Introduction to Software Testing

Lecture 11

Testing GUI Applications

Instructor: Morteza Zakeri

Initial version of slides by: **Fatemeh Bakhshi**

Completed and Modified by: **Morteza Zakeri**

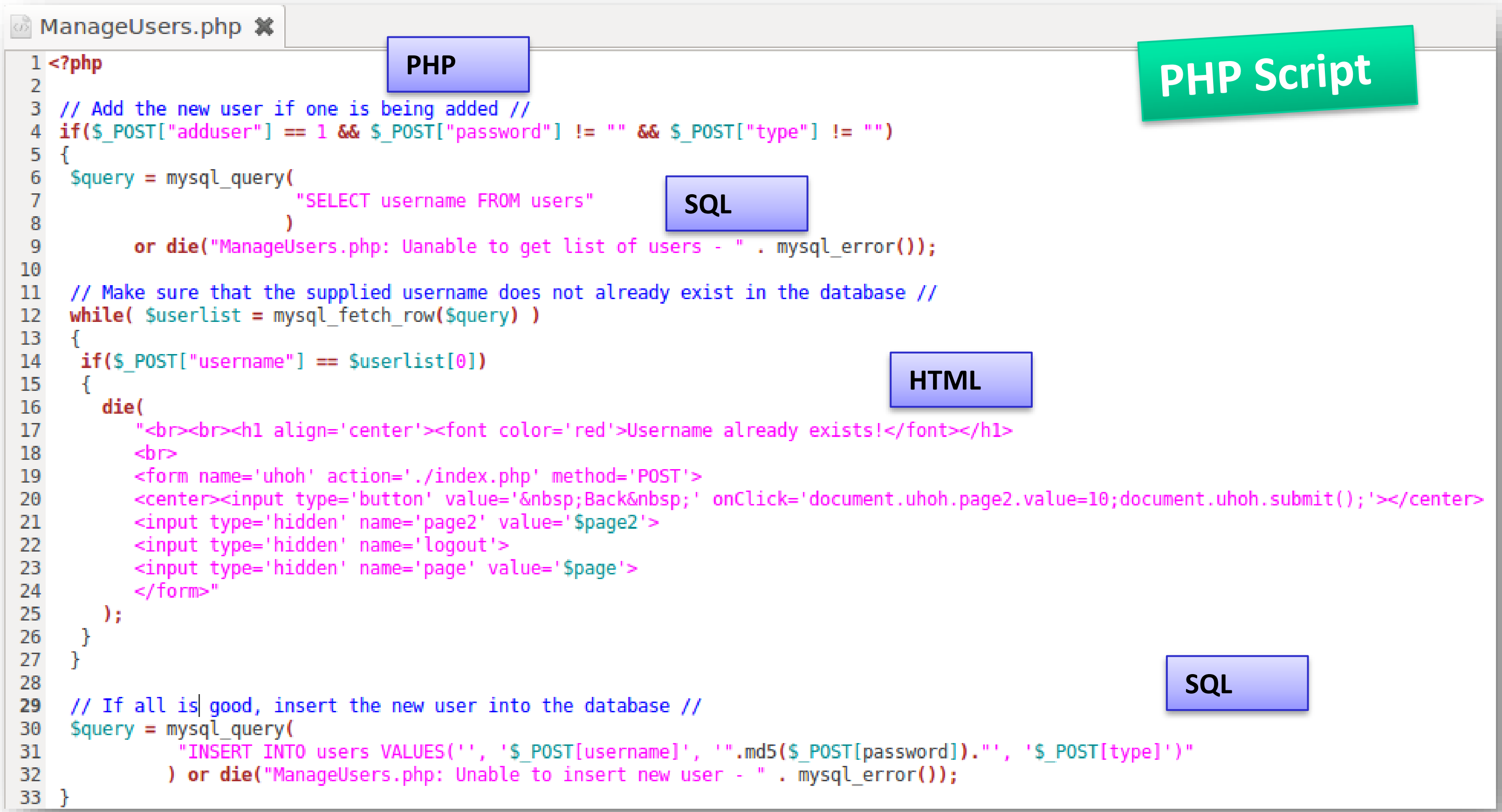
March 2024

What is GUI Testing

- Graphical User Interface (GUI) Testing
- Methods used to identify and conduct GUI tests, including the use of automated tools.
- How to test a GUI?
    - Manual: Resource intensive and Unreliable
    - Automated: Systematic method of test case generation
- Elements of GUI Testing:
    - A process
    - A GUI Test Plan
    - A set of supporting tools
- Most frequent GUI applications
    - **Web applications**

Uniqueness 1: Heterogeneous system

- Server side
  - Can be written in PHP, Java, C#...
  - Communicate with Database server in SQL



```php
<?php

 // Add the new user if one is being added //
 if($_POST["adduser"] == 1 && $_POST["password"] != "" && $_POST["type"] != "")
 {
  $query = mysql_query(
                       "SELECT username FROM users"
                      )
        or die("ManageUsers.php: Uanable to get list of users - " . mysql_error());

  // Make sure that the supplied username does not already exist in the database //
  while( $userlist = mysql_fetch_row($query) )
  {
   if($_POST["username"] == $userlist[0])
   {
     die(
        "<br><br><h1 align='center'><font color='red'>Username already exists!</font></h1>
        <br>
        <form name='uhoh' action='./index.php' method='POST'>
        <center><input type='button' value='&nbsp;Back&nbsp;' onClick='document.uhoh.page2.value=10;document.uhoh.submit();'></center>
        <input type='hidden' name='page2' value='$page2'>
        <input type='hidden' name='logout'>
        <input type='hidden' name='page' value='$page'>
        </form>"
     );
   }
  }

  // If all is good, insert the new user into the database //
  $query = mysql_query(
            "INSERT INTO users VALUES('', '$_POST[username]', '".md5($_POST[password])."', '$_POST[type]')"
           ) or die("ManageUsers.php: Unable to insert new user - " . mysql_error());
 }
```

Uniqueness 2: Dynamic pages

- Client page is dynamic
  - It can change itself in the runtime
  - HTML can be modified by JavaScript
  - JavaScript can modify itself
  - Demo

- Server script is dynamic
  - Client pages are constructed in the runtime
  - A same server script can produce completely different client pages
  - Demo
    - SchoolMate

Uniqueness 3: Performance

- Performance is crucial to the success of a web app
  - Recall the experience to register for a class in the first days of the semester…
  - Are the servers powerful enough?

- Performance testing evaluates system performance under normal and heavy usage
  - Load testing
    - For **expected** concurrent number of users
  - Stress testing
    - To understand the upper **limits** of capacity
- Performance testing can be automated (**Lecture 12**)

Uniqueness 4: Security

- Web applications usually deal with sensitive info, e.g.,
  - Credit card number
  - SSN
  - Billing / Shipping address

- Security is the biggest concern

- **Security** or **penetration testing** should simulate possible attacks
- Security testing can be automated
  - Will be discussed in Lectures 13 and 14

Uniqueness 4: Security

- SQL Injection
  - The untrusted input is used to construct dynamic SQL queries.
  - E.g., update my own password

PHP Script

```
$str = "UPDATE users SET password = \" " . $_POST['newPass'] .
        "\" WHERE username =\"" . $_POST['username'] . "\"";
mysql_query( $str );
```

Normal Case

```
$_POST['newPass'] = pass, $_POST['username'] = me

Query String: UPDATE users SET password = "pass"  WHERE username ="me"
```

Attack

```
$_POST['newPass'] = pass, $_POST['username'] = " OR 1=1 --

Query String: UPDATE users SET password = "pass"  WHERE username ="" OR 1=1 --"
```

Uniqueness 4: Security

- Cross Site Scripting (XSS)
  - The untrusted input is used to construct dynamic HTML pages.
  - The malicious JS injected executes in victim's browser
  - The malicious JS can steal sensitive info

- Solution: Never trust user inputs
- **Design test cases to simulate attacks.**

User Interface (UI) functional tests

- Run on an actual browser
- Mimic the user actions in the application.
- Give tester feedbacks on the integration between multiple components
  - Services, the UI, and the DB.
- Are macro-level
- Focus on validating all the critical user flows
  - One example of a critical user flow in the ecommerce application is searching for a product, adding the product to the cart, paying for the product, and getting an order confirmation.

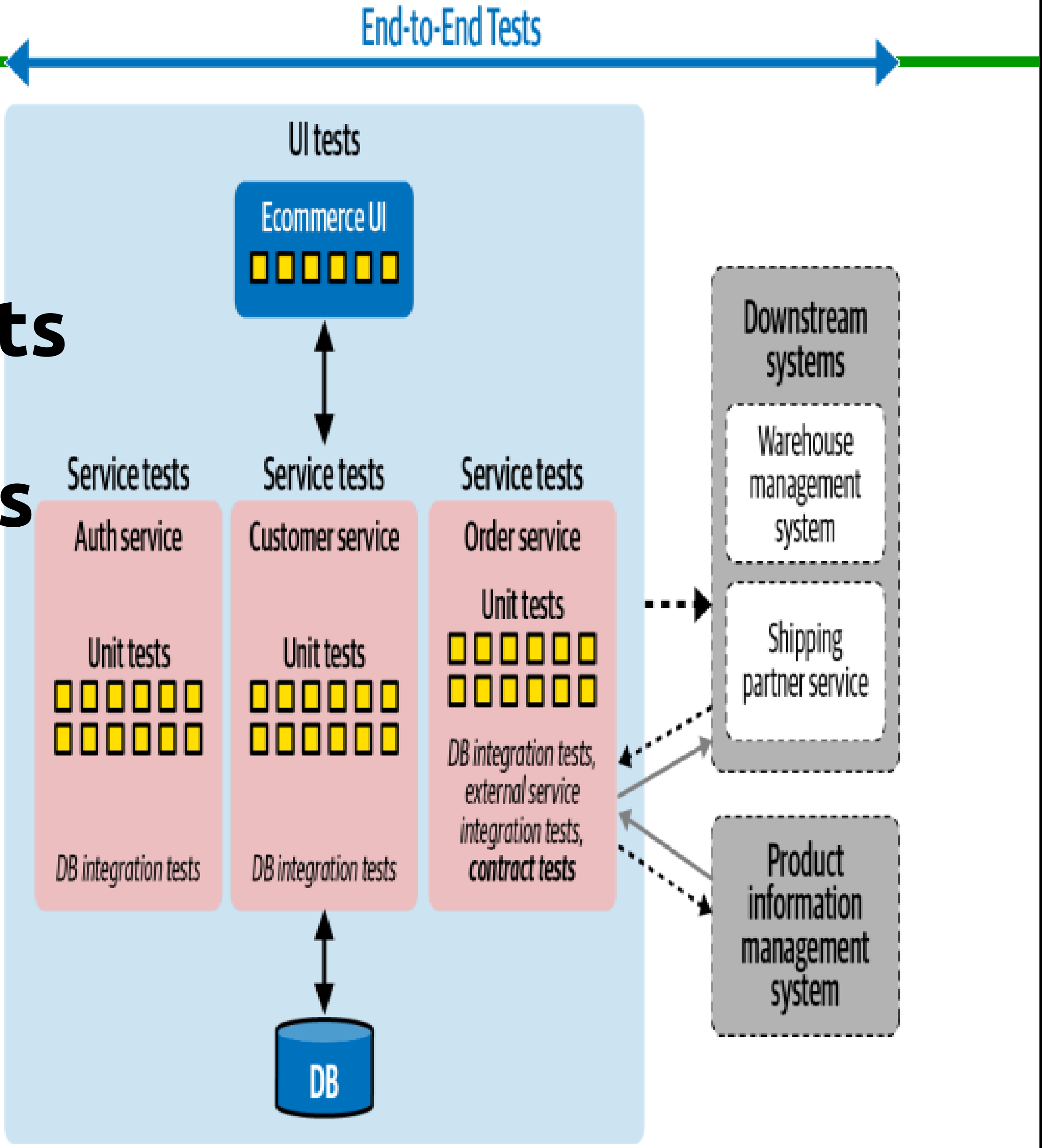
UI functional tests vs. Unit tests

- UI functional tests are **macro-level** tests.
- Should focus on validating all the critical user flows.
- When writing such tests, avoid validating the same details covered as part of the **lower-level micro tests** again.
- This will be redundant and increase their execution time.
- For instance, verifying the **order totals** for different combinations of item prices should be covered by unit tests and need not be verified again as part of a UI functional test.
- UI functional tests are usually kept apart from the application code as a separate codebase.
- Tools like Selenium and Cypress are popularly adopted to write automated UI tests.

