**Please find my answers in blue below-**

**Gherkin**

1. Write Gherkin tests for the program you wrote above. Use any Gherkin features or practices you

want. Don’t write step definitions (i.e., the tests don’t have to be executable).

2. Explain in detail why these tests might be helpful in the future.

* user can write their own step definitions during run time, based on the specific need as and when we get more clarity on the requirements.
* practice of writing scenarios as a process of continuous discovery

Tools

1. In your opinion, what’s helpful about version control systems? What’s annoying about them?

* Main advantage of version control is, it keeps track of who is doing what.
* Every member on team can develop independently without any dependencies by pull the master to their branch and later merging the code with master.
* If found any failure or defect found in the latest version, I can switch back to previous stable working version without much effort.
* Few things I found annoying about version control are, the process takes its own time adding extra layer to coding, merging conflicts due to platform differences, integrating into existing workflows, maintaining too many repositories and environments adds overhead.

2. What are some pros and cons of using Docker to develop, test, and deploy software?

**Pros –**

* maintaining all development environment is easy
* create container image which can be used across development process.
* compatible, runs on different servers without any dependencies.
* portable, can be used in any cloud platform to achieve same behavior and consistency.
* maintain all configurations and dependencies internally.
* speeds up development, test and deploy process with help of same container across environments.
* less documentation and more automated processes.

**Cons –**

* Security, going public can have security issues. Since there are no full OS, hackers can get into the container if it’s not secured well.
* malicious Docker images can be found in Docker hub.
* limitations of using too many of containers on the same server.
* networking is challenging if it’s a secure environment.

3. How do you choose which language to use for a given task? How did you choose the language

for the programming exercise above?

* Coding language is chosen based on the type of application technology platform and how easily it integrates without much hassle.
* I don’t have experience in too many languages, most of my career I used c or java for automated test script development. I chose java since I’m more comfortable with it.

**Testing Methodology**

1. What’s the right role for QA in the software development process?

* Role of QA is to make sure the software meets all quality requirements before its released to production. QA has to understand the end to end process, components involved, various services, integrations, data interactions between systems to build a test strategy which covers all possible scenarios and make sure software is expected to behave as defined.

2. As a QA person, you have 2 weeks to prepare before your team starts writing software. What do

you do?

* I will make sure I understand the software and what is being developed at what phase of the project.
* gather user stories and create test cases against it.
* Create a entry-exit criteria for me to start testing.
* Create a draft test strategy, test plan and get it review with all stakeholders on testing approach.
* setting up my test environment, tools, gather test data, getting necessary access.
* Identify the key people who is involved in the project.

3. When is it appropriate to use automated testing? When is it appropriate to use manual testing?

Automated testing is used when:

* Regression testing. if you are planning to test same scenarios more frequently.
* No. of test cases involved for testing, if there are too many scenarios to test.
* critical scenarios, without any manual errors.
* testing on multiple platforms, systems.
* sanity/ smoke tests.
* Huge data set involved
* for continuous integration, build testing.

Manual Testing is done –

* if test scenarios are asynchronous, involves system delays and external triggers which determines tests.
* when test case numbers are small, and it’s not repeated.
* automation tool constraints in accessing multiple systems for validation and verification purposes.

4. Your dev team has just modified an existing product by adding new features and refactoring the

code for old features. The devs claim to have written unit tests; you’re in charge of integration

testing. Dedicated teams are handling performance and security testing, so you don’t have to. As

is always the case in the real world, you don’t have time to test everything. What factors do you

think about as

* I will first check the change log to see what existing features were touched and what new features were added. And its dependencies with other features.
* I will run my set of end to end critical(integration) test scenarios to make sure those didn’t break.
* If unit tests are documented, I will get the details which will help me in my integration testing
* I will test newly added features and report any defect if found.