
sort
limit

- ① Inserting Data in to table
- ② update col name / col of table
- ③ Delete.
- ④ Drop
- ⑤ UNION
- ⑥ UNION ALL
- ⑦ JOIN, LEFT JOIN.
- ⑧ Primary key, foreign key, like
- ⑨ SQL optimization.

ON
BETWEEN
IN
NOT IN

AND

disadv

① ERP? why do we use ERP

② UFT, Tosca → automation tool

high maintenance.

③ management tool → Jira.

lack of reporting.

④ oracle peoplesoft

limited to web applic.

ms dynamics

cannot test image

work day

scip

People soft HCM.

① Pay role, absence management
recruitment, time and labor.

finance.

ERP



oracle

SAP

~~mid market~~

~~wonders~~

~~microsoft~~

~~infor~~

PeopleSoft + SAP

Core HRMS

work day
ms dynamics

Human Capital management

Finance
Cash management

fixed assets

payroll

expenses.

Eff DATE

<= Current Date

= Current Row

Eff DATE < current row

= History Row

Eff Date > current Date = Future Row

benefits

④ initial, inverse, account inverse

⇒ including benefits in more than one benefit programs.

⇒ Assign and entail the employees to benefit plans.

⇒ calculate benefit-related deduction.

absence management

Apply leave.

Cancel leave

View absence balance

View absence history

Edit the leave.

raise comp-off request * leave type cannot be changed.

time and labour

- track time for the employees in Office (per day)
- track time for public employee (8-5)
- Payroll based on the time.

ERP :- Enterprise resource planning helps run business. Supporting the HR, Finance, manufacturing all these processes are integrated in the ERP.

why to use ERP

- Increase productivity.
- Increases customer satisfaction.
- Simplify product life planning.
- Increase efficiency and stability.