Test Automation

The Power of Test Automation

SoftUni Team

Technical Trainers

Software University

SoftUni

Software University

sli.do

#QA-Fund

Table of Contents

Manual Testing

A Solid Starting Point

Manual Testing

- **Critical step** in the software development life cycle

- Forms the **basis** of quality assurance

- Involves the process of manually executing test cases without the use of any automation tools

- Requires testers to go through the application, step by step, to identify defects and ensure the application meets the specified requirements

The Importance of Manual Testing

- Manual testing is **essential** for ensuring the quality of user experiences

- It allows testers to validate application behavior from an end-user perspective

- Human **intuition and creativity** play a significant role in exploring edge cases and unexpected scenarios

- Particularly useful in the **early stages** of development when the application is constantly changing

- Suitable for **exploratory**, **usability**, and **ad-hoc** testing

Advantages and Limitations of Manual Testing

- **Advantages**:

  - **Flexibility** -Testers can adapt to changes quickly and design test cases based on real-time understanding

  - **User-Centric** - Manual testing focuses on user experience, identifying issues that automated tests might miss

- **Limitations**:

  - **Time-Consuming** - Executing extensive test cases manually can be time-consuming and resource-intensive

  - **Human Error** - Manual testing is susceptible to human errors, leading to inconsistencies and false negatives

Test Automation

What Lies Ahead

What Do We Automate?

- Most of the test automation involves attempts to automate tasks that are:
  - Tedious
    - Such as **regression testing**
  - Difficult to do manually
    - Such as **performance testing**
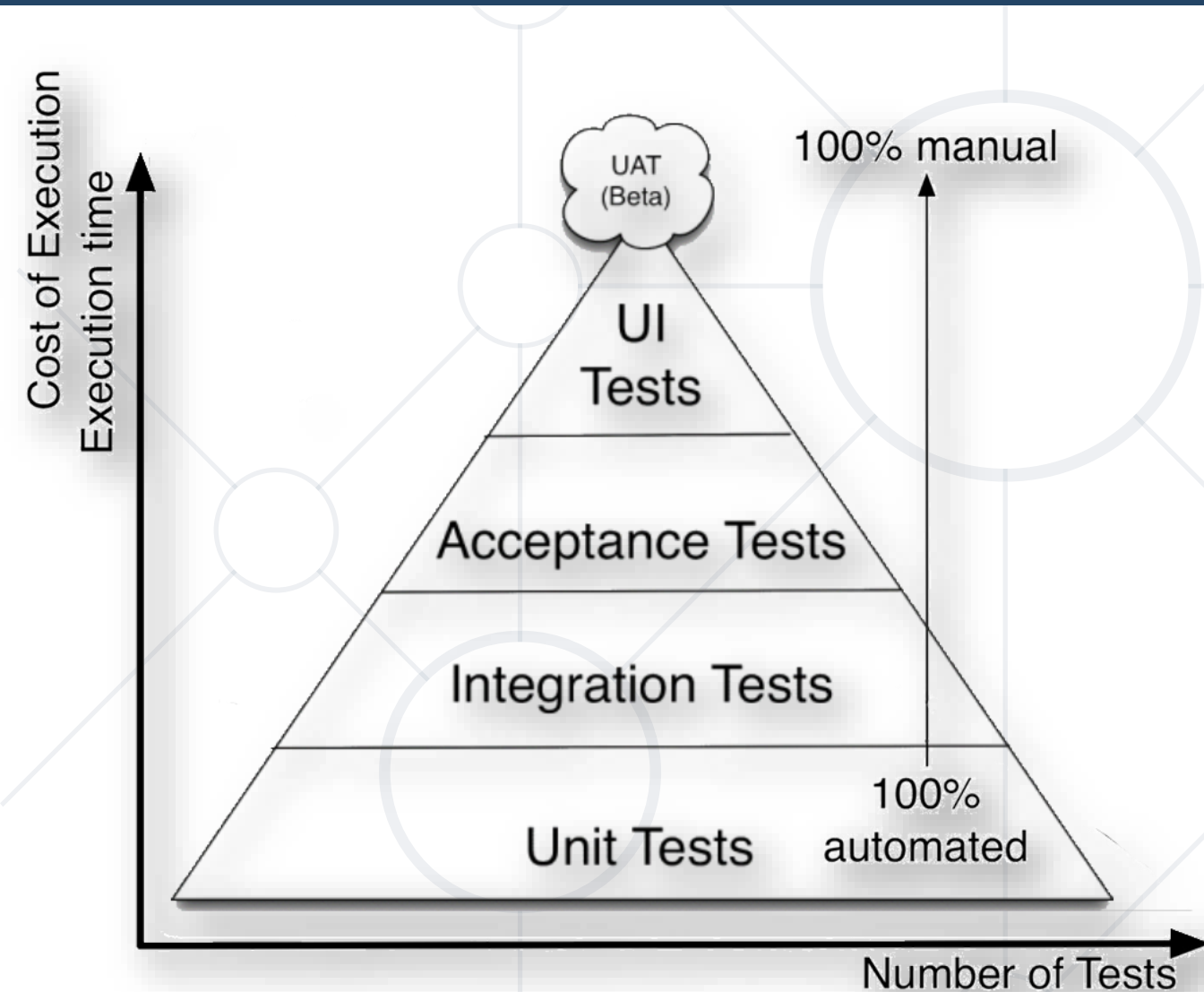- Test automation can also automate other parts of the test process

