

Explanation of Framework

- Framework is a set of rules and guidelines or best practice to be followed while automating any application.
- In order to execute the test cases with multiple inputs on test data we use excel and property file, so we call our framework as Data Driven framework.
- In order to avoid writing repetitive states again and again we use lot of reusable methods so we call our framework as Method Driven framework.
- Since we maintain our framework module wise, so we also call our framework as Modular Driven framework.
- Since it is the combination of two or three framework hence, we call our framework as Hybrid Driven Framework.
- In the beginning of the execution 1st, it executes the Base class which is present in the Generic package, which contains all the configuration methods like @BeforeTest, @BeforeMethod, @AfterMethod, @AfterTest.
- 1st it executes @BeforeTest which contains the code for opening the browser then it will execute Login code which is present in @BeforeMethod
- Then it will execute @Test method where actual test scripts are written in the test script package. while executing the test script it will take the test data from the excel file with the help of Apache Poi jar and performs action on the GUI application by calling necessary method in the POM classes.
- Once it executes the test method it will execute the Logout code which is present in the @AfterMethod annotation.
- Like this it will execute all the test cases one after the other with the help of batch runner. (TestNG.xml suite file)
- After the execution of all the test cases it will close the browser.
- Since we are using TestNG it will automatically generate the default HTML report in the Test-Output folder which contains number of test case passed, failed, skipped etc...
- Since we have implemented the listener feature of TestNG it will automatically take the screenshot of failed test cases in the screenshot folder.

Q. Can you explain the Page Object Model (POM)?

The Page Object Model (POM) is a design pattern in Selenium that creates an object repository for web elements. It improves test maintenance and reduces code duplication by creating separate classes for each page.

Q. What is Data-Driven Testing?

Testing the application with multiple inputs or test data which is keeping external resource file like Property File, Excel File and Database etc is called as Data Driven Testing.

Advantage of Data-Driven Testing:-

- Reusability of common data and test script data.
- Maintenance of data in excel or external resource file is easier.
- Modification of data in excel or external file is easier.
- We can test the application with huge volume of data.
- Test data can be created explicitly before the test script execution.

Q. What is Maven..?

- Maven is a popular build automation and project management tool used for Java projects.
- It supports a wide range of plugins for various tasks, such as compiling code, running tests, generating reports, and deploying applications.
- The pom.xml file is the heart of a Maven project, containing information about the project.

Q. What is TestNG (Test Next Generation)?

TestNG is a unit testing framework tool which is mainly used for Batch execution.

It is a plugin or 3rd party tool for Eclipse which is inspired by Junit with some additional features.

Advantage or additional features of TestNG

1. Batch Execution
2. Parallel Execution (Cross-Browser Testing)
3. Group Execution
4. Generate the reports (HTML report)
5. Run only failed Test Scripts
6. Additional annotation
7. Listener features

CI/CD (e.g., Azure Pipelines, Jenkins)

- CI/CD automates the process of testing, building, and deploying applications. This reduces manual effort and minimizes human error.
- With CI, developers receive quick feedback on their code changes. Automated tests run with each commit, helping catch issues early.
- CI/CD pipelines ensure that the deployment process is consistent across environments (development, testing, and production).
- Automated pipelines allow teams to deploy changes more frequently, enabling quicker delivery of features.
- CI/CD encourages team collaboration by integrating work from different developers regularly, making it easier to manage changes and resolve conflicts.

Git:- Version Control

- Git allows developers to track changes in the codebase.
- This helps manage different versions of software and facilitates rollbacks if needed.
- It enables the creation of branches, allowing developers to work on features or bug fixes.
- It supports collaboration among team members by providing a platform for sharing code and managing contributions effectively.
- It maintains a detailed history of changes, making it easier to understand the evolution of the codebase.
- Git integrates seamlessly with CI/CD tools, allowing automated builds and tests to be triggered on code changes.

Q. Explain how you would **integrate Bitbucket** into your automation testing framework.

- **Version Control:** Bitbucket is used to store test scripts in a repository, enabling version control and collaboration with the team.
- **Branching and Merging:** Developers can create branches for their test scripts, and once the changes are verified, the branch can be merged back into the main branch.

