**QA engineer: Technical challenge:**

**Basics:**

* **How do you deal with Flaky test?**

There are two ways to deal with the flaky tests.

One is to run a test, or even a whole test suite, multiple times. If you didn’t change any code between test runs and the test suite shows a different number of failed tests at every run, you can be sure that something has gone awry with your test.

The other way is to run them in a different order each time. If the tests fail when their order is changed, it’s a sign that we haven’t accounted for inter-test dependencies.

* **Let's suppose there is a test pipeline taking about 1 hour to finish, what would you do to decrease the time of it?**

**To reduce the time, I can perform the following actions:**

1. **Will remove the sleeps and triage tests.**

I will look for the for “sleeps” in test-ware to speed up tests. That are taking time and a sign. In order to reduce time and waiting for a long time. If the test is critical, there are methods for polling more frequently so that the test can restart sooner.

 Also I wilI look for low-value tests that I can run less frequently. The likelihood of failure is combined with the priority of a defect discovered by a failure to determine the value. We can run a test less frequently, say overnight or weekly, if it never fails or only finds low-priority defects

1. **I will run the test in parallel:**

Parallel testing is one of the most effective strategies to speed up tests. Attempting to remove one second from this test and three seconds from that test hasn’t had the same impact as running two tests concurrently rather than serially.

* **Imagine you have the possibility to ask software engineers to develop tools for you that will increase your productivity as full-stack QA, please describe them your requirements.**

**Requirements for a tool that will increase the productivity as full stack QA are as follows:**

* The tool should provide cross browser support.
* It should include the reporting section.

Reporting section is important. Need reports that tell us how our tests are performing, which failed tests need not be re-run, what features do not require extensive debugging.

* Reusable steps-There are many scenarios that have the same steps. So, it would be a challenge to manage and maintain them because every time there is a change in a frequently used step, the script needs to be updated every test that includes it. This would allow the QA automation to save a set of common steps and insert them into any test flow.
* Dynamic element locator this feature will allow to bring in more than one element locator. In case, a particular element is missing, the automation testing tool should be able to find an alternative element locator. This reduces the effort spent in maintaining huge scripts and results in more stable test scripts.
* **Imagine you have the possibility to ask software engineers to develop for you tools that will increase your productivity as full-stack QA, please describe them your requirements.**

Please Note that this question is same as above, it is repeated twice

**Requirements for a tool that will increase the productivity as full stack QA are as follows:**

* The tool must be cross browser support.
* It should include the reporting section.

Reporting section is important. Need reports that tell us how our tests are performing, which failed tests need not be re-run, what features do not require extensive debugging.

* Reusable steps-There are many scenarios that have the same steps. So, it would be a challenge to manage and maintain them because every time there is a change in a frequently used step, the script needs to be updated every test that includes it. This would allow the QA automation to save a set of common steps and insert them into any test flow.
* Dynamic element locator this feature will allow to bring in more than one element locator. In case, a particular element is missing, the automation testing tool should be able to find an alternative element locator. This reduces the effort spent in maintaining huge scripts and results in more stable test scripts.
* **Prepare a test plan for testing the login feature of your application, considering the following requirements:**

● The feature is implemented on iOS, Android and Web and is backend driven

● The mobile apps are native

● The login feature consists of a login form with email and password input and a login button

● After input correct credentials, you land in the main screen of your application

Your testing plan need to answer the following questions:

1. What are you going to test?

2. What would you automate and at which layer of the testing pyramid will you place it?

**Test Plan**

**Introduction**

Test strategy includes all types of testing performed for the mobile app for the complete login process and also the workflow includes the successful login and then leads to the main screen landing page of the mobile application

**Scope**

In Scope

Scope defines the features, functional or non-functional requirements of the mobile app that **will be** tested

Out of Scope

Out Of Scope defines the features, functional or non-functional requirements of the software that **will NOT be** tested

**Quality Objective**

* Ensure the AUT meets the quality specifications defined by the client
* Bugs/issues are identified and fixed before go live
* Ennsure the mobile application is compatible for all screens and easy to use.

**Roles and Responsibilities**

Detail description of the Roles and responsibilities of different team members like

* QA Engineer
* Test Manager
* Configuration Manager
* Developers
* Installation Team

# Test Methodology

## Overview

Mention the reason of adopting a particular test methodology for the project. The test methodology selected for the project could be

* Agile
* Extreme Programming

The methodology selected depends on multiple factors. You can read about Test Methodology [here](http://www.guru99.com/testing-methodology.html)

**Test Levels**

**Test Levels define the Types of Testing to be executed on the Application Under Test (AUT**). The Testing Levels primarily depends on the scope of the project, time and budget constraints.

**Bug Triage**

The goal of the triage are follows:

* To define the type of resolution for each bug
* To prioritize bugs and determine a schedule for all “To Be Fixed Bugs’.

**Suspension Criteria and Resumption Requirements**

Suspension criteria define the criteria to be used to suspend all or part of the testing procedure while Resumption criteria determine when testing can resume after it has been suspended

## Test Completeness

Test completeness includes the follwing criteria

* 100% test coverage
* All Manual & Automated Test cases executed
* All open bugs are fixed or will be fixed in next release.

Test Deliverables

Here mention all the Test Artifacts that will be delivered during different phases of the testing lifecycle.

Here are the sample deliverables

|  |
| --- |
| * Test Plan * Test Cases * Requirement Traceability Matrix * Bug Reports * Test Strategy * Test Metrics * Customer Sign Off |

# Resource & Environment Needs

## Testing Tools

List of Tools required are as follows:

* Requirements Tracking Tool
* Jira - Bug Tracking Tool
* Appium - Automation Tools

**Test Environment**

It mentions the minimum **hardware** requirements that will be used to test the Application.

Following **software’s** are required:

1. iOS and above
2. Android and above