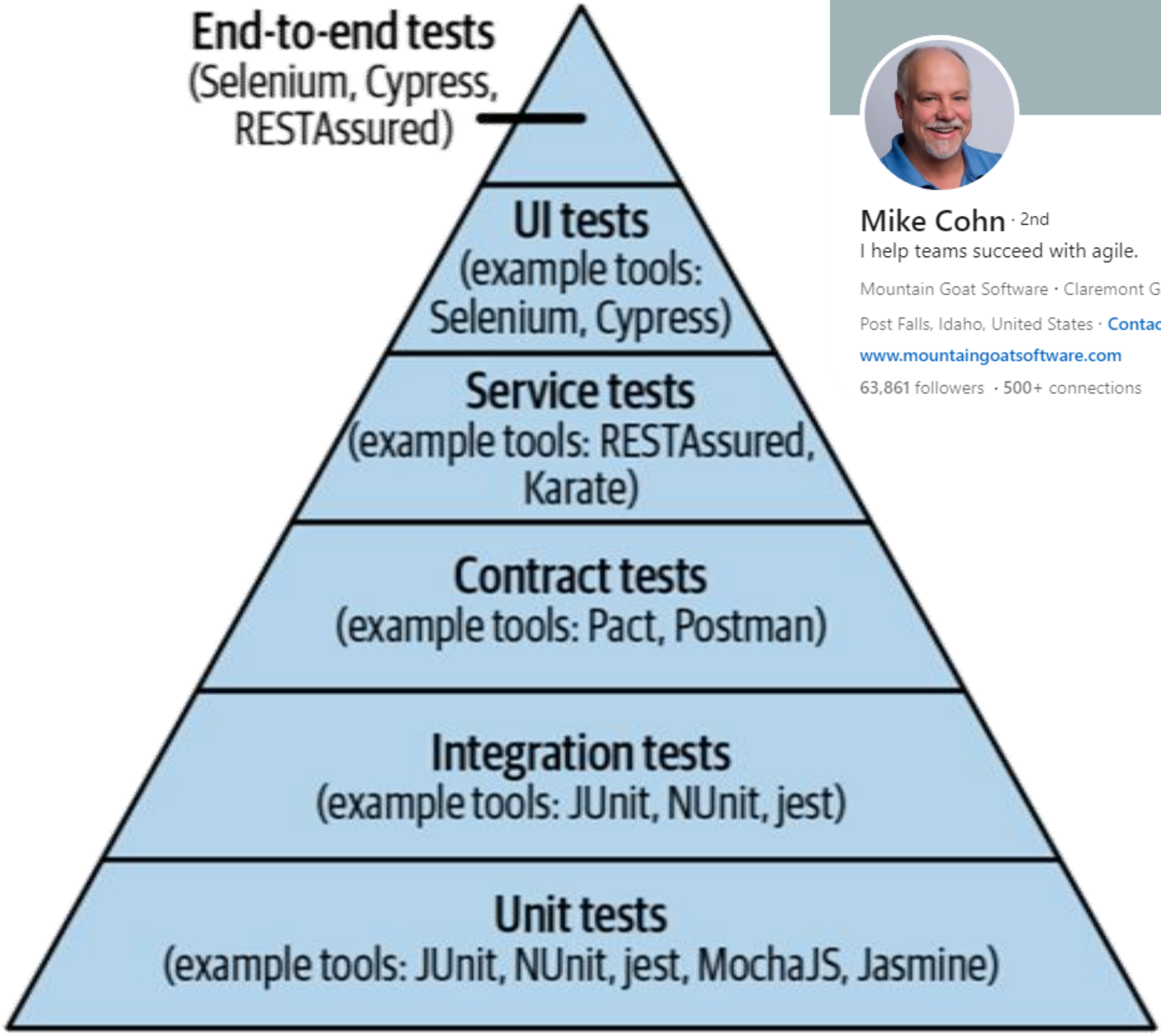
UI functional tests vs. End-to-end tests

- End-to-end tests should validate the entire breadth of your domain workflow, including downstream systems.
- Mimic the user actions in the application.
- Depending on the application context, the UI functional tests **often tend to become end-to-end tests**.
- If not, create separate end-to-end tests using a combination of UI, service, and DB testing tools to cover the entire integration flow.
  - System and integeration testing
- These tests take the longest time to run and require more care in maintaining, as they need a stable environment and test data setup across various systems.
- Just setup a few **test scenarios** that will activate all your components.

UI functional tests
VS.
End-to-end tests
VS.
Unit tests



Test pyramid for a (service-oriented) web application


End-to-end tests
(Selenium, Cypress, RESTAssured)
UI tests
(example tools: Selenium, Cypress)
Service tests
(example tools: RESTAssured, Karate)
Contract tests
(example tools: Pact, Postman)
Integration tests
(example tools: JUnit, NUnit, jest)
Unit tests
(example tools: JUnit, NUnit, jest, MochaJS, Jasmine)

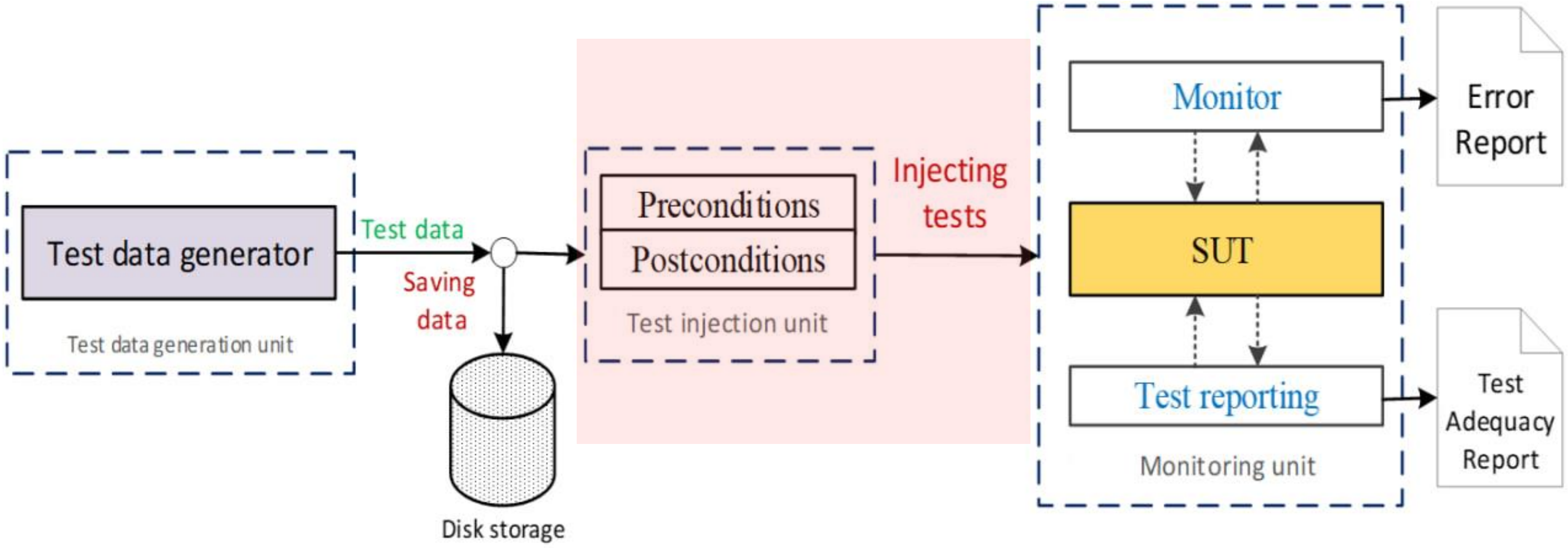
Mike Cohn · 2nd
I help teams succeed with agile.
Mountain Goat Software · Claremont Graduate University
Post Falls, Idaho, United States · Contact info
www.mountaingoatsoftware.com
63,861 followers · 500+ connections

Automated tests vs. Manual tests

Manual tests	Automated tests
Requires human intervention.	Use tools to inject and execute tests.
Requires skilled labor, long time and costs.	Save manpower, times, and cost
Any type of application can be tested.	Recommended only for stable systems.
Can become repetitive and boring.	Boring part of execution handled by tools.

Test automation

- Test data generation
- →**Test injection**
- Monitoring and report



Test from the front end

- Good things
  - Hide the complexity of the backend
  - Uniformed interface
  - Can put a **robot** in the front end and automate the tests

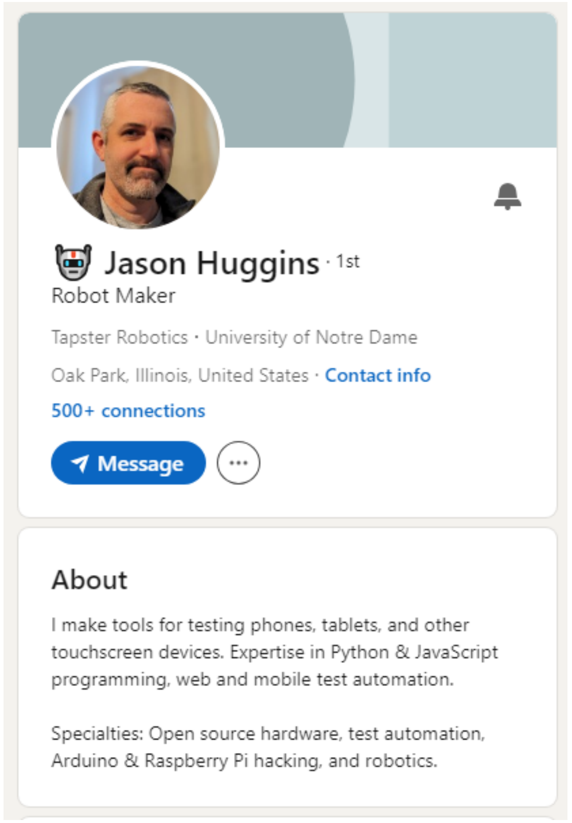
- ❑ Bad things
  - The **front end** is not trustable
    - Crafted malicious requests
    - Front end limits the length of the input values
    - Front end limits the content of the input values
    - Front end limits the combination of the input values

Good things of testing from the front end

- Automated web app testing
  - Compare to commend-line program testing…
    - **GUI testing**: event driven (the difference)
      - "Button A" then "Button B" → **OK**
      - "Button B" then "Button A" → **FAIL**
    - Sensitive to input values (the same)
  - The **robot** should be able to
    - **Provide input values**
    - Simulate user actions

Selenium

- An **open source automated testing tool** for web application.
- Jason Huggins building the Core mode as "JavaScriptTestRunner" in 2004.
- Paul Hammant saw the demo, and started discussions about the open sourcing of **Selenium**.



