**IT Infrastructure**

**Final Assignment**

|  |  |
| --- | --- |
| **Assignment:** | **IT Infrastructure** |
| **Submitted to:** | **Sir Usman** |
| **Submitted By:** | **M.Ahsan Ghafar** |
| **SAP ID:** | **27226** |

**Q-1**

**A:** **What is the test automation framework?**

Test automation frameworks are a set of principles with accompanying tools for creating test cases. Its purpose is to make engineering activities more efficient.

**Types of Automated Testing Frameworks**

* Linear Automation Framework.
* Modular Based Testing Framework.
* Library Architecture Testing Framework.
* Data-Driven Framework.
* Keyword-Driven Framework.
* Hybrid Testing Framework.

**Following down is the brief discussion of some automation frameworks:**

* **Data-Driven Testing Framework**

A sequence of tests must be run before a successful outcome can be determined when testing an automation framework. In such circumstances, you may need to modify the test data to come to a different result. The Data-Driven Testing Framework gives you the option of storing test data on an external device, which can then be accessed to add a new script to the test case later.

* **Keyword-Driven Testing Framework**

The keyword-driven testing framework, which is often thought of as an extension of the data-driven testing framework, not only extracts your test data to an external source but also safely preserves the set of codes. These codes are also known as 'keywords,' and they can be used to change the test script in order to get more information from the test framework. These keywords also determine the functions that each application does.

* **Hybrid Testing Framework**

To get the most out of the aforementioned frameworks, the hybrid-testing framework combines the data-driven and keyword-driven frameworks. It is the best automation framework since it gives you more options for enhanced productivity and a higher success rate.

* **Linear Scripting Framework**

A single person who records each step of the procedure, on the other hand, does linear scripting, by hand. This framework, also known as the Record and Playback framework, requires a significant amount of time to complete. It does not require automation knowledge and is a simple approach to record a script, however it does not allow for script reuse. Maintenance takes time, and you would have to write the test script by hand in order to run the framework in the future.

* **Module Based Testing Framework**

The Module Based Testing Framework, as its name implies, requires the availability of several modules in order to function. To get the best results from the automation test, you would have to write distinct scripts for each module and cooperate them. The modules would be unaffected if the application's functionality changed. Until the scripts are manually changed, they are unaffected. Because the combining of numerous modules generates a high level of modularization, this is a cost-effective management strategy. Productivity is still at an all-time high. However, if there is a necessity, a lot of time and effort can be put into making unique alterations to the test data.

**B: What exactly is selenium?**

Selenium is an open-source (free) automated testing framework for validating web applications across multiple browsers and platforms. Selenium Test Scripts can be written in a variety of programming languages, including Java, C#, Python, and others. Selenium Testing is the term for testing done using the Selenium testing tool.

Selenium Software is more than just a single tool; it is a collection of tools, each of which caters to a specific organization's Selenium QA testing requirements. The following is a list of tools.

* Integrated Development Environment for Selenium (IDE)
* Remote Control for Selenium (RC)
* Selenium Grid with WebDriver

**C: How Selenium Works?**

There are multiple definitions and I have concluded few of them following down for better understanding the working of selenium

* Selenium is controlled via API commands such as GET and POST, and it responds to Selenium script requests. The queries are subsequently transmitted through HTTP to the browser driver's HTTP server as well as the browsers.
* Selenium RC is a Remote Control that operates by intercepting the browser's remote and then inserting the automation code to be tested using custom scripts. Selenium RC uses the Selenium RC Server to interface with browsers. When a web page is loaded, it injects JavaScript functions into browsers.

**Examples defining how selenium works:**

**Steps for Login Automation using Selenium WebDriver**

1. Create a Selenium WebDriver instance.
2. Configure browser if required.
3. Navigate to the required web page.
4. Locate the relevant web element.
5. Perform action on the web element.
6. Verify and validate the action.

**How is Selenium used systematically?**

1. Basic Steps in a Selenium WebDriver Script
2. Create a WebDriver instance.
3. Navigate to a webpage.
4. Locate a web element on the webpage via locators in selenium.
5. Perform one or more user actions on the element.
6. Preload the expected output/browser response to the action.
7. Run test.