Test Automation Benefits

- Saving effort and time

- Consistency and repeatability

- Better coverage

- Objectiveness and test statistics

- Better predictability

- Customized defect reporting

- Unlimited iterations of test case execution

- Rapid feedback to developers

The Testing Pyramid

- **Unit tests**: fully automated

- **Integration tests**: fully automated

- **System tests** / **end-to-end** : automated (most of them)

- **Acceptance tests** / **UI tests**: partially automated

Myths About Automated Testing

- Automation can replace the human element

- Once automated, **cost savings** is a given

- Finds more bugs

- Every test case can be automated

- Testing can be fully automated

- One test tool is suitable for all tasks

- Automation is completely suitable for new functionality

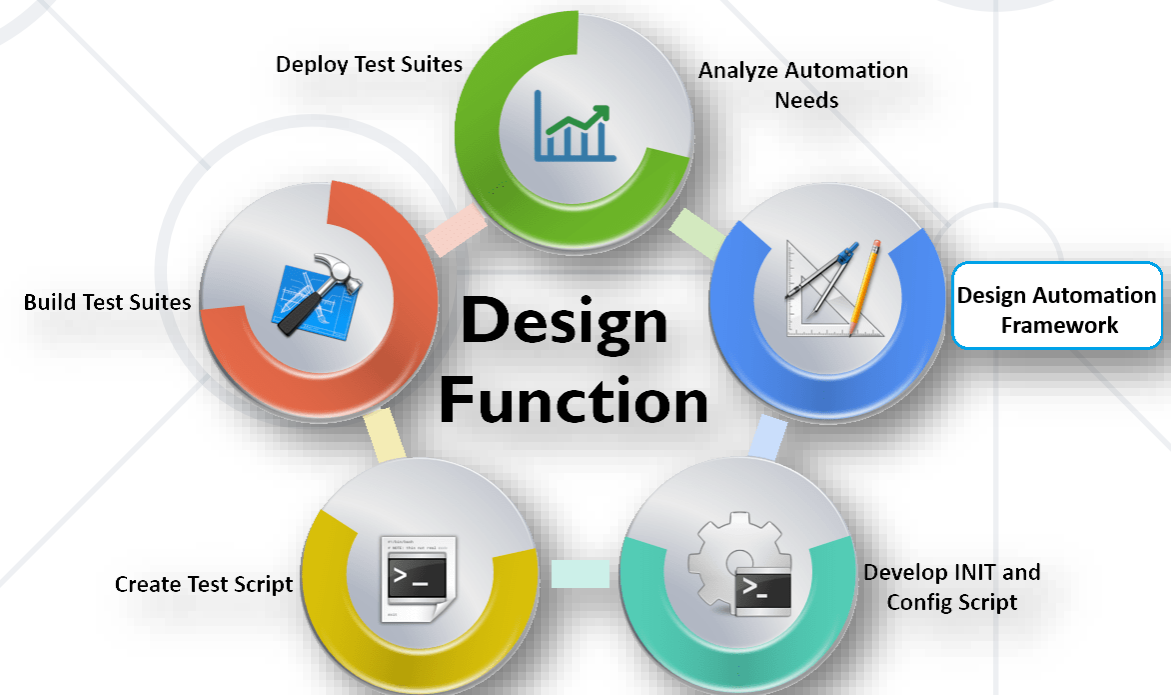# Manual Test Case => Automated Test

- The test case must be **repeatable**

- The expected results must be **consistent** and easily verified

- The test data for the manual test case must be **established**

- The **application** must be in a **stable state**

- The test case must **not** have a **dependency** on an external application that is not installed on the same test PC

- Test cases must be independent

Test Frameworks

What is Testing Framework?

