

**Imagine the following situation. You need to establish a QA process in a cross-functional team. The team builds a front-end application using REST APIs.**

**1. Where would you start? What would be your first steps?**

I would start with 2 main points to look into

- i. **Requirements and Client needs**
  - a. Gather the requirements /features to be developed in the form of SRS (Software requirements Specifications) and go through the requirements and clarify outstanding queries/doubts with Product Manager/immediate stakeholders and team members and finalize the correct requirements
  - b. Get the assumptions clarified or if they are safe to be assumed and taken ahead
  - c. Understand the business rules and the scope for testing
  - d. Get all platforms it should be implemented/tested on like OS, compatibility with browsers and other hardware devices
- ii. **Team Capability and resources/tools at hand**
  - a. Skilled team members available to build the application
  - b. Does it require re-skill or Knowledge transfer and the time required for it
  - c. What all tools do we have for front end, back end, APIs, test Management, Project Management, bug reporting etc.
  - d. What are the risks and if required to come up with contingency plan

Considering all these factors along with the project being Agile, I would come up with a **Test Strategy document** and **Test Plan** and present it to all stakeholders and take approval from all in order to proceed with next steps

**2. Which process would you establish around testing new functionality? How would you want the features to be tested?**

**I would recommend the below steps to test the new functionality**

- i. Take each functionality to be tested, and break it into features to be tested
- ii. Come up with different Use cases/scenarios under each feature
- iii. For each feature, try to think what all criteria is applicable and perform those tests. Example: negative tests, boundary value analysis, state transitioning etc.
- iv. Mark against each feature what all functional tests like unit testing, integration testing, system testing, sanity testing, smoke testing can be carried out and come up with those scenarios
- v. In a similar way, check what all features needs non-functional testing like performance testing, load testing, security testing, compatibility testing, localization testing etc.
- vi. Ensure proper data setup is in place for each testing in environments(virtual)
- vii. Ensure Regression testing is carried out so that this new functionality doesn't affect the BAU of the existing functionality and business continuity is maintained.

### 3. Which tools would you suggest using to help your team with a daily work ?

Below are some of the tools which I have used in previous projects and would recommend the team to use them for daily work

i. **Jira:**

This tool serves multipurpose, we can track

- a. Bugs
- b. Issues
- c. Project Management/resource management using some Plugins

As the team would work in Agile model, Jira also helps in tracking productivity, velocity and being transparent to all stakeholders

- ii. **Trello:** For task Management. Can be used for personal and small team daily updates
- iii. **Confluence:** Can be used to store various documents related to project like SRS, Test Strategy, Test Plans, Client documents. It can be used for daily MOM and other team updates/communication
- iv. **Gherkin:** We can use Gherkin to write features/test scenarios as it is easily understandable and followed. Not only the QA and testers, but entire team (viz. developers, Product Manager) can contribute in writing the test cases.

### 4. If you would do a test automation which techniques or best practices would you use the application?

This depends on the skills/automation tools within the team and client/company preferences. Below are 2 options I could recommend for **front end application development** with REST APIs

- i. **Flask framework**
- ii. **AngularJS framework**

For Test automation, we need to write down which all features can be automated and for those we need to come up with the automation plan. Considering the above 2 options, we can go with below tools for automation of features deemed good for automation

- i. As **Selenium/Appium** can be used with any base code and both are open source so it would be better to go ahead with these tools.  
Write the test cases in **Gherkin** as it is easily understandable by all stakeholders and helps in automation
- ii. **Protractor framework for AngularJS hybrid applications** as it is a combination of Selenium WebDriver, NodeJS, Jasmine, and Cucumber. It is an end-to-end testing framework and we can do non-functional testing (performance testing, load testing etc.) as well.