Test Management - QA

Question Block 1:
1.1. What will be essential points of your test strategy:

For this project I would cover the following points in the test strategy:

Scope and objectives:
- Rest API and Web Portal
- Assure that the integration with other modules and application server is working correctly.

Out of scope:
- Database testing

Types of tests to be performed:
Functional testing:
- Unit tests
  Automated tests written and run by software developers to ensure that a section of an application meets its design and behaves as intended.

- Component tests
  Tests testing is performed on each individual component separately. The tests can be automated or manual

- Smoke tests
  Tests which cover most important functionalities of the component and can reveal simple severe failures enough to reject the trigger for releasing into the upcoming stage or environment. Best benefit if they are automated and running in the pipeline after each merging to the development branch.

- Integration tests
  Testing that the software modules combined together and tested as a group works correctly. The tests can be automated or manual.

- System tests
  Testing that the entire system works correctly. Some system tests are performed as exploratory testing before release.

- Regression tests
  Testing that the new developments are not causing any regression issues. In agile context the regression tests should be automated as much as possible.

- User acceptance tests
  Testing that the software we are building is fit for purpose. Not only devs and QAs should be involved in this process.

Non functional testing

- Security
  Especially important for the HTTP connections to External components. May be in the scope of the infrastructure/ security department rather than QA.

- Load testing
  Testing to assure that the system will perform correctly under the required load (200 users)

- Compatibility
  Testing to assure the system is compatible with different computing environments, in this case relevant for different web browsers, on different devices and so on.

- Compliance testing
  Type of testing to assure that regulations and compliance are assured. Possibly also in scope of different department like legal, internal audits e.t.c

- Performance testing
  Type of testing to be performed to determine how a system performs in terms of responsiveness and stability under a particular workload - 200 users.

Many functional tests should be applied to both components: front-end and back end.

Key business issue to pay attention to:

- 60% of the profit comes from retailers and 40% comest from the resellers.
Both use our REST API service so it is key point to assure good functioning of the API.

Roles and responsibilities:

- The development team is responsible for the quality of the code and unit tests,
- Testers will be responsible for manual functional testing of the features
- Test automation engineers would be responsible for writing automated tests for component testing, integration and system testing, smoke and regression testing.
- QA team with the support of the DevOps will be responsible for Performance and load testing.
- PO should be included in conducting the User Acceptance testing

Refer to Question 4.3

Testing tools:

Refer to Question Block 5.

Issues tracking and reporting

Given the project uses Jira as an issues tracking tool and change management the same should be used for defect reporting.
Tests, Test Plans, Test executions and Test Sets would be tracked in Xray add on for Jira.

Test deliverables:
Release test reports.
Test automation reports
Test automation codebase
Jenkins jobs for triggering automation on every environment

KPIs and Quality gates / tendencies:

KPI	Quality gate / tendency
Unit test coverage	> 90%
Code duplication	<10%
Regression test plan coverage	>60%
Test Plan Execution time	Descending trend
User story coverage	>80%
Automated regression test coverage	>50%
Defects found by automation v.s. defects found by others	Descending trend
Number of bugs	Descending trend
Number of bugs reported in production	Descending trend
Time needed to respond to critical problem	< 2 days
Time needed to respond to minor problem	< 60 days
Average time to respond to quality issues	Descending trend
Quality issues in backlog	Not surpassing agreed tresholds
Findings closure ratio	Ascending trend
Returning issues	Descending trend
Rejected defects	Descending trend
NPS	Satisfactory
Number of complaints per Period	Descending trend

1.2 Which will be the main challenges for testing in this project

Some challenges in the testing process would include:

- To implement E2E, API tests without the authentication module.
- To get a high coverage for features and modules for both REST API and Web Portal.
- Security testing for the HTTP externally open interfaces.
- Time boxed delivery: when project is time boxed, the delivery of features will be prioritised over resolution of bugs.
- Getting and maintaining good and relevant testing environments and data for functional testing, release testing and performance testing.

