What is Manual Testing

- Manual testing is performed by a human sitting in front of a computer carefully executing test steps as designed in a manual test case
- Manual Testing is the most primitive of all testing types and helps find bugs in software systems
- A new application **must** be manually tested **before** automation testing can be done
- Manual Testing requires more effort, but is necessary
- Manual Testing Goal: Ensure that the application is error free and is working appropriately based on the requirements

What is automation testing

- **Automation Testing**: The use of special software, separate from the software being tested, to control the execution of tests and the comparison of actual outcomes with predicted outcomes – Source: Google.Com ☺
  - Make Google your best friend!
- What does this definition mean??
  - Instead of a tester sitting in front of a computer executing the test steps laid out in a test case, an automation testing tool will execute the test steps, compare the expected results with the actual results, and determine if the step passed or failed.

Manual VS Automation testing

- Manual Testing and Automation Testing cover 2 vast areas
- Within each category, specific testing methods are available:
  - Black-Box & White-Box testing, Integration Testing, System Testing, Performance and Load Testing

MANUAL TESTING	AUTOMATION TESTING
Manual testing is not accurate at all times due to *human error*. Therefore, manual testing is **less reliable**	Automated testing is more reliable, as it is performed by automation tools
Manual testing is **time consuming**, taking up human resources	Automated testing is executed by software tools, so it's significantly faster than a manual approach
Investment is required for human resources	Investment is required for testing tools
Manual testing is only practical when the test cases are run once or twice, and frequent repetition is not required***	Automated testing is a practical option when the test cases are run repeatedly over a long period of time***

Testing costs money – Manual Testing Example

- A Load Test requires that 100 users access a website at the same time
- If this was done manually, it would require 100 testers, sitting in front of 100 computers, accessing the website at the same time.
  - Humans are **really good at 1 thing** – Making mistakes!
  - So, some testers may not access the website at exactly the same time, so will have to execute the test again and again
- Let's say it takes 2 hours to complete the test successfully
  - 1 Tester's Salary = $50/hour
  - $50 x 2 hours = $100
  - $100 x 100 testers = $10,000
- $10,000 for 1 test!?!!?
- Even Donald Trump doesn't have that kind of money!!!!

Testing Costs money – automation example

- A Load Test requires that 100 users access a website at the same time
- To automate this, 1 tester would create a test script using an automation load testing tool
- The test would execute in under 1 minute and the tester would have accurate results right away
- Let's say it takes the tester 2 hours to create and execute the test script
- Tester's Salary = $50/hour
- $50 x 2 hours = $100
- $100 for 1 test case
- Automation testing helps organizations save money in the long run ☺

When to use automation

- Use manual testing for the following types of testing:
  - Exploratory testing
    - Requires the tester's knowledge, experience, analytical skills, creativity, and intuition
    - We need humans to understand what and how to test since there may not be any well-defined requirements or there may not be any requirements to test against
  - Usability Testing
    - Requires the tester to measure how user-friendly, efficient, or convenient the software application is for the end users
    - Human observation is the most important factor here, so manual testing is preferred
  - Ad-hoc Testing
    - There is no specific approach to testing here
    - Completely unplanned method of testing where the understanding and insight of the tester is the only important factor
- Use Automation Testing for the following types of testing:
  - Regression Testing
    - Frequent code changes means running the same tests over and over again to check for regression
  - Load, Stress, Performance Testing
    - These types of testing require the tester to determine specific measurements for response times, server loads, etc. and so can only be done using an automation tool
  - Repeated Execution
    - Testing which requires the repeated execution of a task

Automation Testing tools

- There are literally hundreds of test automation tools on the market today
- Testing tools can be **open source** or **commercial**
- **Open Source**
  - Free to download and use
  - Limited user support – You have to rely on forums and blogs to find help for various issues with the tools
- **Commercial**
  - Cost money for licensing and use
  - Part of the cost pays for a dedicated user support system
  - Users are able to contact a help desk representative and ask for help with various issues

Open Source test automation tools

- Selenium
  - Popular Web Testing tool and it helps testers automate web browsers across different platforms
  - It uses JavaScript as its programming language
- Watir
  - Used for automating web testing and allows testers to write tests that are easy to understand and maintain
  - It uses a Java based language called Ruby
- SoapUI
  - It's a cross-platform functional testing tool that has been designed to help automatically test APIs such as SOAP and REST interfaces
- Apache Jmeter
  - It's a Java based performance testing tool
- Sahi
  - It's a Cross-platform web application testing tool, written in Java and JavaScript