- A testing framework is a **structured approach** to creating, executing, and managing software tests

- It provides **reusable components, libraries, and tools** to streamline the testing process

- Frameworks promote **efficiency, organization**, and **maintainability** of tests

Deploy Test Suites

Analyze Automation Needs

Build Test Suites

Design Function

Design Automation Framework

Create Test Script

Develop INIT and Config Script

Manual Testing Framework

- **Manual** Testing Frameworks:
  - Structure and organize manual testing activities
  - Provide guidelines for test design, execution, and reporting
  - Xray, TestRail, Zephyr

- **Automated** Testing Frameworks:
  - Facilitate the creation and execution of automated tests
  - Offer libraries, tools, and practices for scripting, execution, and reporting
  - Selenium, Cypress, Robot Framework

What is Selenium?

What is Selenium?

- A JavaScript based web testing tool

- Very popular open-source tool

- Supports testing web applications

  - On multiple browsers and operating systems

- Tests run directly in browser

- You can find the documentation here

  - **https://www.selenium.dev**

Selenium Tools

- Selenium is a collection of distinct software tools, each tailored to fulfill specific roles in the realm of test automation

- These tools collectively contribute to a comprehensive suite for efficient testing of web applications

  - **Selenium IDE**

  - **Selenium WebDriver**

  - **Selenium-Grid**

Selenium IDE

Record and Play Tests

Selenium IDE

- **Selenium IDE**, short for Integrated Development Environment

- User-friendly **browser extension**

- Simplifies the creation and execution of automated test scripts through a **record-and-playback** mechanism

- Suitable for those with **limited programming experience**

- Accessible **entry point** into test automation

- Limited scope

Test User Login

- **Test Case:** Test_User_Login

- **Description:** Verify that a registered user can log in to the Eventures web application

- **Steps:**

  - Go to the "Login" page on the Eventures web application

  - Fill in the login form with valid credentials (username and password)

  - Click on the "Log in" button

- **Expected Results:** The user is successfully logged in, and a welcome message containing the username is displayed on the Home page

Record and Run User Login with Selenium IDE

Selenium WebDriver

Programmatic approach

Selenium WebDriver

- Most **important component** of Selenium Suite
- Powerful and versatile framework for automating web browsers
- Programmatic control over web elements
- Precise test scenarios across different browsers
- Flexibility and support for multiple programming languages
- Seamlessly integrates into continuous integration and continuous delivery pipelines

Test User Login

```csharp
[Test, Order(2)]
 | 0 references
public void Test_User_Login()
{
    // Arrange: go to the "Login" page
    driver.Navigate().GoToUrl(this.baseUrl + "/Identity/Account/Login");

    // Locate fields and fill them in with valid credentials
    driver.FindElement(By.Id("Input_Username")).SendKeys(username);
    driver.FindElement(By.Id("Input_Password")).SendKeys(password);

    // Act: locate and click on the "Login" button
    driver.FindElement(By
        .XPath("//button[@type='submit'][contains(.,'Log in')]"))
        .Click();

    // Assert user is redirected to the "Home" page and is logged in
    Assert.AreEqual(this.baseUrl + "/", driver.Url);
    Assert.That(driver.PageSource.Contains($"Welcome, {username}"));
}
```

