



TEST AUTOMATION PRACTICE WITH SELENIUM WEBDRIVER

**Benkó Adrián
Bondár Tamás**

EPAM Systems, Budapest, 2019

Final Exercise

- There will be one exercise given to you at the end of our 4th session
- Rating: 10 points as a theoretical maximum can be given as per the followings:
 - 0 point: solution was not provided, finished only the parts of the problems, the solution does not compile
 - 4 points: the exercise is fully solved, the test is green
 - +2 points for using the Page Object Model
 - +2 points for using the good practices mentioned during the course
 - +2 points for using Data Driven testing with separate test data source

Agenda

- 1 Automated Web Testing in general
- 2 Record and Replay
- 3 Introduction to Selenium
- 4 Environment setup
- 5 Assertions
- 6 Navigations



CHAPTER 1

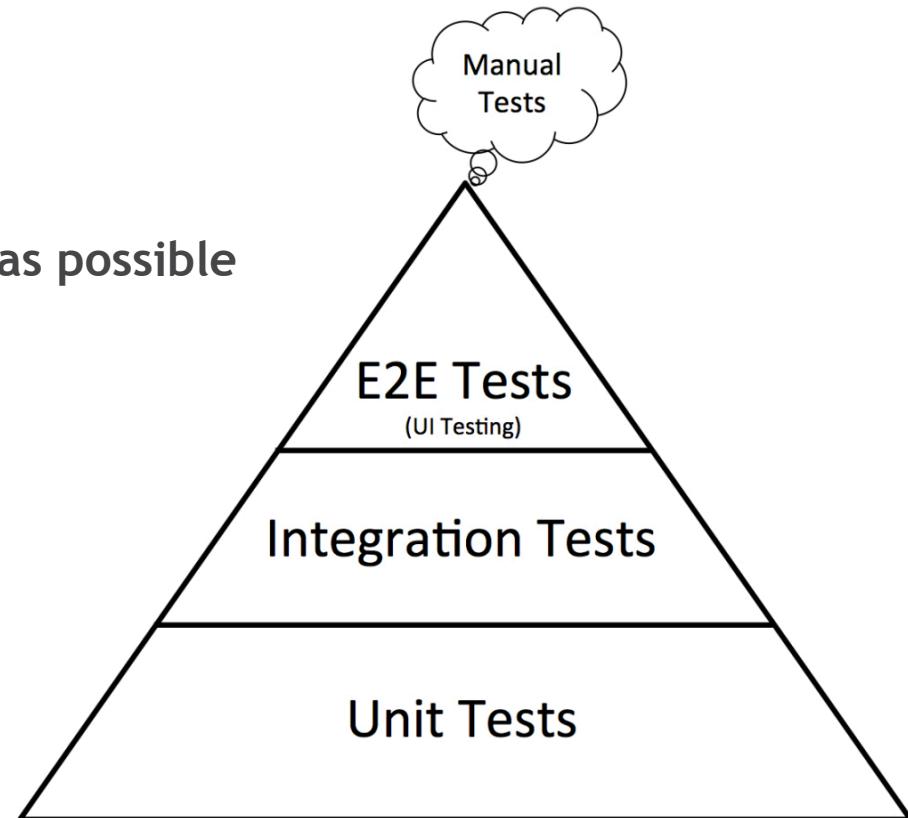
AUTOMATED WEB TESTING IN GENERAL

Automated (web) testing

- Using a software tool to run **repeatable** tests

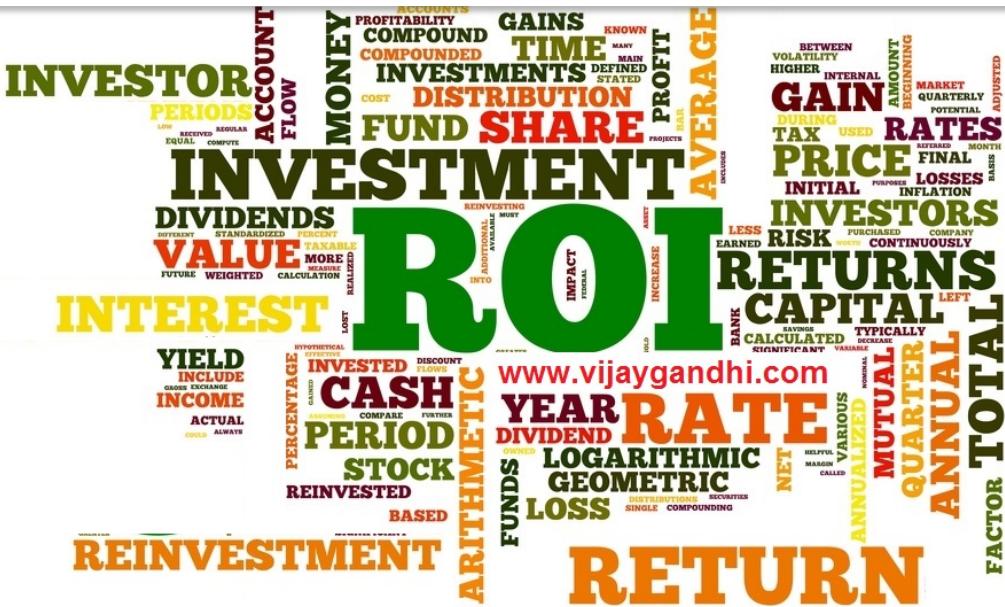
against the application to be tested

- Necessary for regression testing
- Simulate end user behavior as much as possible
 - Record and Playback
 - Language -> Driver -> Browser
- Plays nicely with TDD, BDD software development process



Why/When automate?

- Frequent regression testing
 - Rapid feedback to developers
 - Support for Agile
 - Disciplined documentation of test cases
 - Finding defects missed by manual testing



When **NOT** to automate?

- if the application's user interface will change considerably in the near future, then any automation might need to be rewritten anyway
- automation for images, texts, ads, overall outline of page might be not be worthwhile
- sometimes there is simply not enough time to build test automation.
- For the short term, manual testing may be more effective



CHAPTER 2

RECORD AND REPLAY

Record and Replay

- Record exact user interactions previously performed by the tester
- Easy to create test cases
- Ineffective repetition
- Limited options
- Limited verification capabilities
- e.g. **Selenium IDE**, iMacros, QTP, TestComplete



