1. Imagine the following situation: You need to establish a QA process in a cross-functional

team. The team is building an SDK to implement into partner products front-end application

using REST APIs.

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

I would like to start by involving key parties. In this sense, everyone involved in a project is a key party. Developers, Product owners constantly reviewing requirements and do hands on testing, Project managers, Scrum masters regularly review plans to re-align priorities and extract best value and Partners.

My first steps would be collecting clear requirements and would prepare well understandable and formatted document. This process involves

* Gathering all the REST APIs requirements
* Prepare the draft document
* Analyze the requirements
* Validate the requirements
* Verify the requirements with Team Lead/Product owner
* Finalize the requirement document

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

want the features to be tested?

New functionality testing needs test strategy and test plan

* Test environment, requirements and set up. In our case simulate partner product for future integration of our SDK
* Test approach: Manual (exploratory) and Automation
* Risks and dependencies like API downtime and third party tools
* Non functional requirements like API latency, API security tests, Performance testing
* Test schedule
* Resources required to test this feature

Test strategy will help us to articulate what Operating Systems, OS versions, devices (MacOS, Windows, linux) need to test the API SDK for. Make a list of all the third party software’s that the application will need (If required/supported).

I would like to test the features by following the test plan. Starting by executing the testcase document which contains all the testcases of our SDK integration, exploratory testing, also non functional testing and also would do Regression testing to make sure the new feature implementation has not been impacted on other working features.

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

I would like to suggest team the following tools which are best and easy to use

* Postman: Postman is a good choice for API testing for those who don’t want to deal with coding in an integrated development environment using the same language as the developers. Advantages of this tool
  + Easy-to-use REST client
  + Can be used for both automated and exploratory testing
  + Can be run on Mac, Windows, Linux & Chrome Apps
  + Doesn’t require learning a new language
* Jira tool: To log bugs with clear description, attaching required documents.
* Teslinks/TestRails: For maintaining testcase documents and executing testcases
* SoapUI: SoapUI is a headless functional testing tool dedicated to API testing, allowing users to test REST and SOAP APIs and Web Services easily.
  + Reusability of Scripts: load tests and security scans can be reused for functional test cases
  + Create test quickly and easily with Drag and drop, Point-and-click
* Confluence: To maintain project related information like credentials, documentation of SDK, protocols etc
* Swagger: For API documentation and easy to understand API interaction flow
* Kibana: To check the logs which helps in knowing the root cause of the bugs
* Charles: To check API request and response

4. If you would do test automation which techniques or best practices would you use the

Application?

The following are the best practices/techniques I prefer to use

* Decide what Test Cases to Automate: It is impossible to automate all testing, so it is important to determine what test cases should be automated first. Good test cases for automation are ones that are run frequently and require large amounts of data to perform the same action.
* Test Often: Testing should be started as early as possible and ran as often as needed. Bugs detected early are a lot cheaper to fix than those discovered later in production or deployment.
* Select the Right Automated Testing Tool: Selecting an automated testing tool is essential for test automation. It is important to choose the automated testing tool that best suits your overall requirements.
  + REST-Assured: REST-Assured is a fluent Java library you can use to test HTTP-based REST services and it integrates with any existing Java-based automation framework.
  + Postman: Postman is a nice option for exploratory-type API testing. Postman is an easy-to-use REST client, and we can get started with it quickly leveraging its Chrome plugin.
  + SoapUI: SoapUI is a fully functional test tool dedicated to API testing(load testing, security testing)

- Documenting Test report: After execution of test cases the status should be updated to check whether all functionalities are working as expected.