Test run finished: 17 Tests (17 Passed, 0 Failed, 0 Skip ⚠ 0 Warnings ❌ 0 Errors

Test ▲	Duration
◢ ✅ Eventures.WebApp.SeleniumTests (17)	4.1 sec
◢ ✅ Eventures.WebApp.SeleniumTests (17)	4.1 sec
◢ ✅ SeleniumTestsEvents (10)	2.8 sec
✅ Test_AllEventsPage_ThroughNavigation	205 ms
✅ Test_CreateEventPage_BackToListLink	166 ms
✅ Test_CreateEventPage_CreateEvent_InvalidData	204 ms
✅ Test_CreateEventPage_CreateEvent_ValidData	522 ms
✅ Test_CreateEventPage_ThroughNavigation	118 ms
✅ Test_DeleteEvent	437 ms
✅ Test_EditEvent_InvalidData	489 ms
✅ Test_EditEvent_ValidData	487 ms
✅ Test_HomePage_AllEventsLink	86 ms
✅ Test_HomePage_CreateEventPageLink	92 ms
◢ ✅ SeleniumTestsUser (7)	1.3 sec
✅ Test_HomePage_LoginPageLink_InNavigation	91 ms
✅ Test_HomePage_LoginPageLink_OnPage	88 ms
✅ Test_HomePage_RegisterPageLink_InNavigation	107 ms
✅ Test_HomePage_RegisterPageLink_OnPage	103 ms
✅ Test_User_Login	254 ms
✅ Test_User_Logout	97 ms
✅ Test_User_Register	584 ms

Selenium Grid

- Run tests on different machines against different browsers in parallel
  - Run tests **simultaneously** on different machines
- Follows the **Hub-Node Architecture** to achieve parallel execution of test scripts
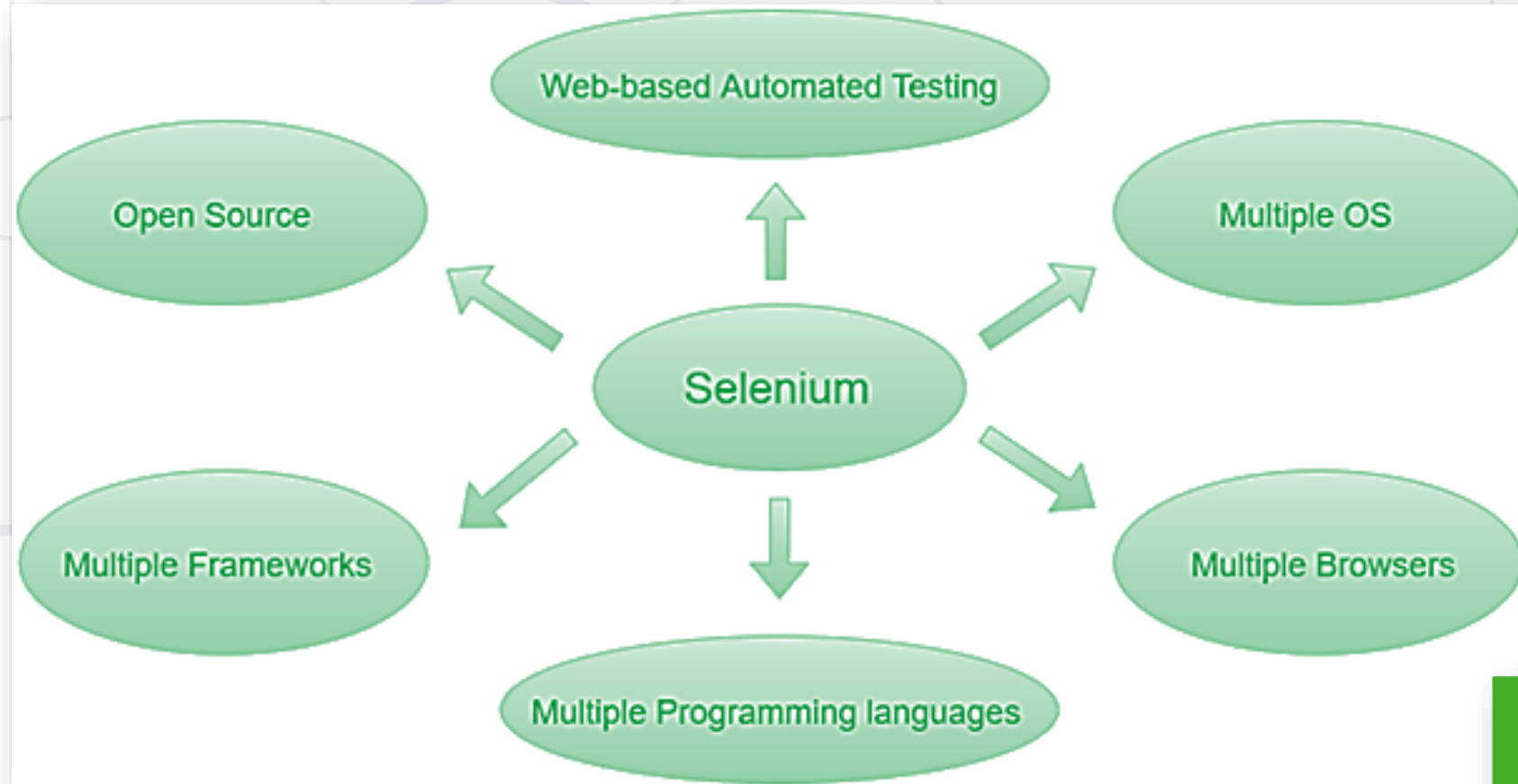 - The Hub is considered as master of the network and the other are the nodes