Jason Huggins · 1st
Robot Maker
Tapster Robotics · University of Notre Dame
Oak Park, Illinois, United States · Contact info
500+ connections

About
I make tools for testing phones, tablets, and other touchscreen devices. Expertise in Python & JavaScript programming, web and mobile test automation.

Specialties: Open source hardware, test automation, Arduino & Raspberry Pi hacking, and robotics.

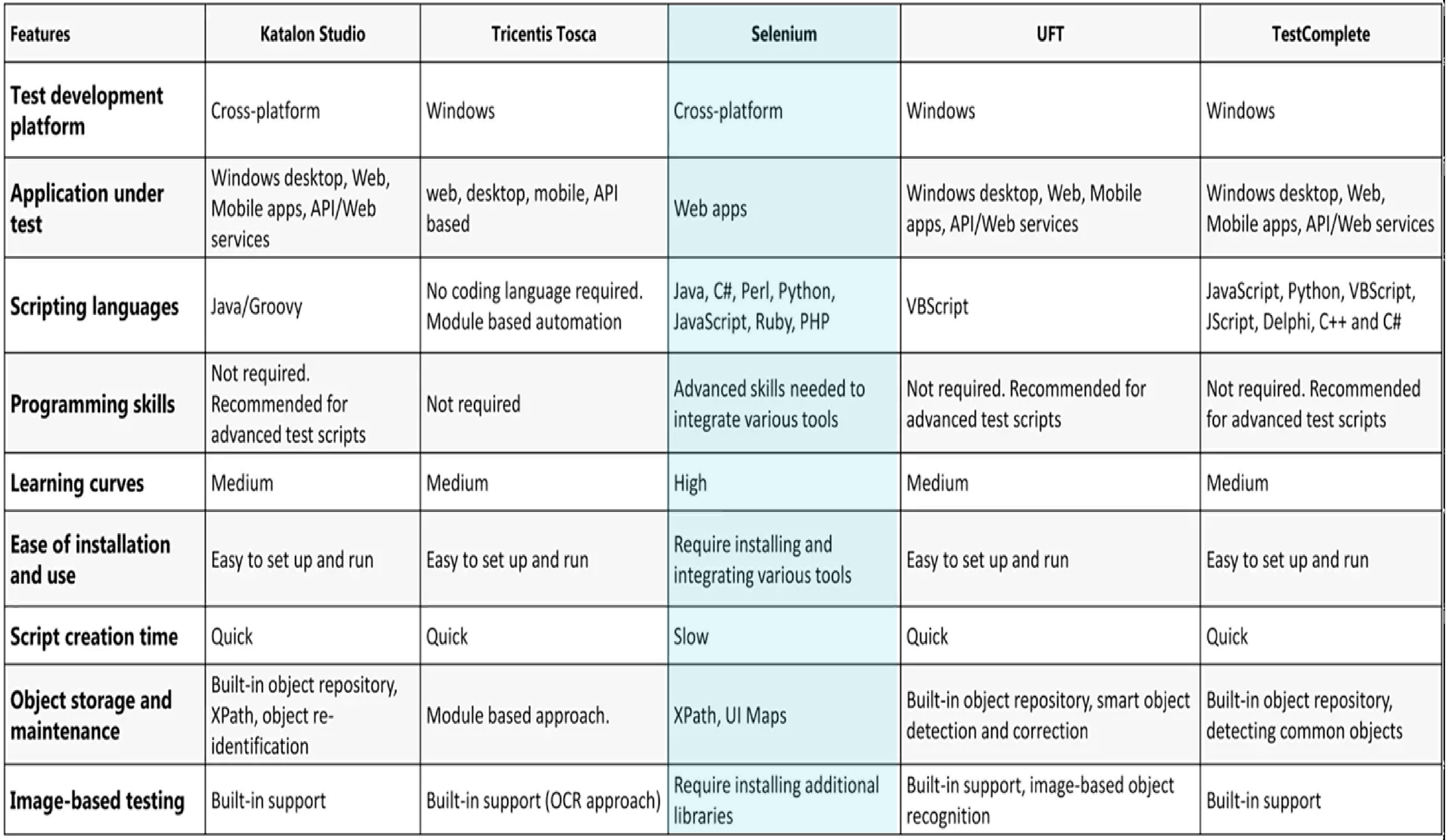
What Selenium can do

- A **solution** for the automated testing
  - Simulate user actions as a **web bot** (This lecture)
  - Functional testing (This lecture)
    - Create regression tests to verify functionality and user acceptance.
  - Performance Testing (Lecture 12)
    - Load testing
    - Stress testing
  - Browser compatibility testing
    - The same script can run on any Selenium platform

Selenium main features

- Open source.
- Supports many languages (Python, C#, Javascript, …).
- Suppurts many browsers (Firefox, Chrome, …).
- Supports many operating systems.
- Flexible.

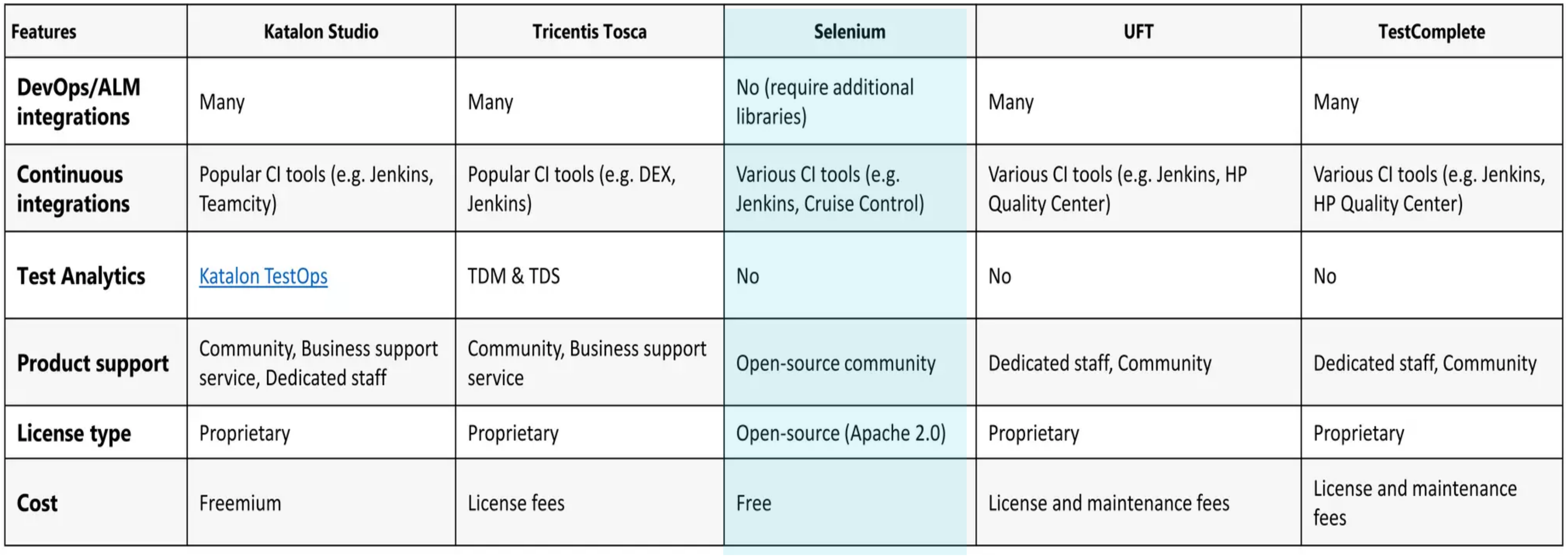
GUI Test Automation Tools



Features	Katalon Studio	Tricentis Tosca	Selenium	UFT	TestComplete
Test development platform	Cross-platform	Windows	Cross-platform	Windows	Windows
Application under test	Windows desktop, Web, Mobile apps, API/Web services	web, desktop, mobile, API based	Web apps	Windows desktop, Web, Mobile apps, API/Web services	Windows desktop, Web, Mobile apps, API/Web services
Scripting languages	Java/Groovy	No coding language required. Module based automation	Java, C#, Perl, Python, JavaScript, Ruby, PHP	VBScript	JavaScript, Python, VBScript, JScript, Delphi, C++ and C#
Programming skills	Not required. Recommended for advanced test scripts	Not required	Advanced skills needed to integrate various tools	Not required. Recommended for advanced test scripts	Not required. Recommended for advanced test scripts
Learning curves	Medium	Medium	High	Medium	Medium
Ease of installation and use	Easy to set up and run	Easy to set up and run	Require installing and integrating various tools	Easy to set up and run	Easy to set up and run
Script creation time	Quick	Quick	Slow	Quick	Quick
Object storage and maintenance	Built-in object repository, XPath, object re-identification	Module based approach.	XPath, UI Maps	Built-in object repository, smart object detection and correction	Built-in object repository, detecting common objects
Image-based testing	Built-in support	Built-in support (OCR approach)	Require installing additional libraries	Built-in support, image-based object recognition	Built-in support

GUI Test Automation Tools

- Comparison of product features



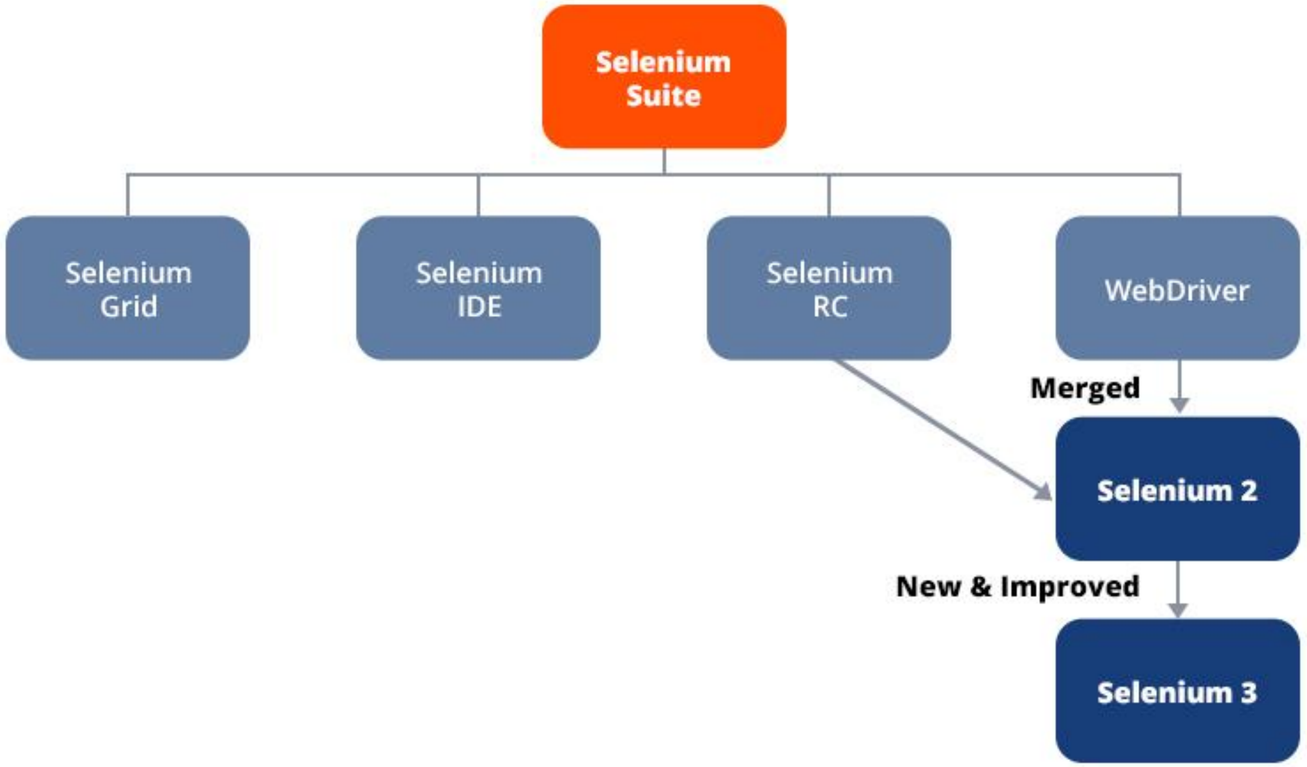
Features	Katalon Studio	Tricentis Tosca	Selenium	UFT	TestComplete
DevOps/ALM integrations	Many	Many	No (require additional libraries)	Many	Many
Continuous integrations	Popular CI tools (e.g. Jenkins, Teamcity)	Popular CI tools (e.g. DEX, Jenkins)	Various CI tools (e.g. Jenkins, Cruise Control)	Various CI tools (e.g. Jenkins, HP Quality Center)	Various CI tools (e.g. Jenkins, HP Quality Center)
Test Analytics	Katalon TestOps	TDM & TDS	No	No	No
Product support	Community, Business support service, Dedicated staff	Community, Business support service	Open-source community	Dedicated staff, Community	Dedicated staff, Community
License type	Proprietary	Proprietary	Open-source (Apache 2.0)	Proprietary	Proprietary
Cost	Freemium	License fees	Free	License and maintenance fees	License and maintenance fees

