**TRANSLATION**

**Agile** is the methodology used for project.

**Product backlog -** prioritised list of user stories.

In agile there is 2 cycles

1. Development cycle.
2. Test / Execution cycle.
3. What are the major challenges you have faced during automation or have you raised any concern in Retrospective meeting?
4. Frequent UI changes which affects the automation scripts.
5. **Synchronization issue –** Lack of development standards. Despite using all the inbuilt waits, the scripts were still failing and hence I had to use **custom wait**.
6. Unplanned leave by the engineers.
7. Unrealistic timeline or deadline taken by the senior without prior notification.
8. Improper peer review of automation scripts.
9. Lack of automation and coding standards.
10. Lack of information about the framework.
11. Suite execution was taking more time and manager asked to reduce time.
    1. Increase the number of threads and also increase the number of selenium grid.
12. What exactly do you review in automation?

**Automation coding standards**

1. Data should not be hard coded.
2. Locator should not be hard coded.
3. Xpath should always be related.
4. Every landing page should have proper comments.
5. Expected result should have proper assertions.
6. A script name should be same as manual test case name which should be mapped in Jira.
7. Automation script coverage(Each step should be automated).
8. Automation script should be developed by taking help of all the reusable components from the framework.
9. I will give you one feature or user story for testing. Whats your approach towards testing?
10. Nowadays everybody follow agile methodology keeping that in mind I’m going to explain testing approach.
    1. Whenever I get user story from the stakeholder, very first initiation will be
       * 1. Go through.
         2. Walkthrough.
         3. Get clarity on requirements from product owner or stakeholder.
         4. Understand requirement thoroughly.
11. All above mentioned activity happens in sprinting or product grooming meeting.
12. Before sprint planning meeting we should come up with effort and estimation plan for the allocated user story.
13. I’m going to submit effort and estimation chart to eam lead, Team lead is going to submit back to stakeholder in sprint planning meeting.
14. Get an approval from stakeholder and create sprint backlog.
15. Entire sprint will be breakdown into two cycle.
    1. Development cycle.
    2. Testing cycle.
    3. In development cycle, I will start writing a manual test case.
    4. Send test cases for review, get it approved, finally push test cases back to test manager tool(Jira).
    5. In testing cycle I will be involved in execution of manual test cases hand raising defect for every new build.
    6. I will also be involved in daily standup meetings and bug triage.
    7. At the end of the sprint, a stable build will be installed in the staging environment.
16. In the Sprint review meeting, developers will walk through the application with the stakeholder, while the testing team will walk through the execution report.
17. After the Sprint review meeting, the application will be deployed to the UAT environment.
18. Finally, we’ll be attending a retrospective meeting to understand what went well and what didn’t.

**As an automation engineer**

1. Collect regression test cases from the manual testing team.
2. Understand the go through and walk through the test cases and understand the functionality.
3. Come up with effort and execution plan for automation development.
   1. Number of steps.
   2. Complexity of the application.
   3. Reusability.
   4. Automation engineer experience level.

|  |  |  |  |
| --- | --- | --- | --- |
| No. of steps | Simple | Medium Complex | Complex |
| 1 to 5 | 8 / day | 3 / day | 2 / day |
| 1 to 10 | 4 / day | 2 / day | 1 / day |
| 1 to 20 | 2 / day | 1 / day | 0.5 / day |

1. Segregate automatable and non automatable test cases.
2. Submit estimation in sprint planning meeting.
3. Get approval from the stakeholders.
4. In the sprint development cycle, start automating the regression test cases.
5. Sending for peer review.
6. Fix the review comments and push the final code to the master branch.
7. Configure the Jenkins for regression testing for every new build.
8. In sprint execution cycle I will be involved in going through Jenkin reports and debug the script.
9. Create the regression defects in Jira.
10. Create a regression defect report and submit to the stakeholders before the sprint review meeting.
11. Discuss what went well and what didn’t.
12. Will attend sprint review meeting and understand the upcoming UI changes.
13. Will attend the sprint retrospective meeting and understand what went well and what didn’t.