**D: Why do we need selenium**

Consider the following scenario for a manual tester: verifying that the web app's signup page (www.example.com/signup) verifies input strings and properly registers a user in the newest versions of Chrome and Firefox on Windows 7.

Assume the signup page includes the following fields: username, email address, and password. On the latest versions of Chrome and Firefox, the tester will acquire a Windows 7 desktop and do the following steps in order:

1. Enter the URL in the address bar (www.example.com/signup)
2. Enter an invalid string in each input field (email, username, and password)
3. Check whether the input strings were validated against corresponding regexes and any pre-existing values in the database
4. Enter ‘valid’ strings in each input field; click Sign Up
5. Check whether “Welcome, ‘{‘username’}’“ page showed up
6. Check whether the system database created a new userID for ‘{‘username’}’
7. Mark the test ‘passed’ if it did, ‘failed’ if the signup feature broke anywhere during the test.

That is a very basic system test. In the real world, testers are more likely to be checking all user workflows on www.example.com for breakage, on as many OS-browser combinations as needed to meet the benchmarked compatibility standards.

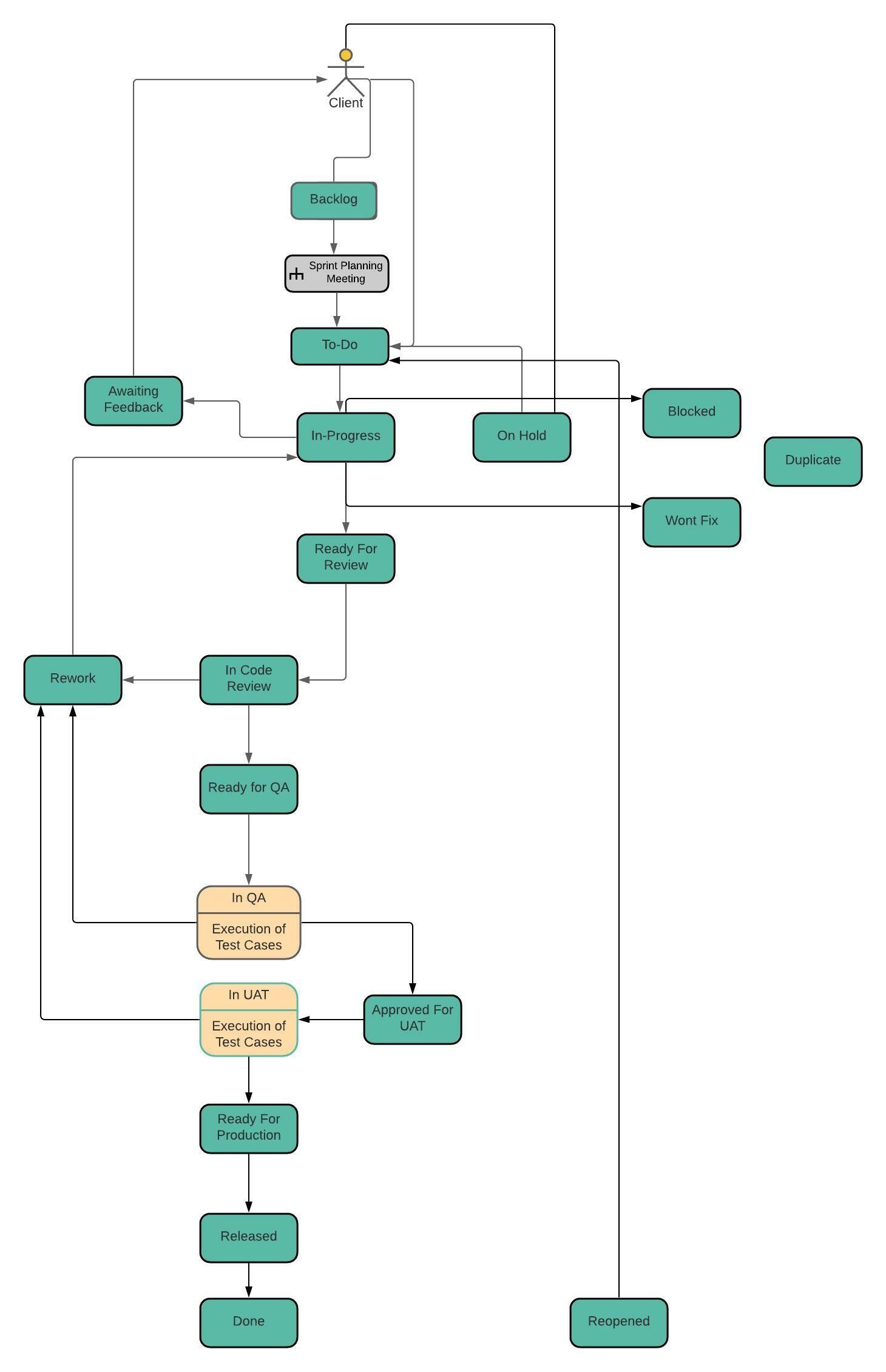
Depending on the number of manual testers (and thoroughness of test cases), it may take anywhere between hours to weeks to be sure that the web app is fully functional.

