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?
2. Which process would you establish around testing new functionality?
3. Which techniques or best practices in terms of code architecture and test design would you use in your automated tests?

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 tests as if they were a part of a real project.

SOLUTION

TASK 1

1.   With organisations having little to no luxury of delivering sub-standard products or services. The need for Quality Assurance (QA) process to ensure that the product is built right the first time is now key. Software Testing is a sub-set of Quality Assurance thus QA is far beyond finding bugs. The goal of the QA team is to deliver high quality product within the scheduled timeline, detect any issues that will serve as blockers for the seamless operation of the developed product and affects user experience

As the first QA Engineer in a team tasked with the above responsibilities, my first task is to create awareness about delivering high quality products or services. The need for Total Quality Management (TQM), where everyone in the team has a role to play in delivering high quality products from product inception to delivery is a given. From product owners defining the requirements correctly, Software Developers understanding every bit of the requirements and implementing code using define frameworks, etc are a must in achieving user friendly applications. Demo session and One on one session will be held to get staff involvement

Part of my few days or weeks will focus on planning and designing a QA process initiative roadmap. The planning stage will involve working out solutions to the problems and lapses identified from teams' feedback from the previous sessions. The next stage will be designing a

roadmap for the new QA process model. The roadmap describes the new process specifics, define quality metrics, and consider risks that may arise in response to changes.

The QA process model will covers the following key areas and will be focused on in my first few days:

- **Create a dedicated Test Environment:** Having an independent test environment similar to production that will help to simulate production like scenarios. The test database will be occasionally refreshed to purge old data. Likewise, with segregation of environments; Development, Test, Pre-production, Production environments established controls are put in place around each environment.
- **Early QA involvement in product development:** Getting QA personnels involved in the product design and development process gives a higher chance of meeting the delivery timeline. QAs must be involved in the early days of product development and not wait until development has been completed before being active. As it is well known that it is far cheaper and time efficient to correct bugs early than late in the project phase.
- **Test Management & Automation Tools**: The use of test management tools has made QAs work more efficiently and effectively. Also, automation tools such as Selenium Webdriver and Cypress for UI test automation have aided early product release and help with drastically reducing reported bugs. Depending on financial constraint, the team will be open to the use of open source tools as they have proven to be reliable as well.
- **Automation Suite for critical applications**: Once the test tools are in place, for the first few days at work my target will be to have an automated regression suite for the frequently changing and critical web applications.
- **Documentation storage**: Categorising and storing test data, documentation and other test artifacts securely is required and will help with easily retrieving whenever needed.
- **Create a deployment pipeline**: In an agile project management set up, A deployment or build pipeline in an agile project defines how a story gets from product backlog to live production site. It defines a process and the activities that happen at each stage.
  In order to implement a successful QA process that ensures we are frequently releasing quality code, the deployment pipeline must be defined and be adhered to by all stakeholders. The deployment pipeline is the spine of software delivery.
- **Introduce release management**: Release management is required anytime a new product or even changes to an existing product are requested. The release management process will govern how changes are moved between various environments.

2.   In testing new application features, the following processes will be established

Create a test plan for the new application functionalities: This document will cover the test objective, test approach, the scope of test, features to be tested, timeline etc. This document is a live document and should be updated and reviewed regularly.

Automate the new functionalities: Create automated test scripts for the new features that can be automated and add to the regression testsuite (if any exist). UI Test automation helps increase productivity by reducing time and effort. Using test framework such as TestNG, Junit and test design pattern will make maintaining the testsuite easier.

Requirement Traceability Matrices (RTM): Create a Requirement Traceability Matrices (RTM) will ensure wider test coverage. All created new test cases must be mapped to a requirement and tested.

Source and Version Control Tools: Using source control management (scm) tools enables colleagues to collaborate seamlessly and manage changes to code and documents. Also SCM tools support versioning so adding new code/test script to testsuite for newly implemented functionalities can be tracked and reverted.

Continuous Integration: In order to effectively test new features, we need to ensure the code works not only on the developer's machine but also on other environments. Continuous Integration helps in identifying any build problems early on in the process, so that when the deployment fails we can start to look where the issue is coming from. Every code change must be tested and run on a continuous integration server (e.g Jenkins) with mail notification enabled.

3.   For the UI automation test, the listed code architecture and techniques will be implemented to make the automated test suite easily extensible and maintainable.

Techniques	Description	Benefits
Page Object	Page Object Module test design approach is used to separate the page objects from the test cases. The web pages are represented as classes, and the various elements on the page are defined as variables in the class. All possible user interactions can then be implemented as methods in the class. While the tests will be created as separate class(es) and will be able to call the methods	Users can easily read and comprehend the test scripts(code). Also it makes code reusable, as elements and methods can be called by any test scripts. This makes test script easily extensible for new functionality
Logging	Using a logging system, information that helps the user to	This will help with debugging when the need arises

	understand the test steps and failures during the test execution are captured.	
Data Driven Test	I will keep the test datasets in files separate from the test script(logic). Test scripts and test data both will be independent of each other, and both can be added, updated and deleted without any implications on each other. For cypress automation, the fixtures folder holds the dataset. While this can be achieved in Selenium using the dataprovider in TestNG	Test Data are organised in a file and separate from the logic. Eliminate hard coding of variables. Thereby it is easy to maintain data sets without needing to adjust the test script. This eliminate tight coupling
Report	Using the TestNG framework in Selenium, we can generate detailed report for our test runs in xml and html format	Generated customized and detailed report for executed test.
Test Listener	Screenshot for failed test cases will be enabled by implementing the interface(listener) to listen to event during test execution	Also helpful during debugging. Stage where the test fails is captured.
Cross Browsing test	Using cypress automation tool, the test can be run across common browsers and reported/known problematic browsers. This can also be achieved in Selenium by invoking the various browser drivers.	Ensure applications run smoothly on different browsers
Simple Naming Convention	Using camel case naming conventions for methods, variables, etc by having simple but meaningful names	Use meaning name and keep a standard naming convention

TASK 2

The test approach to be adopted for testing an HTTP API is to confirm that the API functions correctly. This is functional testing. Thus for API test automation, we include assertions to verify the following response items.

- Correct HTTP status code is returned.

- Correct response headers are returned.

- Verify the response payload returns correct field name, values, messages.

- Verify the response time for performance related issue.