Regression Testing

Company is releasing software in multiple versions

CHANGES

Addition of new func
Deletion of old func
Modification of old func

Testing the same old functionalities again and again across multiple versions of the software is only called as Regression Testing(Full Regression Testing)

Because when changes are done to the software, there is a chance that old functionalities can get affected

Regression testing can be done by taking the help of Automation Tools

Sometimes, when less changes OR no changes are done to the software(just minute big fixes), testing is called Sanity Testing

Testing the functionality completely that was having problem previously is only called as Sanity Testing.

Inbox ==> 30 T.C ==> 10 T.C Failed ==> Sanity Testing(all 30)
Inbox ==> 30 T.C ==> 10 T.C Failed ==> Retesting(check 10) ==> Not Recommended

Developer ==> Software ==> Handed to the Test Engineer ==>
Basic Checking ==> Smoke Testing
==> software is opened or not
==> Navigation working properly or not
==> Check basic func(login), Inbox , ...

==> Deep Testing

Usability Testing
Checking whether the software us use friendly or not

1) Good U.I
2) Faster
3) Less space
4) Multiple Languages

Accessibility Testing

Testing the software whether it can be used by the disabled people or not.

1) Voice support
2) Captions support

What's app message
Check that message == SEND == Again we check that message

Req, Design, Code == SOFTWARE == Testing

Static Testing

P.O / Business Analyst
Architects
Developers

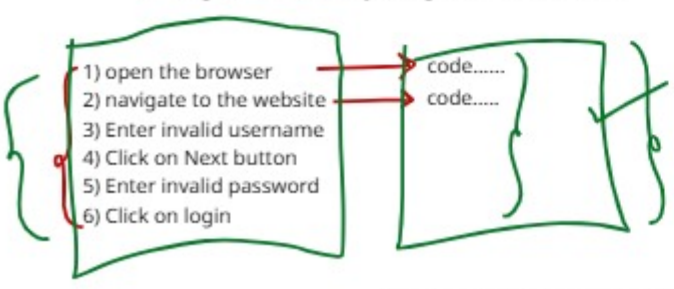
Dynamic Testing(Component Testing, I.T, S.T, U.T, A.T, R.T, ...)
Test Engineers

Testers will also come in Static Testing

T.E ==> Write the Test Cases by referring S.R.S document and keep it ready

Automation Testing

Testing the software by using Automation Tools.



Playwright ==> Javascript code / Typescript code

Selenium, Playwright ==> Automation Tools ==> Libraries
Selenium ==> Library
Playwright ==> Library

College ==> Library ==> contains books, tables, ...
Playwright ==> Library ==> contains methods, classes, variables, ...

If we want to install playwright library, then we use npm command

npm command will be installed if we install Node.js

cmd ==> Node -v

npm install playwright@latest

npm init playwright@latest ==> Install the playwright, it also cerated a new playwright project(means it created some files and folders).

npm init playwright@latest

some options
JS / TS ==> JS
Do you want to install browsers

Playwright has it own browsers, in those browsers only the test cases should run(firefox, chromium and webkit) ==> Playwright sepcific browsers

Computer(Safari) ==> Yes, ...

npx playwright install ==> to install browsers

playwright.config.js ==> Configuration File
This file(js file) should have all the rwquired things to control playwright.

test cases ==> JS file ==> folder

data ==> first.js ==> test cases

package.json ==> contains all the libraries that your project uses

1) Open the browser, playwright automatically open the browser

name: "chromium" →


```
class Page
{
   ---------
   ---------
   ---------
}

new Page()
```

fixtures

```
{
  browser:
  browsercontext:
  page: logic to open the tab

}

{ page }
```

npx playwright test ==> config file ==> tests ==> spec ==> run that test case

By default playwright runs the test cases without opening the browser physically ==> headed mode

locators
locators help playwright to locate/identify web elements

1) id locator
2) class locator
3) name locator
tagname[name="------"]
tagname[id="email"]

button[name='login']
text='Invalid username or password'