SDLC

* Initial
  + Kickoff Meeting
  + High level view of intended project and defining its goals
* Analysis
  + Triple constraints of Project Plan – Resources, Time, Money
  + Work Breakdown Structure (WBS) is created to show tasks to be done by team
* Design
  + High Level Designing and Low-Level Designing
  + All requirements should be in place
* Coding
  + Developers write the programs for the project
  + Developers prepare the source code
  + A contingency plan is developed in case of emergencies
* Testing
  + Arrange Requirements Walkthrough meetings with BA/PM
  + Create Software Test Plan, Test Strategy, Scenarios, Requirements, Test Case, and Test Script
  + Test Design and Artifacts – How you will design your test cases
  + Software Testing Life Cycle (STLC)
  + Bug Life Cycle (BLC)
* Delivery and Maintenance
  + Operational Levels Agreement (OLA) – Roles and responsibilities of individuals/team during delivery process
  + Production Implementation Plan – Product delivery steps, tech support, hardware/software components
  + Release Notes sent to team and upper management

Project Life Cycle

* Initiation – Kickoff Meeting, Concept is discussed,
* Planning and Design – Budget is made, Requirements
* Executing – Developing and Testing
* Monitoring and Controlling – Making Fixes, Then Retesting after fixes are made
* Closing – Release Notes, Delivery of Product

Types of Testing

* Functional and Regression Testing
* Smoke Testing
* User Acceptance Testing
* End to End Testing
* Integration Testing
* Unit Testing
* System Integration Testing

**Agile Methodology** – Customer collaboration over contract negotiation, responding to change over following a plan, working software over comprehensive documentation, Individuals and interactions over processes and tools

**Waterfall Methodology** – Lack of flexibility, difficulty in predicting actual needs, no scope on moving backwards or forwards in step, you can only move on when one step is over

**Jira**

* Jira is a project management tool
* Web Application
* Create a Product Backlog – List of features to be implemented/exploration of features
* Product Backlog gets made by BA or Product Owner
* Create a Sprint Backlog to keep track of features to be done from Product Backlog during a Sprint
* Used to organize Sprints
* Allows teams to organize features to be done and post when features are finished
* Allows team to communicate and view features that they are working on
* Contains User Stories/Use Cases, Defects, SCRs
* Tasks get updated and prioritized by Product Owner
* Sprint Backlog gets updated on a daily base so you can be up to date with the progress of the team
* Post bug reports for features that are not working properly
* Can view Burndown Chart for project
* Burndown Chart shows a graph representation of work done and left to be done in the sprint
* Burndown Chart shows visibility of work states/progress and is updated every day
* Burndown Chart shows transparency about the current performance

**Automation**

* Automation – Writing code to test to replace manual testing
* Selenium is an open source automation tool used exclusively for web-based applications
* Classes in Java is a blueprint from which individual objects are created
* String is a value stored into a variable using double quotations
* Int is a integer stored into a variable
* Arrays allow you to store multiple values/sets of data into a single variable
* Dynamic Array is an array that you need to define the size. Syntax: String[] xxx = new String[5];
* Array List is an array where you don’t need to define size making it more flexible than a dynamic array. Syntax: ArrayList<String> xxx = new ArrayList<>();
* A loop statement allows us to execute a statement or series of statements multiple times
* For Loop Syntax example: for(int i = 0; i < 3; i++){
* While Loop Syntax example: int i =3; while(I < 3){ …. I= I +1; }
* While loop allows you to iterate through a statement until numbers of iteration is truthful
* Conditional Statement allow you to execute a condition based on if an object exist or doesn’t
* Java method is a collection of statements that are grouped together to perform an operation. Example: captureText, mouseHover, submit, sendKeys, etc..
* Exception is an event, which occurs during the execution of a program, where any issue which make your test case stop in between the execution
* Data driven testing is preformed when there are sets of data stored in an external files such as excel and those data gets executed using tools like selenium webdriver
* Maven is a build Automation tool. It is mainly used for java projects. It makes build consistent with another project.
* Maven is also used to manage dependencies
* 3 Frameworks we used (testNG, Page Object, Cucumber)
* testNG organizes program using @beforesuite, @beforemethod, @aftermethod, @test, @aftersuite. Prioritizing is key when using testNG
* Page Object uses a base class to store elements from page classes to then later be called to use in your test case, making you test cases look a lot cleaner, shorter and flexible
* Cucumber uses feature file to list test steps from a scenario. Those steps are then later defined in code in the step definitions. Then the get tested in the test runner. It is used to organize test cases using given, when and then to show the order of the steps of the test case.
* Cross browser testing is used to perform test cases on multiple browsers to see if they perform the same or differently on different browsers
* Reusable methods are used to reduce time and work by calling a reusable method rather than repeating lines of code to perform certain commands
* Chrome Options is used to set arguments for chrome browser, like incognito, headless, maximize, etc..
* Implict wait will wait till element is present and will move on if it can’t locate element
* Explicit wait will wait till element is present but if it isn’t located it won’t continue to the next steps

Web element command allows you to store a locator from a web page into a user defined variable