Selenium Stack

- Supports variety of platforms such as
  - Windows, Linux, Solaris and Macintosh
- Languages supported include
  - C#, Java, JavaScript, Perl, Python, Ruby, Kotlin
    - Most popular with Java and C#
- Browsers supported by Selenium include
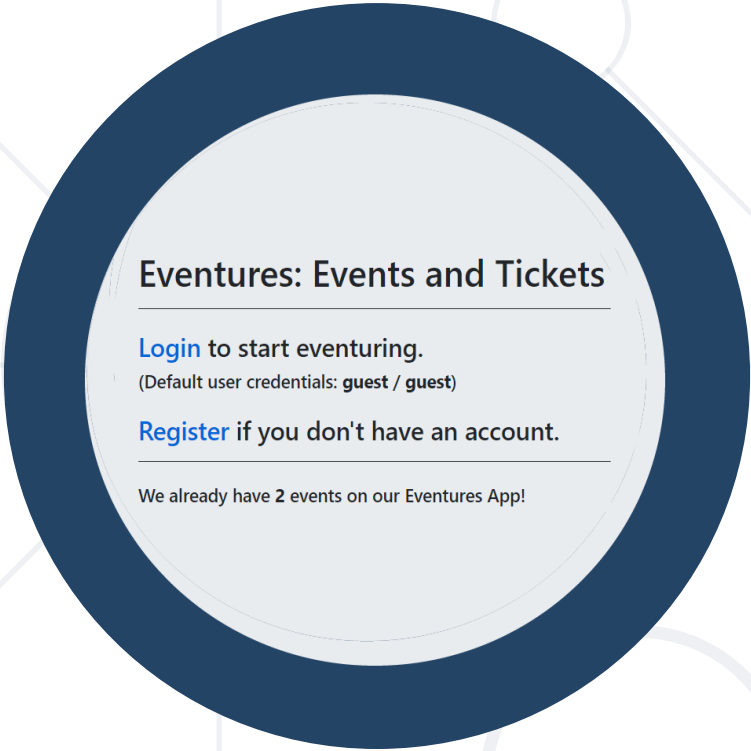 - Internet Explorer/Edge, Mozilla Firefox, Google Chrome and Safari

All about Selenium

Selenium Features

- Open source

- Record and playback

- Export functionality

- Parallel execution

- Integration with Unit testing framework

- Requires fewer resources

- No need for server installation

Automation Testing

Selenium and GitHub Actions Demo

Eventures – Automation Testing

- https://github.com/QA-Automation-Testing-Demo/Eventures-3.0

WebApp-tests
succeeded 3 minutes ago in 51s

> ✓ Set up job
> ✓ Run actions/checkout@v2
> ✓ Set up Node.js
> ✓ Run dotnet restore Eventures.WebApp.SeleniumTests
> ✓ Run dotnet build Eventures.WebApp.SeleniumTests --no-restore
> ✓ Run dotnet test Eventures.WebApp.SeleniumTests --no-build --verbosity normal
> ✓ Post Set up Node.js
> ✓ Post Run actions/checkout@v2
> ✓ Complete job

```
NUnit Adapter 4.5.0.0: Test execution complete
  Passed Test_User_Register [579 ms]
  Passed Test_User_Login [249 ms]
  Passed Test_User_Logout [75 ms]
  Passed Test_AllEventsPage_Anonymous [30 ms]
  Passed Test_HomePage_LoginPageLink_InNavigation [70 ms]
  Passed Test_HomePage_LoginPageLink_OnPage [76 ms]
  Passed Test_HomePage_RegisterPageLink_InNavigation [87 ms]
  Passed Test_HomePage_RegisterPageLink_OnPage [78 ms]

Test Run Successful.
Total tests: 18
     Passed: 18
 Total time: 19.7891 Seconds

Build succeeded.
    0 Warning(s)
    0 Error(s)

Time Elapsed 00:00:21.11
```