1.3 How would you approach the test automation strategy in this project

Ideally if a team really wants agile test automation, has to think differently than the pyramid of writing only automated tests. Of course, unit tests, API tests, basic UI automation are needed. But as long as they are seen (only) as tests, there will not be getting the gains in speed needed. It's only when it’s worked with them as executable specifications, first at a high level, then working down into the implementation details, that one can really leverage test automation in an agile context.

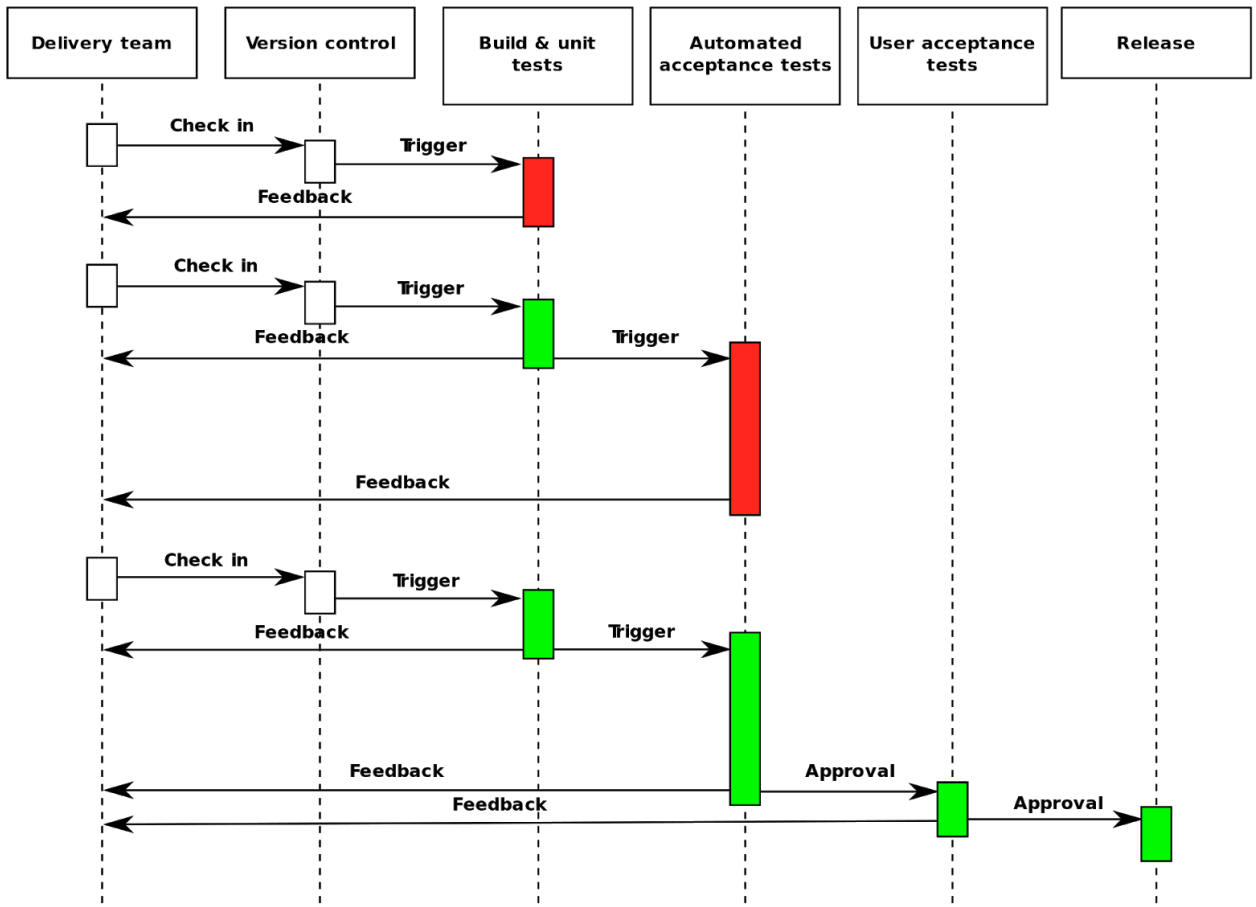

Photo by: Wikipedia.org

1.4. How would you approach the integration of automation tests into the project CI/CD

Much of the integration strategy would depend on the release management of the application: depending if the front and back-end of the system are released as one or separate versions. And if released separately we need to take in account the compatibility between the front and backend versions.