Selenium Components

- **Selenium** integrated development environment **(IDE)**
- **Selenium remote control (RC)**
- **Webdriver**
- **Selenium Grid**
- **Selenium Action API (Python, Javascript, …)**

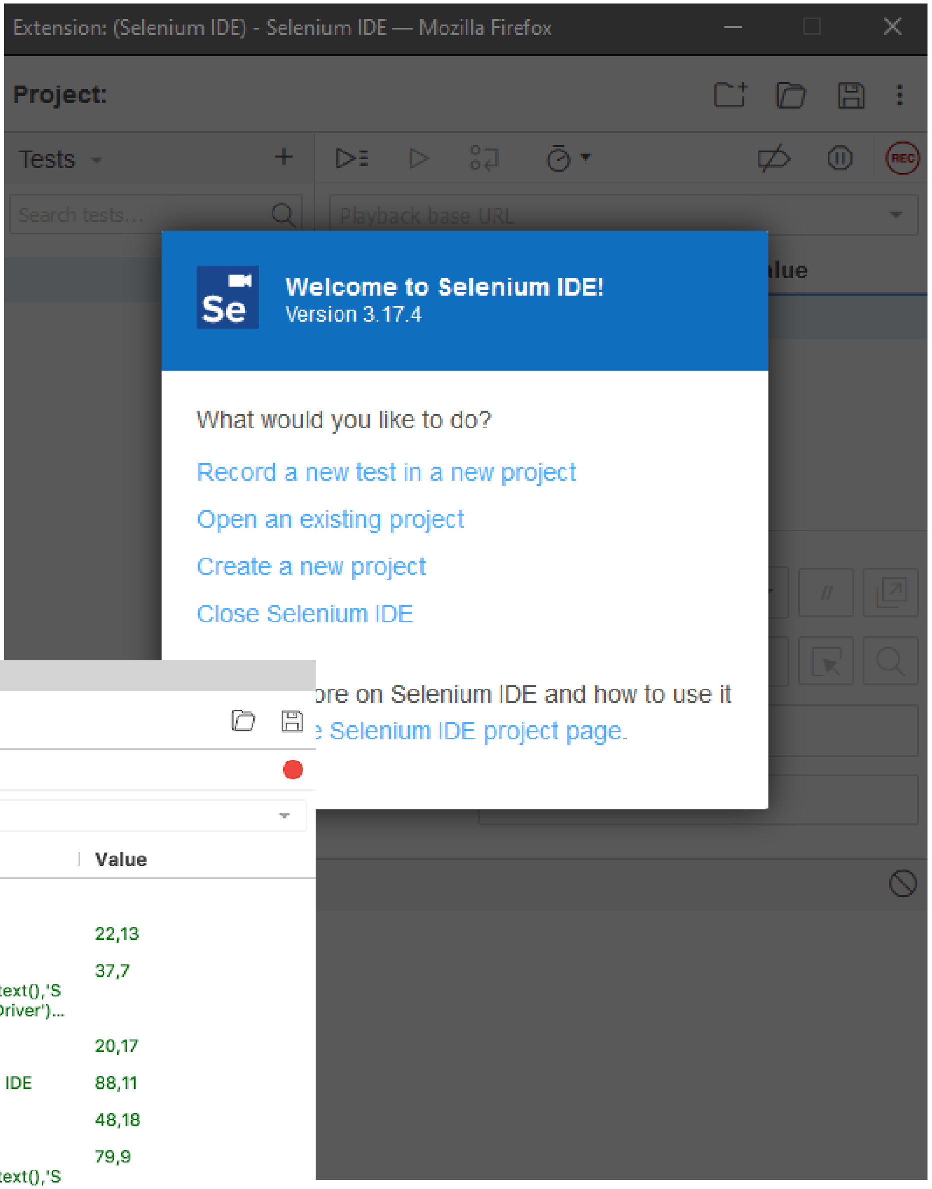
Selenium Components

- **Selenium** integrated development environment **(IDE)**
- **Selenium remote control (RC)**
- **Webdriver**
- **Selenium Grid**
- **Selenium Action API (Python, Javascript, …)**



Selenium IDE

- Firefox and Chrome extension.
- Easy record and replay.
- Debug and set breakpoints.
- Selenium Grid
- https://www.selenium.dev/selenium-ide



Extension: (Selenium IDE) - Selenium IDE — Mozilla Firefox

Project:

Tests

Search tests...

Playback base URL

Welcome to Selenium IDE!
Version 3.17.4

What would you like to do?

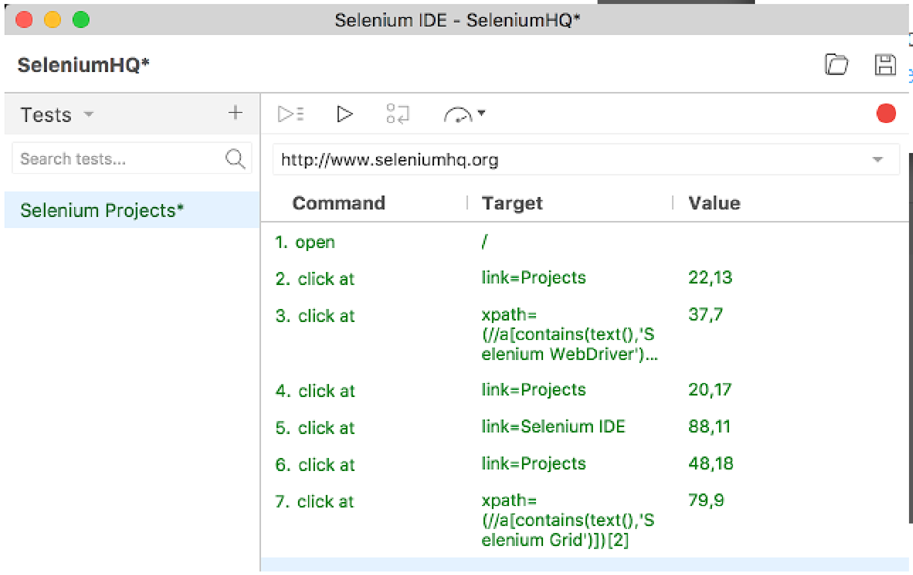
Record a new test in a new project

Open an existing project

Create a new project

Close Selenium IDE

…ore on Selenium IDE and how to use it
…e Selenium IDE project page.



Selenium IDE - SeleniumHQ*

SeleniumHQ*

Tests

Search tests...

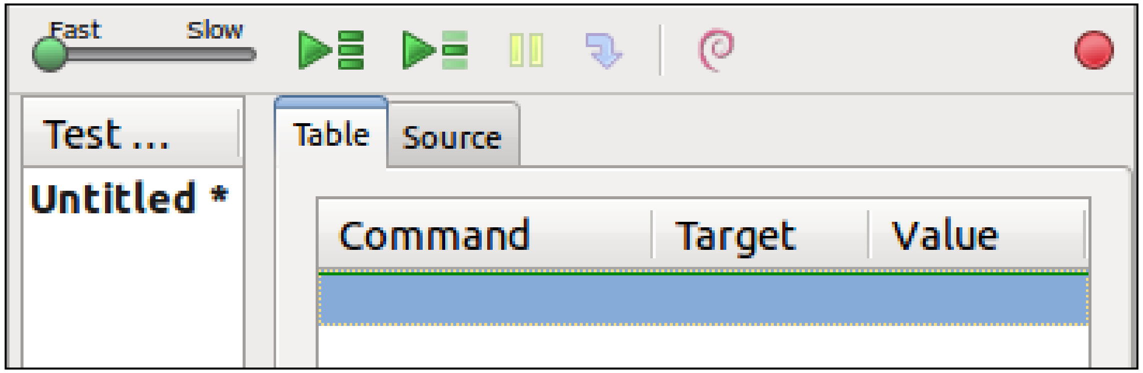
Selenium Projects*

http://www.seleniumhq.org

Command	Target	Value
1. open	/	
2. click at	link=Projects	22,13
3. click at	xpath=(//a[contains(text(),'Selenium WebDriver')...	37,7
4. click at	link=Projects	20,17
5. click at	link=Selenium IDE	88,11
6. click at	link=Projects	48,18
7. click at	xpath=(//a[contains(text(),'Selenium Grid')])[2]	79,9

Selenium IDE test cases

- **Selenium** saves all information in an HTML table format



Command	Target	Value

- Each record consists of:
  - **Command** – tells Selenium what to do (e.g., “open”, “type”, “click”, “verifyText”)
  - **Target** – tells Selenium which HTML element a command refers to (e.g., textbox, header, table)
  - **Value** – used for any command that might need a value of some kind (e.g., type something into a textbox)

How to record/replay with Selenium IDE

1. Start recording in Selenium IDE
2. Execute a test scenario on running web application
3. Stop recording in Selenium IDE
4. Verify / Add assertions
5. Replay the test.