- **CI/CD Integration:** Bitbucket can be integrated with tools like Jenkins to trigger automated tests whenever changes are pushed to the repository.

Benefits of Agile:

- **Flexibility:** Adapts quickly to changing requirements and market needs.
- **Incremental Delivery:** Delivers functional software in small, frequent increments.
- **Collaboration:** Enhances teamwork through regular communication and cross-functional collaboration.
- **Improved Quality:** Continuous testing and feedback help catch issues early, leading to higher-quality products.
- Engages customers regularly to ensure the product meets their needs.
- **Transparency:** we can openly communicate about progress and challenges, improving stakeholder visibility.
- It encourages self-organizing teams, ownership and motivation.
- Identifies and addresses risks early in the process.
- **Value Focus:** Prioritizes delivering features that provide the most customer value.

Q. How do you manage test dependencies in TestNG?

- TestNG provides the `dependsOnMethods` and `dependsOnGroups` attributes to manage test dependencies.

```
@Test(dependsOnMethods = {"testLogin"})
public void testDashboard() {
    // Code to test dashboard functionality after login
}
```

This ensures that testDashboard() is executed only after the successful execution of testLogin().

Q. What is your **approach to debugging a failing automation script?**

- **Check for errors:** Identify the line of code causing the failure using logs or screenshots.
- **Verify locators:** Ensure that the web element locators (IDs, XPath, CSS selectors) are correct and haven't changed.
- **Handle synchronization issues:** Use waits to manage timing issues between Selenium and the web page.
- **Isolate the issue:** Run the test script independently to pinpoint the cause.

What is Framework

Framework is a set of rules or guideline to be followed while automating any application.

1. BDD Framework (Cucumber) Behavior Driven Development

Framework In framework there are three stages

- i. Framework Designing/Development
- ii. Framework Implementation
- iii. Framework Execution

Defect Life Cycle (Bug Life Cycle)

- The Defect Life Cycle, also known as the Bug Life Cycle, is the process through which a defect or bug passes during its lifetime, from initial identification to final closure.
- It helps track the current status and progress of the bug and ensures that it gets resolved efficiently.

Stages of the Defect Life Cycle

1. Identify the defect
2. Report the defect
3. Assign the defect
4. Fix the defect
5. Test the defect fix
6. Close the defect (if the fix is successful)
7. Reopen if necessary.

- **New-:** When a tester identifies a defect for the first time, it is assigned the status "New" and is reported in the defect tracking tool.
- **Assigned-:** After reviewing the defect, the team lead or manager assigns the defect to a developer or a team for fixing.
- **Open-:** Once the developer begins working on the defect, its status is updated to "Open." This indicates that the defect is being analyzed and worked on.
- **Fixed-:** After the developer has applied a fix, the defect is marked as "Fixed." At this stage, the fix is implemented in the code, but it hasn't been verified yet.
- **Pending Retest-:** The defect is now ready for testing, but the tester has not yet started retesting the fixed defect. This stage is often called "Pending Retest" or "Ready for Retest."
- **Retest-:** The tester retests the application to verify whether the defect is indeed fixed. If it passes the test, the defect moves to the next stage.
- **Verified-:** If the defect is fixed and the functionality works as expected during retesting, it is marked as "Verified."

- **Closed-:** Once the tester confirms that the defect is successfully fixed and does not occur again, it is marked as "Closed."
- **Reopened-:** If the defect persists after being marked as "Fixed" or "Closed," the tester reopens the defect, and it re-enters the life cycle. The developer must then investigate the issue again.
- **Deferred-:** A defect may be marked as "Deferred" if it is decided that the bug will not be fixed in the current release but will be fixed in future releases. This may be due to lower priority or time constraints.
- **Rejected-:** The defect may be marked as "Rejected" if the developer or lead determines that it is not a valid defect. This could happen if:
 - The defect is not reproducible.
 - It is an expected behavior of the system.
 - The defect is a duplicate of another defect already reported.