In any case great agile integration of automation tests is when relevant feedback is received often.

Key points for reaching that are:
- Tag the tests depending on when they should be executed.
- Add new test tags for each environment (for example: smoke, regression, full, performance) and run when the code is deployed in that environment.

Question Block 2 - Methodology

2.1. Which methodologies would you recommend for this project development?

For this and any other project development I advocate for Lean thinking combined with Agile development

2.2. Why?

Considering the timeline for the go-live of the project is short from Lean we can benefit in keeping the focus on:
- delivering value from our customer's perspective,
- eliminating waste,
- continuously improve the process.
- and strive for 0 defects.

Lean principles also propose reducing the work in progress or concentrating on one piece at the time which maximises quality.

To be able to achieve the best results it is important to have cross-functional team ready to find solutions through the collaborative effort of self-organising, release often and improve the process, which are the fundamentals of Agile development.

Question Block 3 - Test Team

3.1. How would you configure the test team for this project?

FE Team	BE team
QA Lead	
Test analyst	Test analyst
Automation Engineer	Automation Engineer
Manual tester	Manual tester

3.2. Which skills would you consider essential for your team members to have?

For different roles in the QA team, different skills are required.

Manual tester: Would be responsible for manual testing of the featured being developed during the sprint and manual exploratory testing of the release.
Skills required: analytical mind, knowledge in testing techniques and good management and communication skills.

Test analyst: Would be responsible for defining the test cases, test plans and test executions for the team, working through written and non written requirements analysis and prepare tests as soon as the requirements are communicated.
He needs to have good organisation, test management and communication skills, technical knowledge and analytical mind. Bonus points for experience with BDD.

Test automation engineers: In this context it makes sense to know JS because most of the team members use that programming language and they can help writing/ maintaining the test in the case of need or absence. Good knowledge of REST API services, DevOps and agile methodology mindset.

QA Lead: Would be responsible for defining the test strategy, assuring good integration of the test automation within the CI/CD, defining and measuring KPIs, implementing quality gates and continuous improvement of the process. Skills required: analytical mind, technical skills, experience, DevOps and Agile methodology mindset, Lean thinking. Good communication skills, good listening and dialoguing skills.

3.3. Which mindset and attitudes would you look for in your team members

According to me attitude is more important than the skills. Skills can be learned, and attitude and character not so.
It is even desirable for the people not to be overly skilled for the job, so they can learn and keep high motivation, but it is very important to have the right mindset and attitude.
A good team member is proactive, positive and kind.
Also he/she is a good team player, supportive and always looking for the collective benefit.

Question Block 4 - Test Process

4.1. How would you identify the scenarios or test cases needed?

In the ideal word the test scenarios would be already defined and written as executable specifications. In case of evaluation which ones need to be covered with automation first, I would approach the decision by defining a severity / priority matrix of the test cases.