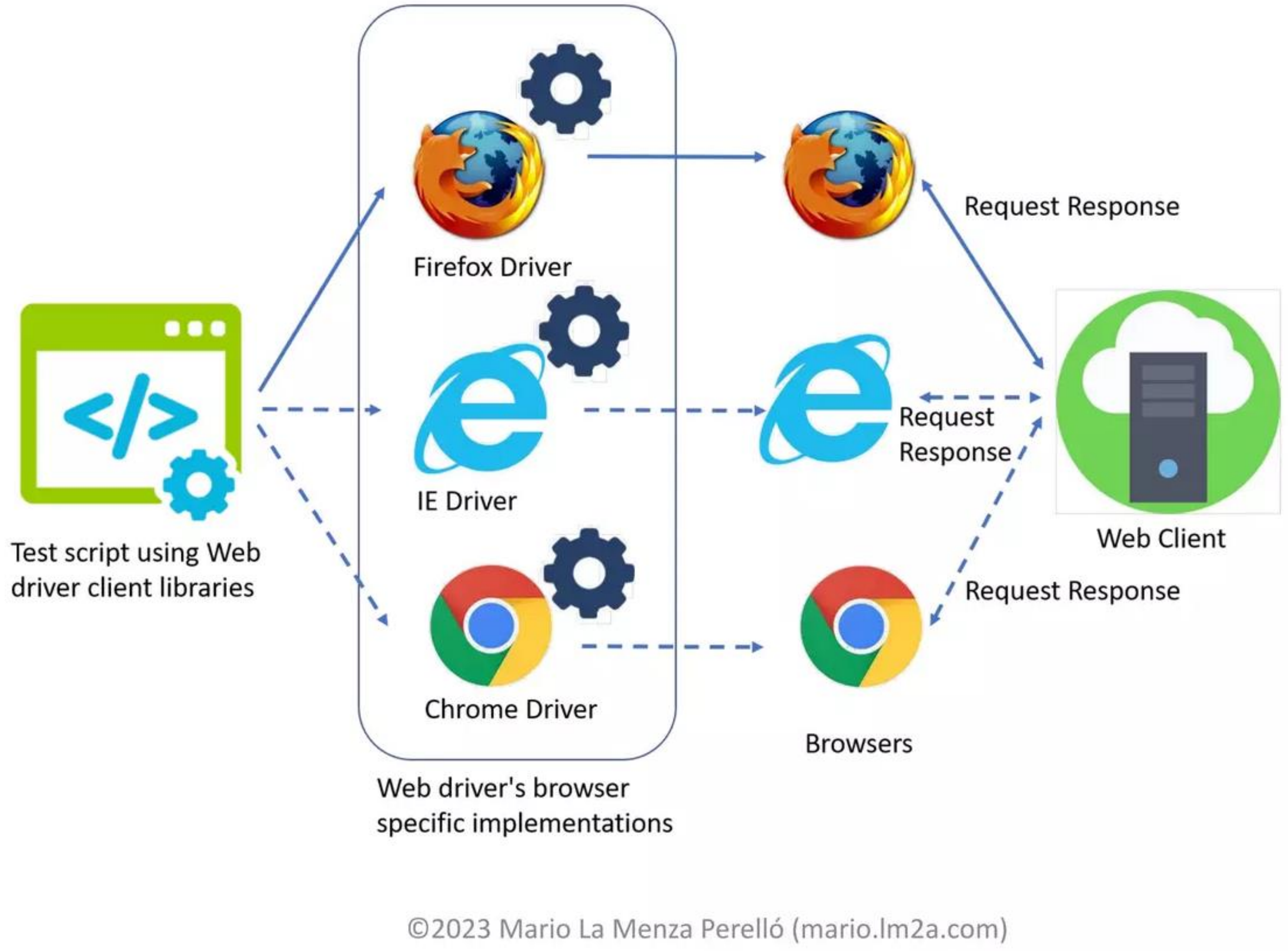
Selenium IDE Demo …

Limitation of Selenium IDE

- No **multiple browsers** support
  - It currently runs in Mozilla Firefox and Google Chrome.
- No manual scripts
  - *E.g.*, conditions and Loops for Data-Driven Testing
- Fancy test cases → Selenium WebDriver

Selenium Webdriver

- Webdriver is a browser automation framework that accepts commands and sends them to browsers.


Firefox Driver
Request Response
IE Driver
Request
Response
Web Client
Test script using Web driver client libraries
Request Response
Chrome Driver
Browsers
Web driver's browser specific implementations
©2023 Mario La Menza Perelló (mario.lm2a.com)

Selenium WebDriver (Selenium 2)

- Selenium-WebDriver
  - A piece of program
  - Control the browser by programming
  - More flexible and powerful

- Selenium-WebDriver supports multiple browsers in multiple platforms
  - Google Chrome 12.0.712.0+
  - Internet Explorer 6+
  - Firefox 3.0+
  - Opera 11.5+
  - Android – 2.3+ for phones and tablets
  - iOS 3+ for phones
  - iOS 3.2+ for tablets

Selenium WebDriver

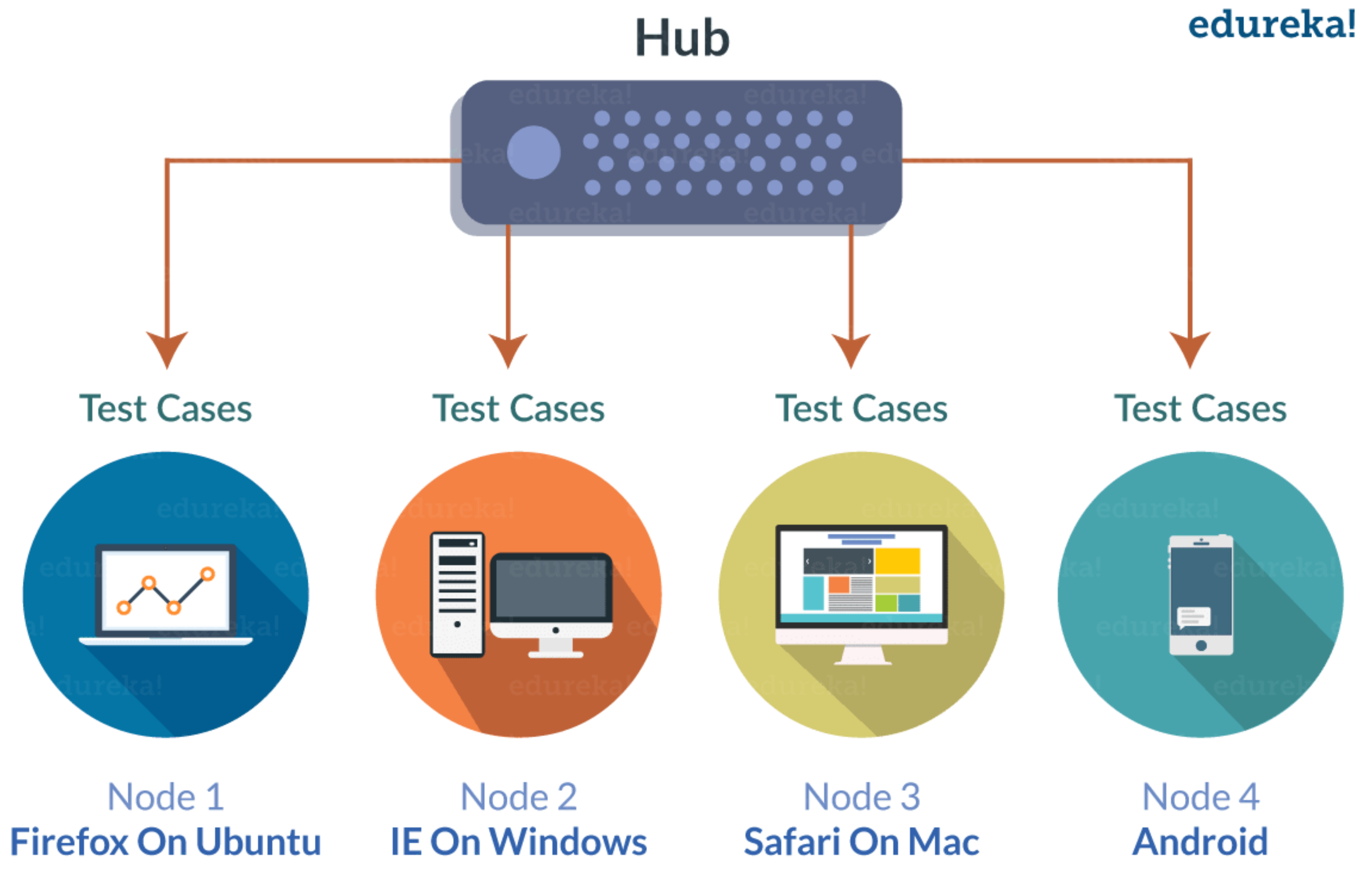
- WebDriver is designed to providing a simpler and uniformed programming interface
  - Control Web Browser by programming
  - Same WebDriver script runs for different platforms
- Support multiple programming language:
  - Java, Javascript, C#, Python, Ruby, PHP, Perl…
- It is efficient
  - WebDriver leverages each browser's native support for automation.

Selenium Webdriver

- A separate test program/ application
- Supports multiple browsers in multiple platforms
  - Chrome,
  - Firefox,
  - Opera,
  - Internet explorer,
  - Andriod,
  - iOS.

Selenium Grid

- Distributed test execution on several machines



Writing UI Functional Tests with Selenium

How to use Selenium WebDriver

(1) Go to a page

(2) Locate an element

(3) Do something with that element

……

(i) Locate an element

(i+1) Do something with that element

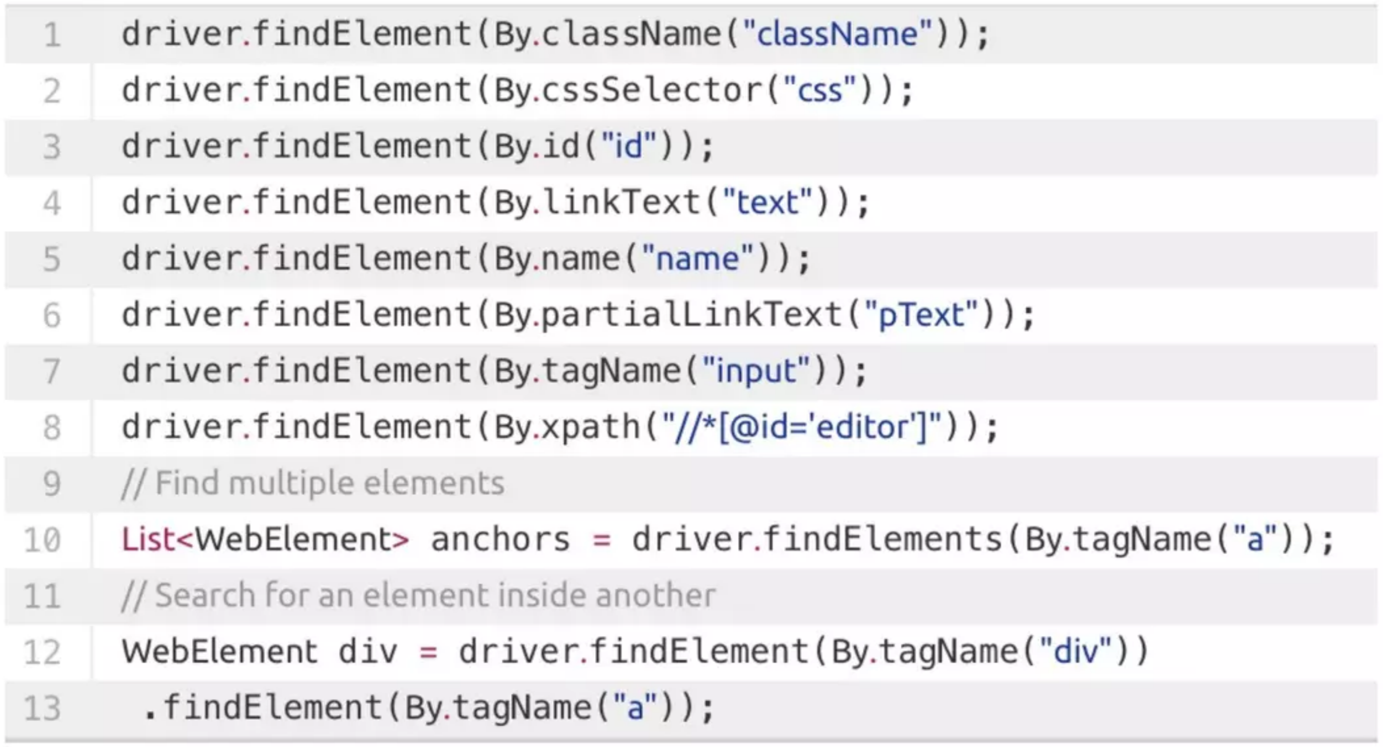
(i+2) Verify / Assert the result

Web Element

- Anything presented in webpage (front-end)
  - Box
  - Labels
  - Button
  - Dropdown