DEMO – SELENIUM IDE

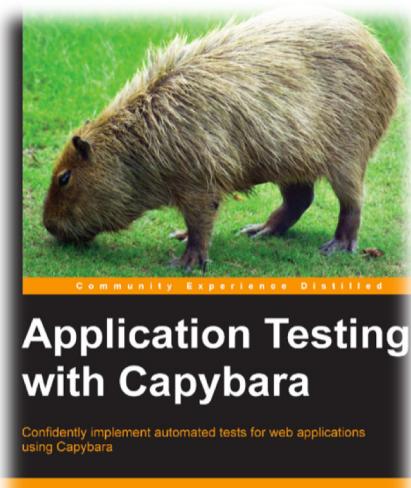
CHAPTER 3

SELENIUM WEBDRIVER

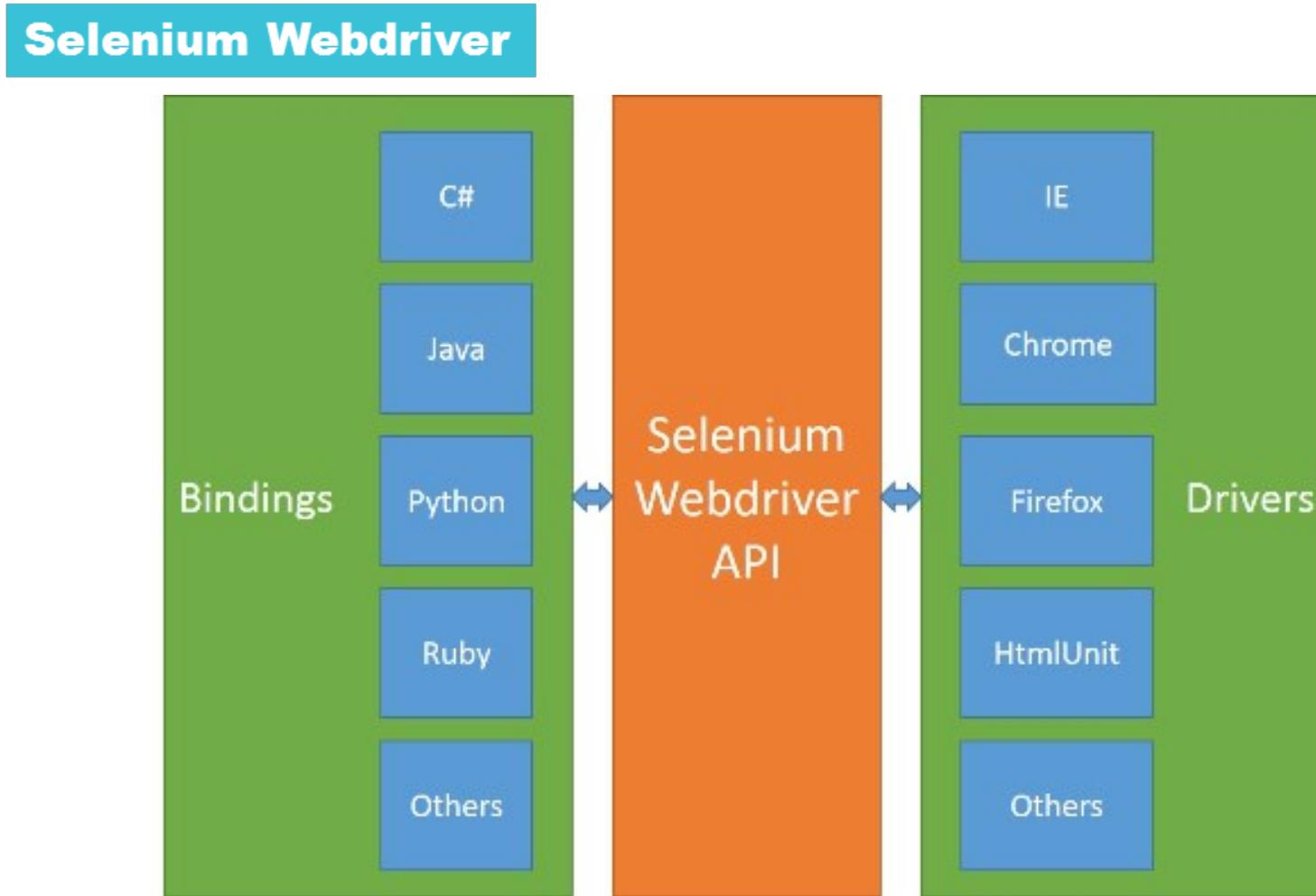
Script->Driver->Browser

- **Selenium Webdriver**

- Watir, Capybara
- WebDriverIO



Script->Driver->Browser structure



Browsers



HtmlUnit

GUI-Less browser for Java programs

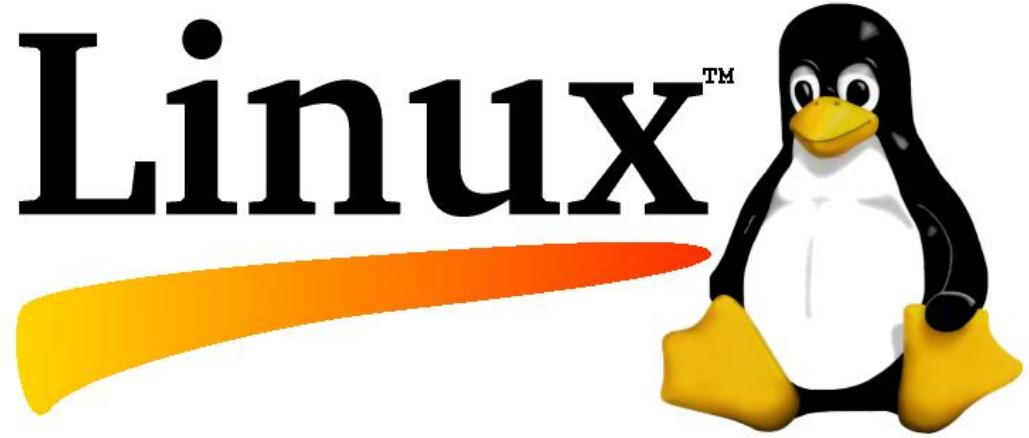
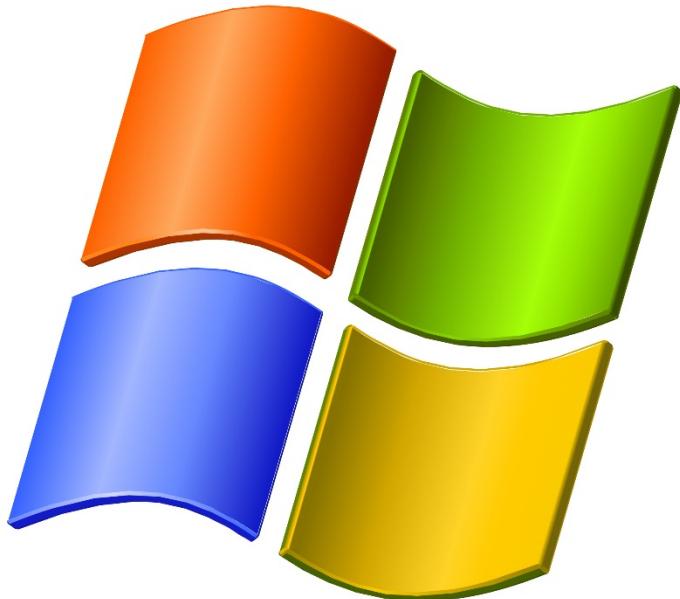


Challenges – Browser differences

- Major functionality is consistent but there can be subtle differences in:
 - Browser size
 - Screenshots
 - Element visibility
 - Performance
 - Native events
 - Cookie management
- IE is the worst offender



Supported OS



iOS

Supported Programming Languages



JavaScript



Chapter 4

ENVIRONMENT SETUP

Eclipse setup

- [Download eclipse](#)
- Deploy eclipse: use administrator mode
- Open project:
 - File -> Import -> Existing Maven Projects -> Next
 - Root Directory -> Browse ->
TestAutomation\webdriver_ui_testing\java\source\2019_autumn_uni\ -> Choose the pom -> Finish
- Window -> Preferences -> General -> Workspace -> change character encode to UTF-8
- Run test

CHAPTER 5

ASSERTION

Assertion

Fail method

`fail(error_message)`

Conditional assert

`assertTrue(error_message, boolean_condition)`

Equality assert

`assertEquals(error_message, expected, actual)`

Provide meaningful messages in assertions!

Assertion

Identity assert

```
assertSame(error_message, expected_object, actual_object)
```

Custom assert

```
assertThat(actual_object, Matcher<object> matcher)
```

String assert

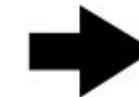
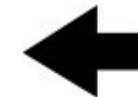
```
assertThat("myString", containsString("ring"))
```

CHAPTER 6

NAVIGATION

What can you do with a browser?

- navigate to url



- get title, get url



- navigate back/forward



- refresh page



Navigation

Loading a web page in current browser window

- `driver.get(java.lang.String)`
- `driver.navigate().to(java.lang.String)`
- `driver.navigate().to(java.net.URL)`



Navigation

Move back & forward

- `driver.navigate().back()`
- `driver.navigate().forward()`

Refresh page

- `driver.navigate().refresh()`



Questions