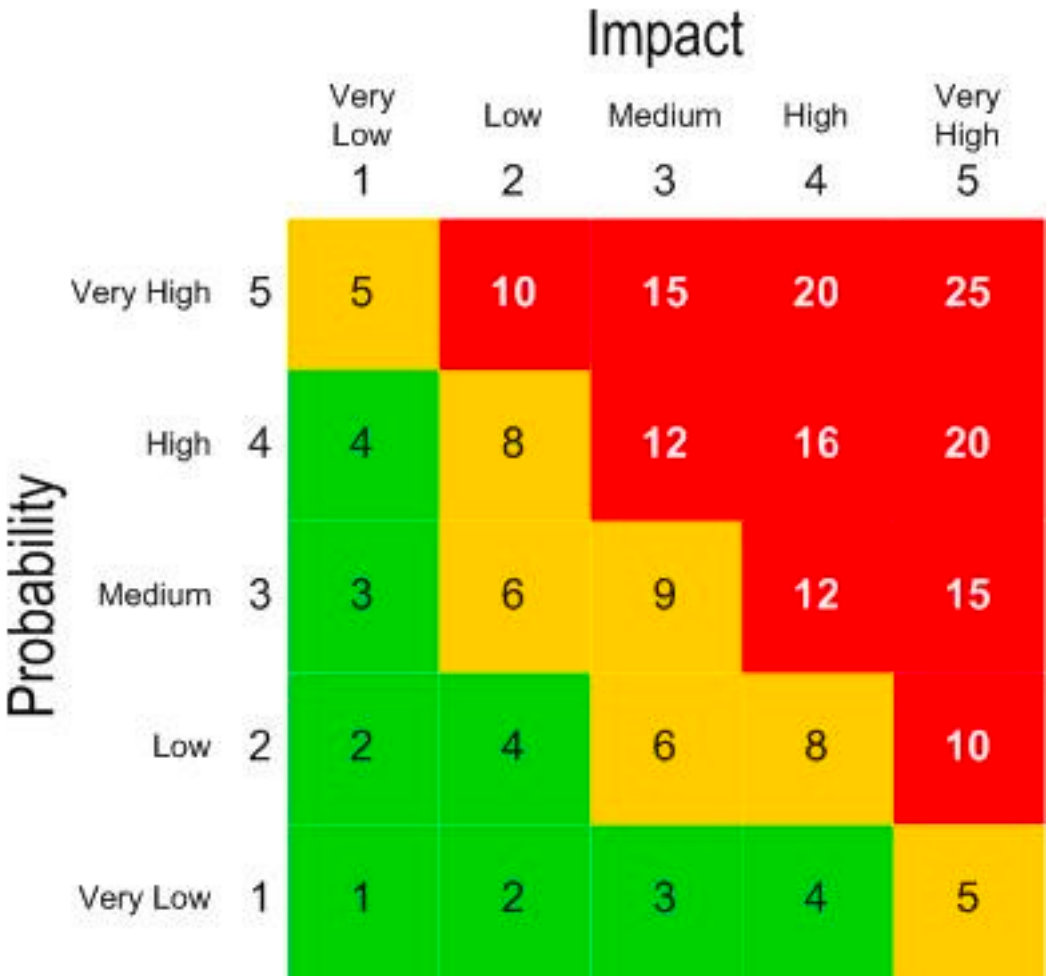

Photo by: Getzephyr.com

4.2. How would you keep traceability between test scenarios and requirements?

If JIRA is used as change management tool I would use **Xray add-on for Jira** as a test management tool. That would also allow us to measure the coverage of requirements and user stories and deliver KPIs about the QA and team's performance.

4.3. How would you split the testing responsibilities between developers and testers?

Quality unit	Responsibility
Code quality	Developers
Unit tests	Developers
Component tests	BE test automation engineer
Integration tests	Manual BE tester
System test	Front end tester
Regression tests	FE Automation engineer
User Acceptance test	PO + retailers / resellers
Performance testing	QA with the support of BE devs and DevOps

Question Block 5 - Tools

5.1. Which tools would you use for Test Management?

- Xray add on for Jira for test management,
- Jira for defects logging

5.2. Which tools would you use for Performance testing?

When using a Performance tool we should be taking a look for many ways the tool would be fit for our purpose before making the decision. Some are:

- financial fit
- scripting fit
- accuracy fit
- scalability fit
- extensibility fit
- collaboration fit
- reporting fit

If we are taking in consideration that the framework that will be used for automation be JS then Artillery might be a good choice as it's open source, powerful and easy-to-use Load Testing solution written in NodeJS compatible with npm modules.

5.3. Which tools/frameworks would you use for API and API automation?

Mocha JS: a JavaScript test framework that runs on node js. There are a lot of features pre-build with the Mocha test framework like:

- Grouping the tests
- Skipping tests
- Using hooks to set up or tear down tests
- It also comes with reporting as well as retry support
  along with many other useful features

For manual testing: Postman.

5.4. Which tools/frameworks would you use for FE testing and FE automation?

For the automation solution I suggest WebDriverIO + JS + Selenium framework using Allure framework for reporting

For the manual testing I would use proxies to inspect the traffic.

5.5. Which other tools would you use?

EasyBI or Tableau for following trends and extracting KPIs,
Confluence for documentation,
Jenkins for CI/CD.
Bitbucket for code repository
SonarQube for code quality control.