Web element

- They are identified by locators
  - id
  - name
  - class
  - xpath



```java
driver.findElement(By.className("className"));
driver.findElement(By.cssSelector("css"));
driver.findElement(By.id("id"));
driver.findElement(By.linkText("text"));
driver.findElement(By.name("name"));
driver.findElement(By.partialLinkText("pText"));
driver.findElement(By.tagName("input"));
driver.findElement(By.xpath("//*[@id='editor']"));
// Find multiple elements
List<WebElement> anchors = driver.findElements(By.tagName("a"));
// Search for an element inside another
WebElement div = driver.findElement(By.tagName("div"))
 .findElement(By.tagName("a"));
```

Example 1: Verify page title

```
public static void main( String[] args )
{
        // Create a new instance of the Firefox driver
        WebDriver driver = new FirefoxDriver();
        // (1) Go to a page
        driver.get("http://www.google.com");
        // (2) Locate an element
        WebElement element = driver.findElement(By.name("q"));
        // (3-1) Enter something to search for
        element.sendKeys("Purdue Univeristy");
        // (3-2) Now submit the form. WebDriver will find the form for us from 
the element
        element.submit();
        // (3-3) Wait up to 10 seconds for a condition
        WebDriverWait waiting = new WebDriverWait(driver, 10);
        waiting.until( ExpectedConditions.presenceOfElementLocated( 
By.id("pnnext") ) );
        // (4) Check the title of the page
        if( driver.getTitle().equals("purdue univeristy - Google Search") )
            System.out.println("PASS");
        else
            System.err.println("FAIL");

        //Close the browser
        driver.quit();
}
```

How to locate an element

- **By id**
  - **HTML:** `<div id="coolestWidgetEvah">...</div>`
  - **WebDriver:**

```
driver.findElement( By.id("coolestWidgetEvah") );
```

- **By name**
  - **HTML:** `<input name="cheese" type="text"/>`
  - **WebDriver:** `driver.findElement( By.name("cheese") );`
- **By Xpath**
  - **HTML**

```
<html>
  <input type="text" name="example" />
 <input type="text" name="other" />
</html>
```

  - **WebDriver:** `driver.findElements( By.xpath("//input") );`
  - **There are plug-ins for firefox/chrome to automatically display the Xpath**

Time issue

- There are **delays** between submitting a request and receiving the response
- We can wait until the response page is loaded

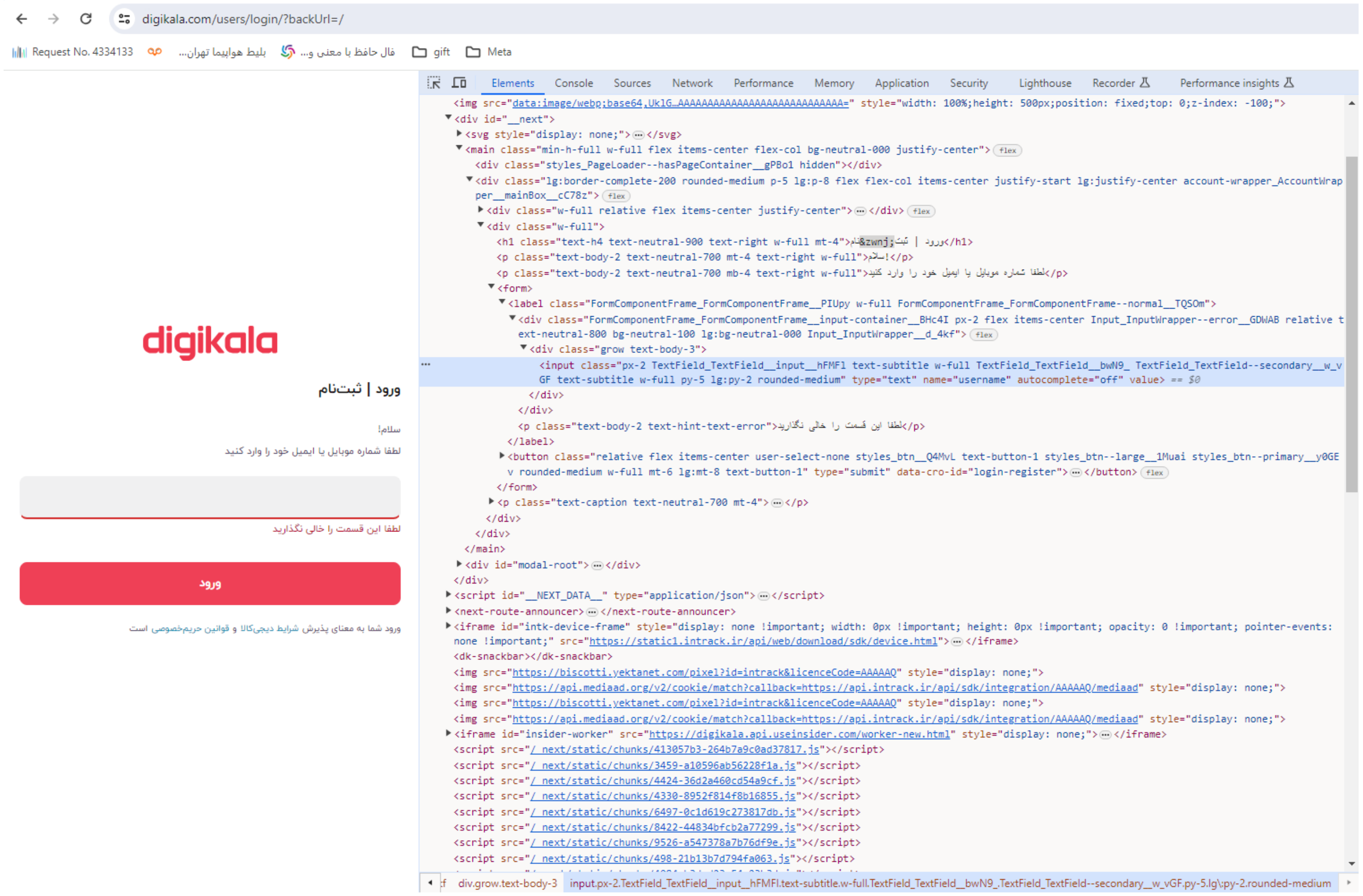
- Robot does not know!
- In WebDriver, sometimes it doesn't work if
  - Submit a request
  - Verify the response immediately
- Solution:
  - Simulate the wait. Wait until some HTML object appears
  - Demo

Example 1: Verify page title

```
public static void main( String[] args )
{
   // Create a new instance of the Firefox driver
   WebDriver driver = new FirefoxDriver();
   // (1) Go to a page
   driver.get("http://www.google.com");
   // (2) Locate an element
   WebElement element = driver.findElement(By.name("q"));
   // (3-1) Enter something to search for
   element.sendKeys("Purdue Univeristy");

  //(3-2) Now submit the form.
   //WebDriver will find the form for us from the element
   element.submit();

   // (3-3) Wait up to 10 seconds for a condition
   WebDriverWait waiting = new WebDriverWait(driver, 10);
   waiting.until( ExpectedConditions.presenceOfElementLocated(By.id("pnnext")));
        // (4) Check the title of the page
        if( driver.getTitle().equals("purdue univeristy - Google Search") )
            System.out.println("PASS");
        else
            System.err.println("FAIL");

        //Close the browser
        driver.quit();
}
```

Example 2: Test Digikala login feature

- Digikala login page elements



ورود | ثبت‌نام

سلام!