The QA Curriculum

Become a QA Automation Engineer and Start a Job

Curriculum

Entry Module

QA Basics ✔

1 month

Fundamentals Module

QA Fundamentals ✔

4 months

Professional Modules

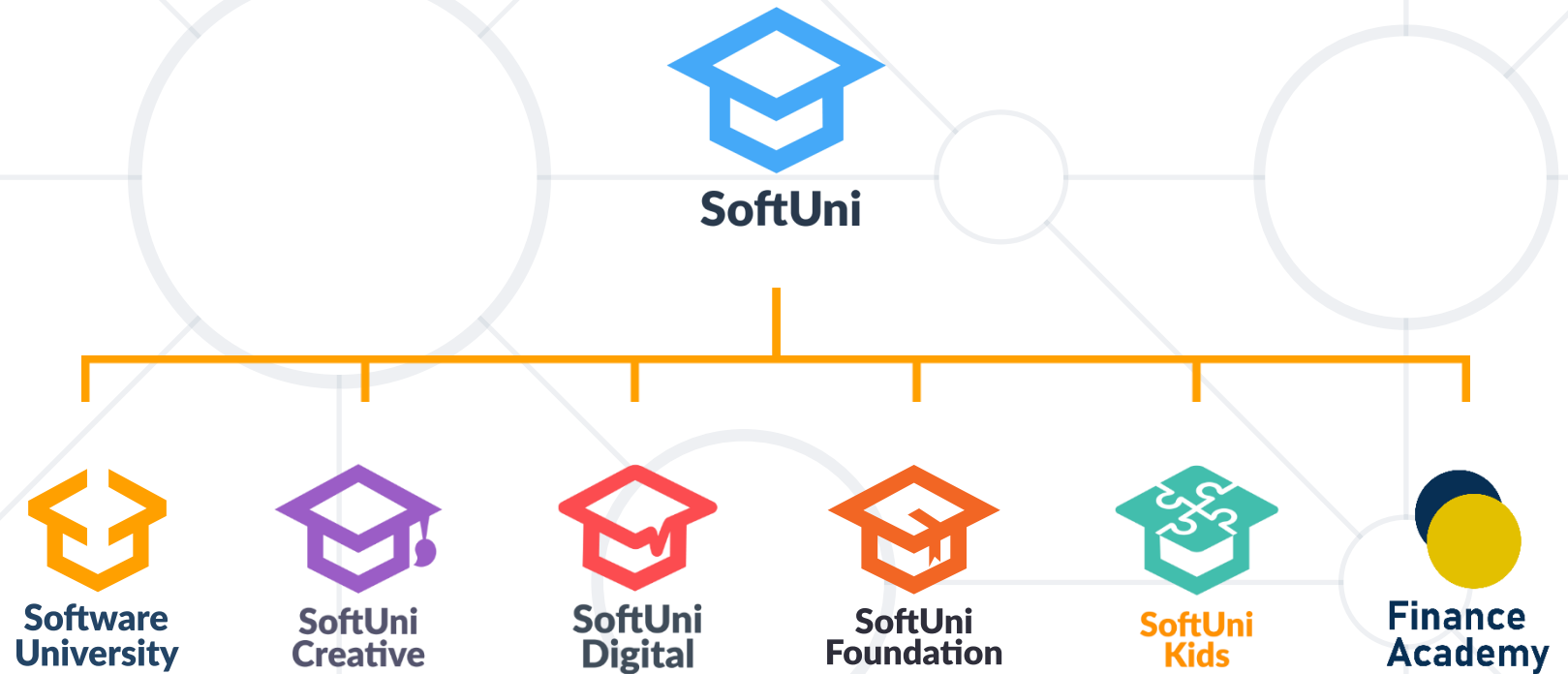| Programming for QA | Back-End Test Automation | Front-End Test Automation |

12 months

https://softuni.bg/qa/curriculum

Summary

- Manual Testing

- What is **Test Automation**?

- What is **Test Framework**?

- What is **Selenium**?

- Automation Testing in Practice

Questions?

SoftUni Diamond Partners

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
    - softuni.bg, about.softuni.bg

- Software University Foundation
    - softuni.foundation

- Software University @ Facebook
    - facebook.com/SoftwareUniversity

License

- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**

- Unauthorized copy, reproduction or use is illegal

- © SoftUni – https://about.softuni.bg/

- © Software University – https://softuni.bg