**Coding standards**

1. Variables are class or package name should be specified.
2. What we review is logic of the program.
3. Every class should have an author.
4. Every method should have a document.
5. Product is released to the production, customer got a bug(Bug leakage).
6. For every bug leakage in the live environment will have 24 hours deadline within which the bug has to be fixed & provide hot fix to the UAT or the live environment.
7. Steps to follow during hot-fix.
8. Stop current sprint activity.
9. Development, Testing, Devops team will get into the War room.
10. The development team will fix the defect, Manual testing team will be involved in executing automatable & non automatable test cases, Automation team will be involved in executing regression test suites.
11. Within 24 hours we should provide execution report & hot fix.

We have 1000 test scripts & all the scripts got failed, what might be the reason?

1. Environment test data issue(web URL,DB URL, invalid username or password or base path or Client Id client secret or APKs).
2. Compatibility issue.
3. Ex : Browser version not matching with Selenium. Pytest not matching with Maven.
4. There might be a bug in the root itself.
5. The application might be broken because one of the services might not be running.(Web server or DB server)

Out of 1000 test scripts 50% of the scripts got failed. What might be the reason or what’s your approach?

* 1. We'll look into high level repot & check which results got failed.
  2. We'll look into low level report & check which line is being failed.
  3. Will look into the exception & line number.
  4. Go to the particular script, put a breakpoint & run it in debug mode & find the root cause of the issue.
  5. If the script is failed because of the product issue then raise the defect or else update the script to the master repository.
  6. You found a critical defect in the application but your manager says make it as minor & release, as there is no much time for fixing the defect.
  7. As a test engineer I should follow the manager's instructions without deviating from the hierarchy & at the same time I'll list out all the problems because of the changes & provide a suggestion not to change the status(Critical to minor).

Explain your application technology stack.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Front-end | Middleware | Architecture | Host | Database |
| HTML – 5 | Java + spring boot | 2-tier | On premises | MySql |
| Angular JS | Python + Django | 3-tier | Cloud   1. Amazon AWS 2. MS Azure 3. IBM Cloud 4. Google Cloud Platform | MongoDB |
| React JS | JavaScript + nodeJS | N-tier | Oracle Sql |

Explain your Automation technology stack.

1. For Web Automation we used
   1. Web driver Java
   2. Web driver IO.
2. For API Automation : RestAssured.
3. Database Automation : JDBC.
4. Mobile Automation : Appium.
5. Source management tool : Git.
6. Unit testing tool : TestNG, Pytest.
7. Build management tool : Maven(Java), pip(python).
8. CI / CD tools : Jenkins, Bitbucket, Git Bash , Microsoft Azure, AWS.

How often you will get build / How you will get new build?

1. If the build process is not automated :
   1. The developer will commit the master code into the Git Repository.
   2. The devops engineer will get the code from the build repo & create a executable file.
   3. Upload the executable file into the build repo.
   4. Install the build manually in the test environment.
   5. Write an email to the testing team & the development team about the arrival of the new build.
2. If the build process is automated all the above steps are automatically done by Jenkins.
3. In real time we will have a dedicated build in the repo storage where we maintain all the versions of the application.

What is the entry criteria for automation

1. Stable build.
2. Skilled Automation resource.
3. Automation plan, Effort & Estimation plan should be ready.
4. Updated manual regression test cases.
5. The framework should be ready.
6. Framework user guide document.

What is the exit criteria for automation

1. Execution report with no critical defects.
2. Updated framework along with the user guide document should be available in Git.

What is the Acceptance criteria?

1. Whenever we deliver a stable build into the UAT server If the Stakeholder does not come back within 15 days that means the build is accepted.

What are the automation deliverables for Stakeholder?

1. Effort & Estimation plan.
2. Automation plan for each sprint.
3. Weekly progress report.
4. Execution report.
5. Working framework along with the necessary documents.