لطفا شماره موبایل یا ایمیل خود را وارد کنید

لطفا این قسمت را خالی نگذارید

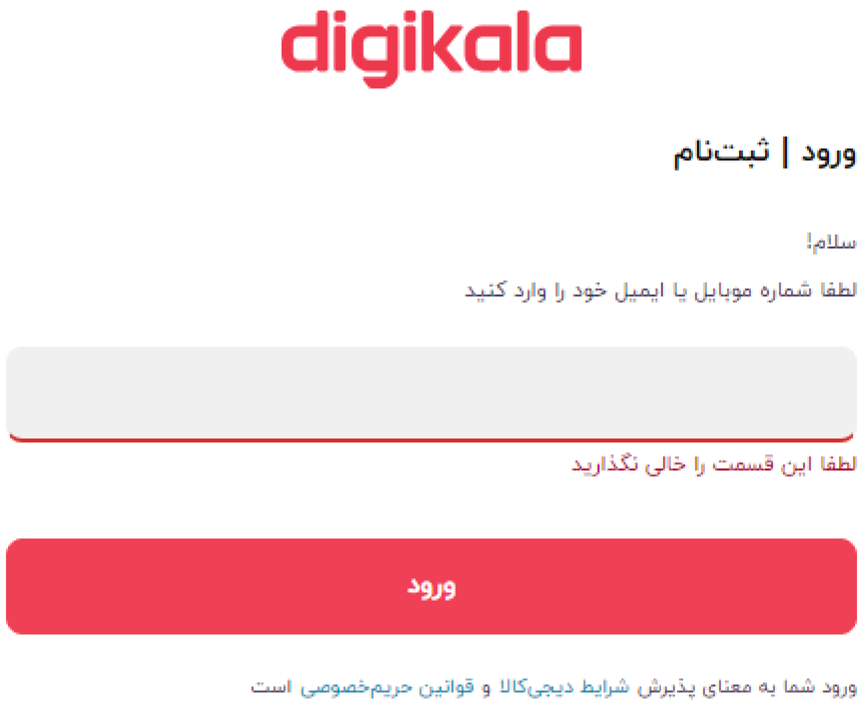
ورود

ورود شما به معنای پذیرش شرایط دیجی‌کالا و قوانین حریم‌خصوصی است

```
<img src="data:image/webp;base64,UklG…AAAAAAAAAAAAAAAAAAAAAAAAAAA=" style="width: 100%;height: 500px;position: fixed;top: 0;z-index: -100;">
▼<div id="__next">
  ▶<svg style="display: none;">⋯</svg>
  ▼<main class="min-h-full w-full flex items-center flex-col bg-neutral-000 justify-center"> flex
     <div class="styles_PageLoader--hasPageContainer__gPBo1 hidden"></div>
    ▼<div class="lg:border-complete-200 rounded-medium p-5 lg:p-8 flex flex-col items-center justify-start lg:justify-center account-wrapper_AccountWrap
     per__mainBox__cC78z"> flex
      ▶<div class="w-full relative flex items-center justify-center">⋯</div> flex
      ▼<div class="w-full">
         <h1 class="text-h4 text-neutral-900 text-right w-full mt-4">ورود | ثبت&zwnj;نام</h1>
         <p class="text-body-2 text-neutral-700 mt-4 text-right w-full">سلام!</p>
         <p class="text-body-2 text-neutral-700 mb-4 text-right w-full">لطفا شماره موبایل یا ایمیل خود را وارد کنید</p>
        ▼<form>
          ▼<label class="FormComponentFrame_FormComponentFrame__PIUpy w-full FormComponentFrame_FormComponentFrame--normal__TQSOm">
            ▼<div class="FormComponentFrame_FormComponentFrame__input-container__BHc4I px-2 flex items-center Input_InputWrapper--error__GDWAB relative t
             ext-neutral-800 bg-neutral-100 lg:bg-neutral-000 Input_InputWrapper__d_4kf"> flex
              ▼<div class="grow text-body-3">
...              <input class="px-2 TextField_TextField__input__hFMFl text-subtitle w-full TextField_TextField__bwN9_ TextField_TextField--secondary__w_v
                 GF text-subtitle w-full py-5 lg:py-2 rounded-medium" type="text" name="username" autocomplete="off" value> == $0
               </div>
             </div>
             <p class="text-body-2 text-hint-text-error">لطفا این قسمت را خالی نگذارید</p>
           </label>
          ▶<button class="relative flex items-center user-select-none styles_btn__Q4MvL text-button-1 styles_btn--large__1Muai styles_btn--primary__y0GE
           v rounded-medium w-full mt-6 lg:mt-8 text-button-1" type="submit" data-cro-id="login-register">⋯</button> flex
         </form>
        ▶<p class="text-caption text-neutral-700 mt-4">⋯</p>
       </div>
     </div>
   </main>
  ▶<div id="modal-root">⋯</div>
 </div>
▶<script id="__NEXT_DATA__" type="application/json">⋯</script>
▶<next-route-announcer>⋯</next-route-announcer>
▶<iframe id="intk-device-frame" style="display: none !important; width: 0px !important; height: 0px !important; opacity: 0 !important; pointer-events:
 none !important;" src="https://static1.intrack.ir/api/web/download/sdk/device.html">⋯</iframe>
 <dk-snackbar></dk-snackbar>
 <img src="https://biscotti.yektanet.com/pixel?id=intrack&licenceCode=AAAAAQ" style="display: none;">
 <img src="https://api.mediaad.org/v2/cookie/match?callback=https://api.intrack.ir/api/sdk/integration/AAAAAQ/mediaad" style="display: none;">
 <img src="https://biscotti.yektanet.com/pixel?id=intrack&licenceCode=AAAAAQ" style="display: none;">
 <img src="https://api.mediaad.org/v2/cookie/match?callback=https://api.intrack.ir/api/sdk/integration/AAAAAQ/mediaad" style="display: none;">
▶<iframe id="insider-worker" src="https://digikala.api.useinsider.com/worker-new.html" style="display: none;">⋯</iframe>
 <script src="/_next/static/chunks/413057b3-264b7a9c0ad37817.js"></script>
 <script src="/_next/static/chunks/3459-a10596ab56228f1a.js"></script>
 <script src="/_next/static/chunks/4424-36d2a460cd54a9cf.js"></script>
 <script src="/_next/static/chunks/4330-8952f814f8b16855.js"></script>
 <script src="/_next/static/chunks/6497-0c1d619c273817db.js"></script>
 <script src="/_next/static/chunks/8422-44834bfcb2a77299.js"></script>
 <script src="/_next/static/chunks/9526-a547378a7b76df9e.js"></script>
 <script src="/_next/static/chunks/498-21b13b7d794fa063.js"></script>
```

```
f  div.grow.text-body-3  input.px-2.TextField_TextField__input__hFMFl.text-subtitle.w-full.TextField_TextField__bwN9_.TextField_TextField--secondary__w_vGF.py-5.lg\:py-2.rounded-medium
```

Example 2: Test Digikala login feature



ورود | ثبت‌نام

سلام!

لطفا شماره موبایل یا ایمیل خود را وارد کنید

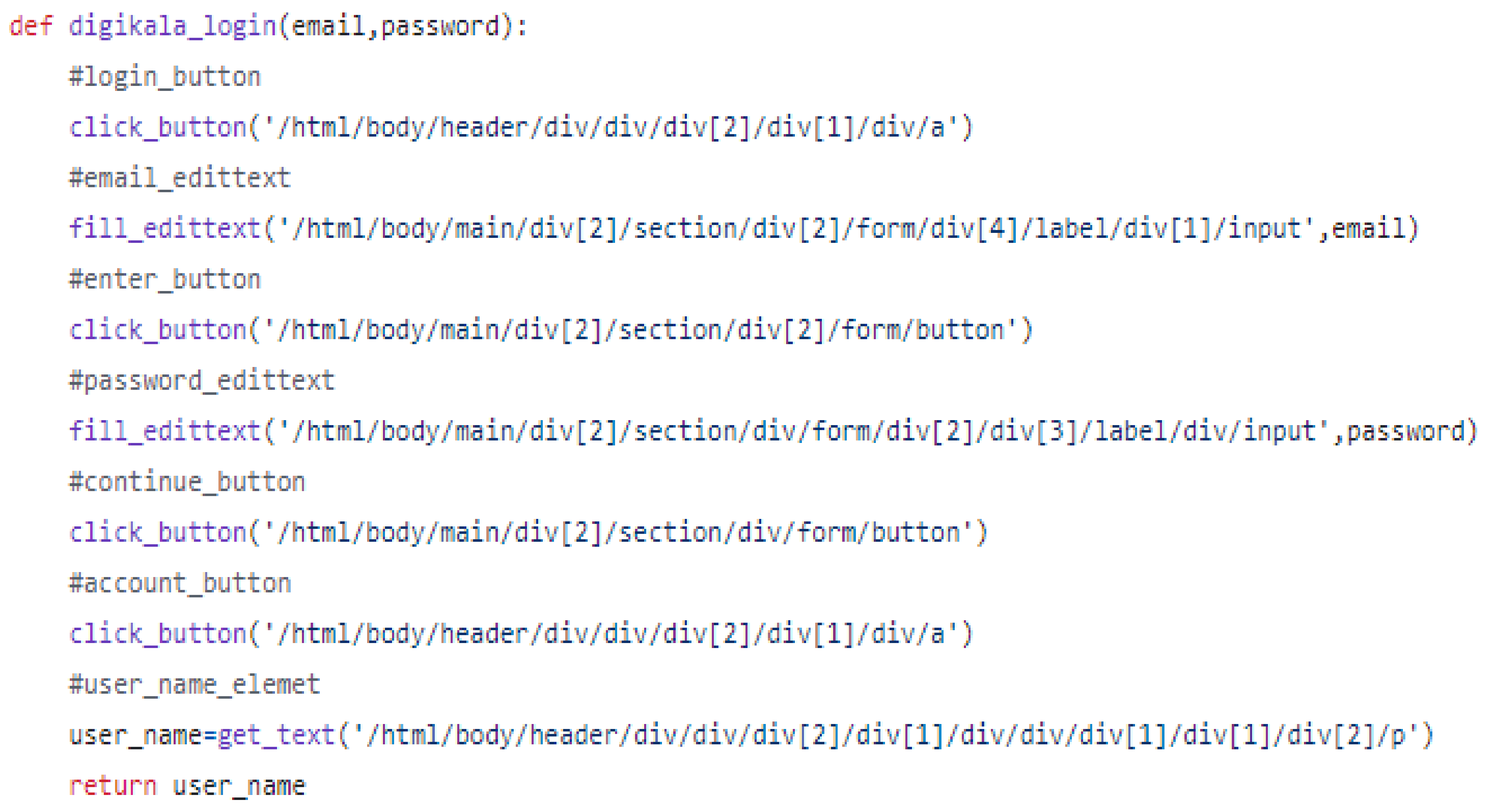
لطفا این قسمت را خالی نگذارید

ورود

ورود شما به معنای پذیرش شرایط دیجی‌کالا و قوانین حریم‌خصوصی است

```
import getpass
from selenium import webdriver
from selenium.webdriver.common.by import By

chrome_browser=webdriver.Chrome()
#chrome_browser.maximize_window()
chrome_browser.get('https://www.digikala.com')
# User's credentials
email = input("Enter your email or phone number: ")
password = getpass.getpass("Enter your password: ")
user_name = getpass.getpass("Enter your name: ")

usr_nm=digikala_login(email,password)
try:
    assert (user_name==usr_nm)
    print('passed!')
except Exception as e:
    print(e)
    print('failed!')
finally:
    chrome_browser.close()
```

Example 2: Test Digikala login feature



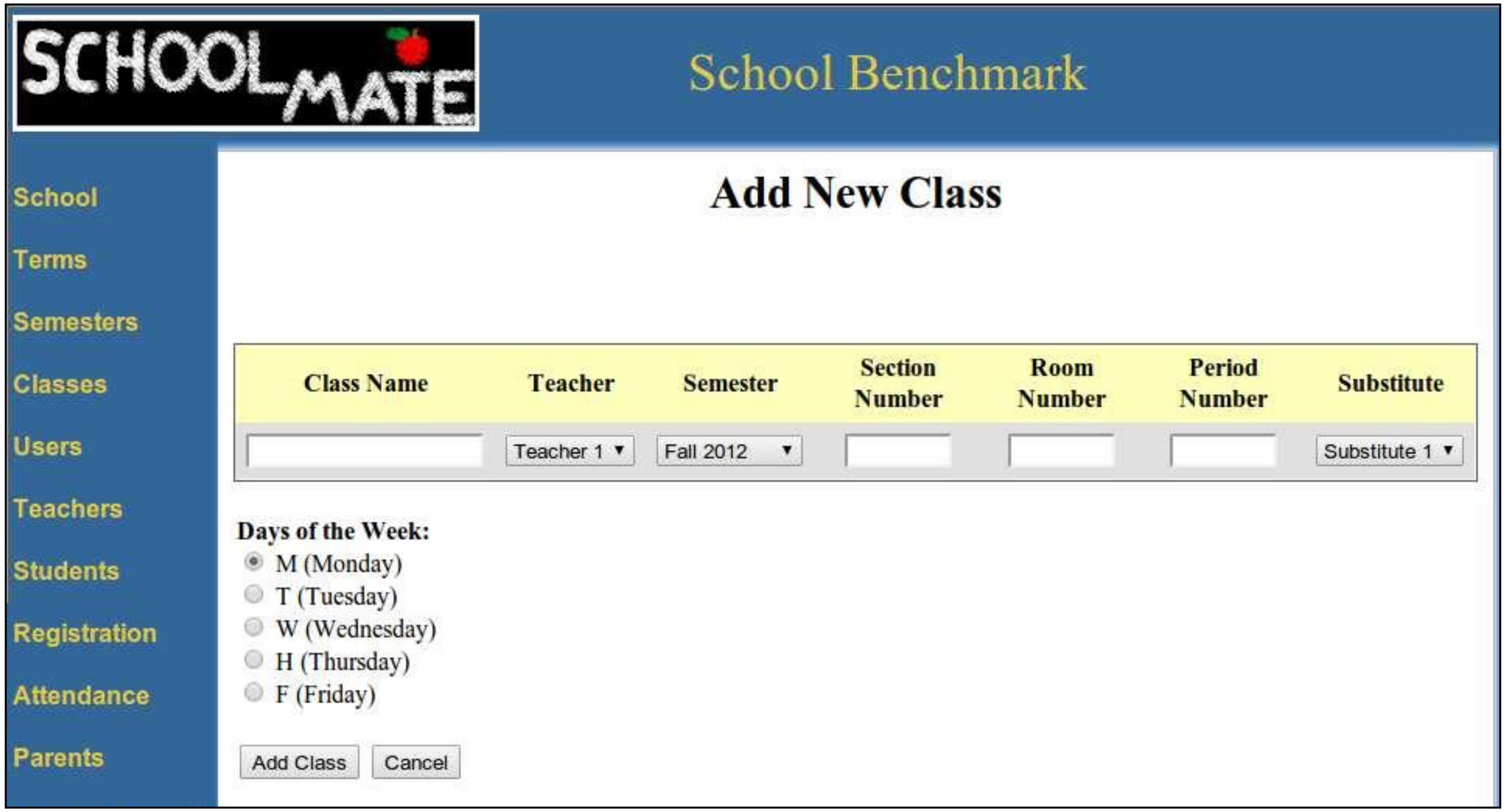
```python
def digikala_login(email,password):
    #login_button
    click_button('/html/body/header/div/div/div[2]/div[1]/div/a')
    #email_edittext
    fill_edittext('/html/body/main/div[2]/section/div[2]/form/div[4]/label/div[1]/input',email)
    #enter_button
    click_button('/html/body/main/div[2]/section/div[2]/form/button')
    #password_edittext
    fill_edittext('/html/body/main/div[2]/section/div/form/div[2]/div[3]/label/div/input',password)
    #continue_button
    click_button('/html/body/main/div[2]/section/div/form/button')
    #account_button
    click_button('/html/body/header/div/div/div[2]/div[1]/div/a')
    #user_name_elemet
    user_name=get_text('/html/body/header/div/div/div[2]/div[1]/div/div/div[1]/div[1]/div[2]/p')
    return user_name
```