Modern developers and product teams do not have that kind of time to allot for testing, but they can’t set aside exhaustive testing in a hurry to release either. This is why they super-charge their testing with automation, powered by Selenium.

**Selenium in agile projects**

Running any project under agile methods and running repeatable tests utilizing automation — Selenium WebDriver for a web-based application – is necessary in the software industry. Selenium WebDriver is widely used in agile projects because it allows each team member to contribute to the automation of test cases. By definition, agile methodology is a strategy that is particularly useful for responding to short customer-induced change requirements and so lends itself well to frequent modifications during the application's entire development.

***Following down is the scrum model we used at Innovadel technologies to test the projects:***



***Q-2*** ***the most common tools that are used for configuration management are packer and ansible. You need to concisely compare both of them.***

***Answer:***

***Let’s discuss ansible and packer side by side***

|  |  |
| --- | --- |
| **Ansible**  Configuration management, application deployment, task execution, and multi-node orchestration engine are all made extremely simple. Ansible is a tool for IT automation. It can configure systems, deploy software, and handle more advanced IT processes like continuous deployments and rolling updates with no downtime. The primary goals of Ansible are simplicity and maximum ease of usage. | **Packer**  From a single source configuration, create identical machine images for numerous platforms. Packer is a programme that automates the production of machine images of any form. It encourages you to use automated scripts to install and configure software within your Packer-created images, embracing contemporary configuration management. |
| Ansible belongs to "Server Configuration and Automation" category | Packer can be primarily classified under "Infrastructure Build Tools". |
| Sysadmins, developers, and IT managers can accomplish automation projects in hours, not weeks, thanks to Ansible's natural automation language. | Infrastructure deployment is lightning quickly. Packer images allow you to start machines that are fully provisioned and configured in seconds rather than minutes or hours. |
| Instead of requiring agents everywhere, Ansible uses SSH by default. Increase security, reduce "managing the management," and save CPU cycles by avoiding superfluous open ports. | Portability between several providers. Packer can run production in AWS, staging/QA in a private cloud like OpenStack, and development in desktop virtualization solutions like VMware or VirtualBox because it builds similar images for many platforms. |
| Ansible is a solution that automates app deployment, configuration management, workflow orchestration, and even cloud provisioning. | Stability has improved. Packer instals and configures all of a machine's software now the image is created. If these scripts include flaws, they will be discovered sooner rather than later, rather than several minutes after the computer has been run. |

***I have gathered some reviews of few solution architect’s and developers on packer and ansible, which are mentioned below:***

***Stephen pepp***

I have been working with Puppet and Ansible. The reason why I prefer ansible is the distribution of it. Ansible is more lightweight and therefore more popular. This leads to situations, where you can get fully packaged applications for ansible (e.g. confluent) supported by the vendor, but only incomplete packages for Puppet.

The only advantage I would see with Puppet if someone wants to use Foreman. This is still better supported with Puppet.

**Gabriel Pa**

CEO at NaoLogic Inc

If you are just starting out, might as well learn Kubernetes There's a lot of tools that come with Kube that make it easier to use and most importantly: you become cloud-agnostic. We use Ansible because it's a lot simpler than Chef or Puppet and if you use Docker Compose for your deployments you can re-use them with Kubernetes later when you migrate

*References:*

*Stack overflow*

*Stack tech*

*Stack share*

*Selenium IDE*

*Innovadel Tech*