What process did you follow in your previous project?

**Agile :** It is a process in which we develop the project in an iterative & incremental way, where development& testing happens parallelly.

Explain automation process followed in your previous

Automation approach always follow 4 phases

1. Design & planning phase.
2. Test development phase.
3. Execution phase.
4. Maintenance phase.
5. **Design & planning phase :**
   1. We will collect manual test case from manual team.
   2. Segregate automatable & non automatable test case.
   3. Come up with Effort & Estimation.
   4. Design the base layer framework.
   5. Configure the automation environment.
   6. Add all the dependencies.
6. **Test development phase :**
7. Get framework from Git master branch.
8. Create elements & library.
9. Create test data.
10. Convert manual test case into test script using framework components.
11. Create a pull request for review.
12. Incorporate or fix review comments.
13. Merge final code to master branch.
14. **Execution phase :**
15. For every new build received from development environment to testing environment.
16. Jenkin was configured to execute regression suite.
17. Post execution Jenkin send us the execution report.
18. Verify the execution report & find the root cause of the issue.
19. Raise a regression defect in Jira if product issue or else update the script.
20. Above steps will continue for every new build arrived.
21. **Maintenance phase :**

In agile process requirement changes are always welcome, for every addition deletion modification of functionality we have to maintain the test scripts in Git repository.

Manual Test case

Master branch

Requirement Gathering

Git Automation Repository

Segregate Automation & automation test cases

Pull Framework

Pull Request + fix

commit

fix

Re Execute

Defect

New build

commit

Plan Estimate

Local branch

DMT

Maintenance

Execute

Report Analysis

Debug

Devops Test script

Build library & element

Jenkins

Explain automation life cycle.

Requirement gathering

Tool Selection

Automation Strategy

POC(Proof of Concept

Automation Plan

Segregate Automation / Non- Automation test case

Framework Development

Test script Development

Review

Test Execution

Report Verification

Bug tracking

Maintain test script Git repository

Automation life cycle typically consists of 5 phases :

1. Proposal & POC phase.
2. Design & planning phase.
3. Script development phase.
4. Execution phase.
5. Maintenance phase.
6. Proposal & POC phase
   1. In this phase participants are Automation Architects, Automation Center of Excellence (ACOE) team & managers.
   2. In this stage managers are going to create quotation & automation strategy based on stakeholder & request for proposal(RFP).
      1. Quotation contains price information.
      2. Automation strategy is an high level documentation. It contains scope, objective, automation approach, framework, tools & technology.
7. Automation strategy is getting created before getting the project.
   * + 1. In POC stage ACOE team collect 5 – 10 challenging scenario from the stakeholder to create a POC.
       2. Before development of POC, automation architect is going to select the tool based on application support.
       3. E-Commerce application – WebDriver.
       4. Google Maps – Sikuli.
       5. SAP application – QTP (QuickTest Professional) & Test complete.
       6. POC will help us to get the project from the stakeholder & also check the automation tool feasibility, availability of skilled resources & challenges associated with the tool.
       7. Once the POC is developed, Architect is going to give demo to the stakeholder to get the project. If POC is approved then we get into planning & designing phase.

Explain the (parameters) content of Automation plan

1. Scope .
2. Objective.
3. Milestone.
4. Automation approach.
5. Automation tools & technology.
6. Automation framework.
7. Execution reports.
8. Review process.
9. Entry & exit criteria.
10. Acceptance criteria.
11. Deliverables.
12. Engineer’s roles & responsibility.
13. Risk & mitigation plan.

Write few complicated scenario which you have automated in your previous project.

Explain 1 complicated scenario which you automated in your previous project.

Scenario 1:

Fetching data from APIs & generating reports

Explain few critical defects which is found by your automation scripts.

Explain automation challenges faced in your previous project.

Explain your daily activity.

Briefly introduce yourself.

Explain roles & responsibility in your previous project.

