What is your definition of testing?

Testing is to affirm the quality of software and to deliver an error-free application. It is an activity to check whether the actual results match the expected results after the execution of the software component of an application under test. It helps to evaluate the functionality of a application under test with the intention to find out whether the developed software met the specified requirements or not and to deliver defects free application in order to produce quality products

 Good practice for setting up UI automation, include any tools or good framework set up that may be relevant?

From my personal experience, I will say it is very difficult to come up with the exact rules everyone need to follow or best practice in setting up automation framework because every company is different . Despite saying that, there are some good practice and steps ones can take in other to build a good automation framework. Below are examples of what I think it will be a good practice using selenium webdriver.

**Look beyond automation UI**

When designing automation framework, it is advisable to look beyond automation UI, for whatever reason organisation may decide to take off the whole application UI, you should be confidence that you can still catch all or most of the defect by running the test. Therefore it advisable to design a framework can automate beyond the application UI by adding a sub package on the framework layer that will allow a web services (API) test will be very beneficial

Take the organisation technology into consideration

In my own personal experience, I have worked in organisation where all the developers only know PHP, JavaScript, CSS etc. They did not know either C# or Java language and when I ran into any trouble in my script written in Java or having problem getting automation test to live deployment that is beyond my knowledge. This had a massive impact on the automation testing

**Using behaviour driven development (BDD) framework**

Writing test in BBD gherkin syntax helps to create a specification that will help the whole team understand the tests without the need of the QA engineers to explain their test. BDD also helps to understand the requirement better as well.

The fact BDD is written in a simple plain English, it makes everyone in the team to understand the values and benefit the test will bring to the project and any recommendation or suggestion can be brought forward because everyone is on the page in term of understanding what the test is doing

BDD helps to avoid duplication code because if different sceanrio share the same steps, you can use the same step in different scenario.

In order to get BDD working in .Net environment, you just need to install specflow via nudget and packet manager while in intelli J, you can add cucumber jar,

It is recommend to be a good practice to create a folder called Feature that will hold all the feature files

Consider using Page Object Model (POM)

When automating a UI application, I will need the web element of the page that need automating. It is better for me to create a separate class that will hold the web element of that page using POM. The class will have the element of page (such as clicking, button, textfield etc). Then I can use selenium webdriver to handle the actual interaction

I will write my method on how the I want my application to interact in this page. I will just call any method I want in my step definition/implementation.

There are other benefits such as:

Making test easy to read and understand the page interaction

Easy code mainatainace

Avoid duplication because the method can be used in multiple steps definition without need to specify the page locator more than once.

Well organised code structure

Naming conventional

It is a very good practice to name methods, class, fields and any variables used in automation framework wisely and related to the project e.g. Information about customer can be CustomerDetails, CustomerAddress etc. It makes code easy to understand, read, maintainable and avoid using different name for the same functionality

Writing a generic code

Making code generic is one of the best way to avoid duplication of code and also help code maintenance

Reporting

As tester I can see the result of my executed test on my screen, but it is not practical to call everyone in the team to show them the outcome of my test result, therefore having a test report is very essential in setting up automation framework. There are free reporting software that can be used such as pickleUI, extent report etc

Folder Structure

When setting UI automation framework, I need to have a separate folder for each file, e.g framework should be separate from testing folder.

Within test written, there should be a separate folder for each test of the same type e.g all feature files should be kept under "Feature Folder", steps implementation/definition should be kept under "Step Folder", reference, namespace, dll, jars, assemblies should all be in a separate folder

Avoid Using Thread.Sleep

It is best practice to avoid using this method instead use waitUntil certain page is found because thread.sleep consume resources and block the current thread

Using Inheritance

I can inherit a class and framework that will be using frequently e..g using selenium WebDriver that handle the actual UI application interaction, I will create a protected base class that will contain all the WebDriver instance in order the UI application to interact. This base class can be inherit base page class that hold all other page that are visible to all pages and steps in the application without need to duplicate the code

Using Hooks

Create a separate class that will hold all the hooks methods such as before test, before scenario, after test, test fixture, test etc

Consider a peer review

I consider this as a good practice because nobody is perfect and it is good to have another person within the team that understand how the UI automation framework should work to check the automation framework configure incase of anything missing or bring any suggestion even if it is not implemented at that stage

Reporting, Separate folder to hold files e.g. Features, step Implemetation, inheritance, Hooks, browsers compatibility, separate test from the test auto framework, neccessary assemblies/references, access to all these necessary assemblies/references should be available, Peer review to point out any missing part during the setting up of the framework, naming of test wisely and make it relate to the project you are working on, avoid using thread,sleep if possible, look beyond automation UI and design a framework can automate by

adding a sub package on the framework layer that will allow a web services will be very beneficial

Adding a sub-package to your framework layer for web services is a very beneficial approach.

1. **As a Scott Logic consultant, you may be required to travel both domestically and/or internationally. are there any known reasons that would restrict you undertaking travel as and when required?**

It depends on when I need to travel. I have a daughter that I go and see every other weekend, if it's not the weekend I need to have my daughter then it's fine. I don't mind travel both internationally and within the country

1. To your knowledge is there anything that could stop them from being Security Cleared? E.g. immigration status, bankruptcy/insolvency.