Commercial test automation tools

- Ranorex
  - Allows testers to automate web application testing (among other things) and both record user interactions and play them back to execute tests
- Tellurium
  - It's a web automation tool that allows testers to design and write automated tests using plain English without any scripting or programming experience
- Squish
  - It's a GUI testing tool for various platforms, including Windows and Mac applications
  - Allows testers and developers to build automated tests using familiar scripting languages such as JavaScript, Perl, Python, and Ruby (all java based languages)
- TestComplete
  - Automated testing tool for Windows platform
  - Allows testers to record, script, and run GUI tests for applications built using different frameworks and languages (.NET, or C++)
- eggPlant
  - It's a GUI test automation tool for professional software applications and enterprise teams
  - It can be used to automate different application types such as .NET, Java, and Flash applications

Unified Functional Testing

- HPE Unified Functional Testing automates testing through an intuitive, visual user experience that ties manual, automated, and framework-based testing together in 1 IDE

- Some Key Functionalities:
  - Cross-Browser & multi-platform capabilities
    - UFT provides support for testing across various platforms such as Windows, iOS, and Android (for mobile testing)
    - Also provides support for multiple browsers including Internet Explorer, Firefox, Chrome, and Safari
  - Image-based object recognition
    - UFT learns objects as a human would, through the way they would look on the screen

- It is a Functional and Regression Testing tool which also supports some performance testing

- It is an Object-based testing tool, it performs tests based on front-end objects

- It contains a SQL Engine through which we can automate backend database testing

- It contains a VBScript Engine for scripting so we can use all VBScript features like variables, constants, operators, Built-in Functions, user-defined functions, Conditional Statements, Loop statements, and Automation objects in our tests

Version History of UFT

- Astra Quick Test – Versions 1.0 – 5.0
- In 2001, Mercury Interactive Corporation purchased Astra Quick Test, enhanced some features, and released the tool with a new name – QuickTest Professional (QTP)
  - QTP 5.6 – 2002 (now owned by MIC)
  - QTP 6.5 – 2003
  - QTP 8.0 – 2004
  - QTP 8.2 – 2005
  - QTP 9.0 – 2006
  - QTP 9.2 – 2007
- In 2007, Mercury Interactive Corp. was bought by HP
  - QTP 9.5 – 2008 (now owned by HP)
  - QTP 10.00 – 2009
  - QTP 11.00 – 2010
  - **UFT 11.50 – 2012**
  - **UFT 12.00 – 2014**
  - **UFT 12.02 – March 2015**
  - **UFT 12.50 – July 2015**
  - **UFT 14.00** – **2017** (now owned by Micro Focus – an HPE subsidiary)

UFT licensing and Os environments

- There are 2 types of licensing offered by HPE UFT
  - Seat license and Concurrent/float license
- Seat licenses are valid for each individual machine
- Concurrent Licenses are able to be used across multiple machines
- UFT is supported on Windows OS only
- UFT is not supported on UNIX or Linux Operating Systems

More on UFT's Environments

- ADD-INS
  - Add-in = additional component
  - Add-ins in terms of UFT = Environment Compatibility Files
- UFT recognizes objects in terms of Add-ins only
  - If you want to test an application that was built using Java, then the Java add-in is required
  - If you want to test an SAP application, then the SAP Add-in is required
- There are 2 types of Add-ins:
  - Internal Add-ins
  - External Add-ins

UFT Internal Add-ins

- Standard Windows (built-in)
- ActiveX
- Visual Basic
- Web
- By selecting any add-ins from the Add-in Manager window, we are telling UFT that we will be working with that type of application, and so to look out for objects that are normally found in that specific type of application
  - Example: By selecting the Web Add-in, we are telling UFT to look out for WebEdit Boxes, WebLists, Links, WebElements, etc.

UFT EXTERNAL ADD-INS

Java	.NET for Windows forms	.NET for web forms
WPF	SAP for GUI	SAP for web
PeopleSoft	Power Builder	Web Services
Siebel	Delphi	TE
Oracle	Stingray	Smalltalk
Silver Light		

What you'll learn to do with uft

- Create automation testing scripts using Record & Run and other features built into the UFT tool
- Create automation testing scripts using VBScript and Descriptive Programming by using minimal built-in features of UFT
- Create Test Automation Frameworks
  - Modular/Linear
  - Data-Driven
  - Keyword-Driven
  - Hybrid