1. Manual Activity
   1. We used to get manual test cases from Jira through test management tool.
   2. Our team lead is going to allocate test cases to us.
   3. Once test case is allocated, I was involved in executing them manually to understand the functionality.
   4. If there is any deviation in the test case, get a clarity from the manual team.
   5. Check whether test case is automatable or not.
2. Plan & Estimation activity :
   1. In my previous project estimation is always done by team lead, but I was a part of estimation whatever the user story which was allocated to me.
   2. Submit Effort & Estimation to stakeholder in sprint planning meeting & get an approval.
3. Script development activity:
   1. Get the latest framework from the Github
   2. Involved in creation or updation form classes.
   3. Involved in creation of generic library.
   4. Involved in creation of business or workflow library.
   5. Involved in creation of test data.
   6. Involved in conversion of manual test cases to automation test script with the help of framework component.
4. Review Activity:
   1. Involved in creation of pull request.
   2. Involved in review test script with respect to automation coding standards or python or java standards.
   3. Involved in creation of merge request.
5. Aggregation & debugging activity :
   1. In my previous project Jenkin was confirgured to execute smoke & regression suite for every new build arrival.
   2. For every new build, Jenkin used to send us the execution report.
   3. Involved in verifying the execution report & find the root cause of the failure.
   4. Involved in debugging the scripts in debug mode.
   5. Involved in updation of the scripts.
   6. Involved in raising regression defects.
6. Agile activity :
   1. Involved in all the stub meetings.
   2. Partially involved in Effort & Estimation.
   3. Involved in development of the framework in sprint one or before starting automation project.
7. Jira activity :
   1. Before starting sprint we should always get the user story & manual test cases from the Jira.
   2. Post execution of the automation script I was involved in updating the execution status in Jira.
   3. Involved in raising & tracking regression defects in Jira.
8. Git activity :
   1. Good knowledge on git architecture.
   2. Good understanding on Git commands.
   3. Involved in handling the Git conflicts.
   4. Involved in creation of pull requests.
   5. Good understanding on local & master branches.
9. Jenkin activity :
   1. Good knowledge on CI / CD.
   2. Configured the execution with framework available in Git.
   3. Good understanding of Jenkin pipeline.
   4. Good knowledge on Jenkin parameters.
   5. Good knowledge on Jenkin execution types.

Explain your daily activities

1. Task Allocation
2. Get the latest framework from Git
3. Create form & reusable libraries.
4. Create test data.
5. Develop test script.
6. Agile meetings.
7. Daily standup meeting.
8. Daily closures meeting.
9. Review requests.
   1. Check automation coding standards.
   2. Check Java or Python coding standards.
10. Execution report.
    1. Get the email from Jenkins, go through high level report, go through low level report, check for exception & line number, put a breakpoint & debug the script.
    2. Find the root cause of the issue.
    3. Raise the defect or fix the script.
    4. Generate execution report.

|  |  |  |  |
| --- | --- | --- | --- |
|  | Simple | Medium Complex | High Complex |
| Team Size | 1 Manager  1 Dev team lead – 7 developers  1 Testing lead – MT – 3 AT – 1 | Senior Manager – 1  Dev team lead – 1  Testing team lead -1  Dev – 10, MT – 7, AT - 3 | CTO 🡪 Architect 🡪Project Manager 🡪Dev team & Testing team ….etc |
| Test cases | 600 – 1200 cases | 3k – 7k cases | < 1000 cases |
| Sprint duration | 3 – 4 weeks | 3 – 4 weeks | 3 – 4 months |
| Test cases in each sprint | +50 | +250 | +100 |

|  |  |  |  |
| --- | --- | --- | --- |
| Steps | Simple | Medium Complex | High Complex |
| 1 - 5 | 5 – 6 | 3 – 4 | 1 - 2 |
| 1 - 10 | 3 - 4 | 2 - 3 | 1 |
| 1 - 20 | 2 | 1 | 0.5 |

.