Test tasks for QA Engineer at Relayr

Task 1

Imagine the following situation: you are joining a cross-functional team which builds a front-end application using REST APIs. You are a first QA engineer and need to establish a QA process in the team.

1. What would you do in your first few days of work? Where would you start?

   - As a QA engineer, it is very important to understand the business and the application to be tested. My first task would be to gain the business knowledge, it broadens the functional understanding with respect to testing.
   - Next task would be to understand what kind of solution the application to be tested, is going to provide? what kind of technology being used to develop it? How the data is flowing in the application? Does the application use third party service? Do we depend on other services to proceed with our tasks? I would try to get hands on by using the application to get a grip on the functional flow.
   - Since it is a cross-functional team, I would try to understand the process and methodologies such as if we are following Agile how frequent are the releases, how the scrum is conducted, does the sprint last for 2 weeks or 4 weeks, retrospectives, sprint planning and so on. Based on the information, I would strategies and establish QA process such as in sprint automation for API or current sprint + 1 automation for UI. Test strategy is the mandatory action to be completed before starting the QA process.
   - Once all the knowledge transfer is completed, After gaining all the required accesses, I would start documenting some sanity testing scenarios to be performed from API and UI perspective. I would perform sanity testing on the application under test End to End and report defects if any.
   - As a QA engineer, I would be collaborating with different teams for different purposes such as DevOps, if there is other QA team I would like to know which process do they follow for manual, automation and non-functional testing? I would try to understand If any framework is in place for automation
   - If there is no framework existing for automation testing, I would create a basic skeleton for API and UI that would include test case reading and step definition with data flow, locators repository, reporting for execution, utilities for reusable methods. These would be enhanced in parallel with the QA tasks as per the application.
   - I would strategies for Regression testing as well such as executing all the regression test cases on every new change in the code, nightly or weekly execution of test cases through Jenkins and so on.

   This would accommodate a week or a couple of weeks for a QA but performing these tasks initially would be helpful in long run.

2. Which process would you establish around testing new functionality?

   - Firstly, testing any application requires clarity on the requirements i.e. proper acceptance criteria defined for the respective functionality. It is a mandatory step

to discuss possible scenarios around the functionality. The business analyst, developers and QA should be on same page with respect to functionality and agree on the high level scenarios to be developed and tested. It is usually called “3 Amigos” meeting which will filter almost all the confusions and define a boundary on the scenarios to be considered while developing and testing.

- Secondly, the impact of new functionality on the existing one should be analysed as how much of effort would be invested on performing Regression to understand the impact. As a QA, we should be aware about how much the code is stable i.e. the unit test coverage to understand the clustered area to be focused more for testing.
- Once the new functionality is deployed (It should be dev tested thoroughly), one round of automated regression test cases should be executed and then manually sanity should be performed to understand the impact
- A set of sanity and regression test cases should be maintained to get a quick result after every deployment. The edge cases should be tested and if possible, it should be added in regression.
- For every new functionality, update on the testing suite should be done to increase the coverage of testing through automation.

3. Which techniques or best practices in terms of code architecture and test design would you use in your automated tests?

- UI Test Automation:
    - Using Page factory model to maintain locators, test actions, utilities and step definitions and test logic separately
    - Hooks (Common and repeated actions) should be introduced for pre-requisites to be performed before starting the tests.
    - Handling multiple environment (profiling) and multiple browsers (if the application supports multiple browsers) should be done to perform cross browser testing
    - Data generation should be in place to make it compatible with testing as for some test cases a particular kind of data is needed. So the pre-requisite for that test case should be met by data generation
    - Independent execution of test cases should be done so that if one test case fails the other one should not be impacted, and parallel execution is possible when the test cases are independent
    - A configuration is needed to switch between the tags, if the QA want to execute selected test cases, a single configuration needs to be done ton control such execution
    - Reporting is very important. A screenshot should be taken on each step (assertion) during the execution whether it passes or fails.
    - Also for frequent executions, QA can integrate it with Jenkins to run the test cases quickly and in headless mode

- API Test Automation:
    - API testing requires json schema validation which could be done by using swagger model or postman collection. A utility should be created to fetch the structure of each API request and perform request submission

- Utility such as parsing the json file should be created in separate class, test action, data flow should be maintained in different class
- Jars for generic utilities should created and used for both API and UI
- For authentication handling should be done for all the common APIs
- If the output from one API is used as an input for another, it should be covered in one flow (scenario)
- Reports for execution should be generated at the end of execution

Task 2

How would you approach test automation of an HTTP API of some service? Choose a service from this list https://github.com/toddmotto/public-apis or any other of your choice. It is better to pick one without authorization as it is easier to test.

Document several test cases. Implement one or two automated tests based on the test cases. Write the test cases as if they were a part of a real project.

Solution: Task solution has been added on Github