Complete source code: *https://github.com/FtmhBkhsh/Test*

GUI Testing Assignment

GUI Testing Assignment

- Test a functionality without the source
- The subject web application
  - "Add New Class" in "SchoolMate"



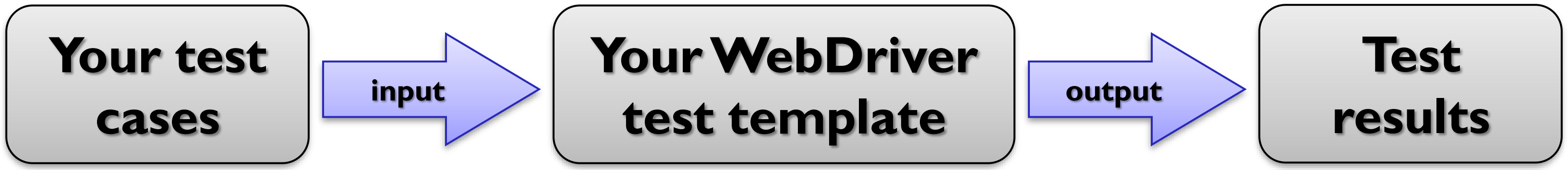
Add New Class

Class Name	Teacher	Semester	Section Number	Room Number	Period Number	Substitute
	Teacher 1	Fall 2012				Substitute 1

Days of the Week:
- [✓ M (Monday)]
- T (Tuesday)
- W (Wednesday)
- H (Thursday)
- F (Friday)

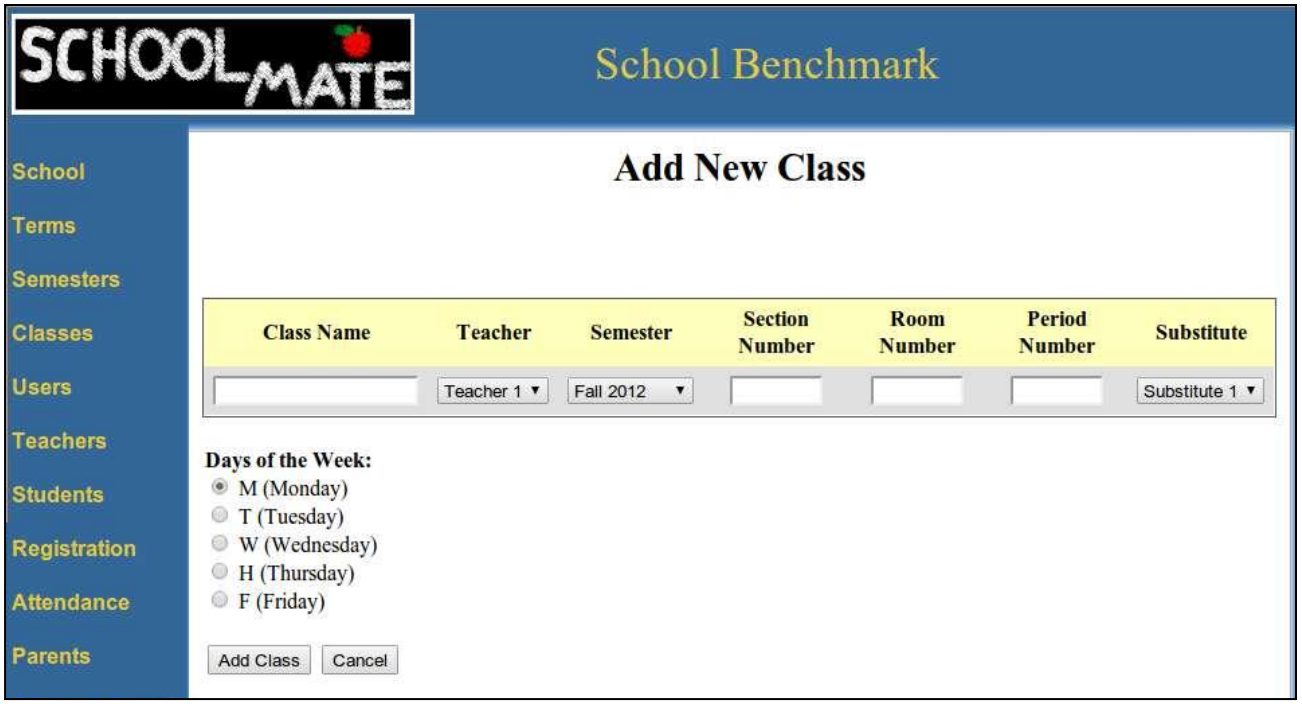
GUI Testing Assignment: Part 1

- Part 1: Overview
  - Design test cases against the requirement
    - The tests should consider
      - all possible cases
      - equivalence class partitioning
  - Implement Selenium WebDriver Script for “Add new class”



Part 1: requirement analysis

- Requirement for values entered/selected
  - **[R-1] Class Name** : alphabets and numbers are allowed.
  - **[R-2] Section Number** : only numbers are allowed.
  - **[R-3] Room Number** : only numbers are allowed.
  - **[R-4] Period Number** : only numbers are allowed.
  - **[R-5] All textbox fields** : no Cross-Site Scripting (XSS) injection vulnerabilities.



Class Name	Teacher	Semester	Section Number	Room Number	Period Number	Substitute
	Teacher 1	Fall 2012				Substitute 1

Days of the Week:
- [✓ M (Monday)]
- T (Tuesday)
- W (Wednesday)
- H (Thursday)
- F (Friday)

Part 1: requirement analysis

- Requirement for the “add” function
- After clicking the “Add class” button…
  - **[R-6]** The class record added is successfully shown in the table.
  - **[R-7]** The values are exactly the same as those were entered or selected.

Part 1: Design testcases

- For each requirement, design test cases
- Only need test one field in each case
  - Do not need consider combinations of fileds
- *E.g.*
  - **[R-1] Class Name** : alphabets and numbers are allowed.
  - You need consider all possible cases
  - Divide the test space and find the equivalence class
    - Alphabets
    - Numbers
    - ….
- The test case format is defined
  - Test cases will be used as input to your WebDriver Script

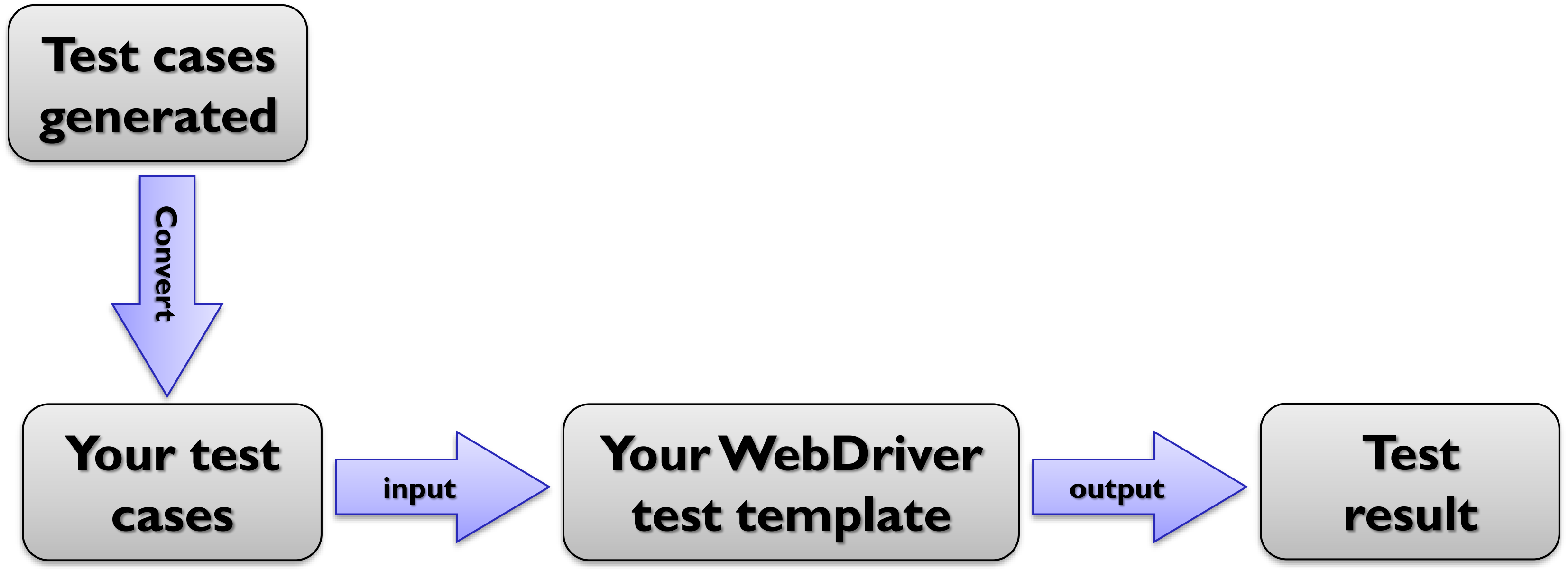
Part 1: Implement WebDriver Script

- In WebDriver script, Simulate the user action
  - Navigate to the subject page
  - Enter values into textboxs based on input values
  - Select options based on input values
  - Click the “add class” button
  - Check the result against the requirements

- Run all your test cases and report the results.

GUI Testing Assignment: Part 2

- Part 2: Pair wise testing
  - Use an existing pair wise testing tool fire-eye
  - Largely you may reuse the WebDriver template in Part 1



Part 2: Generate test cases

- Consider the combination of all textboxs/options
- Use the existing tool, fire-eye, to generate test cases
- Export the test case in “Nist form”

- Parse and convert the exported test cases to the form that your WebDriver can accept.
- Run all pair-wise test cases.
- Report the results.

GUI Testing Assignment: Part 3

- Solution Design
- Selenium can do more …

- Black Friday is coming, hot items can be sold in a few seconds

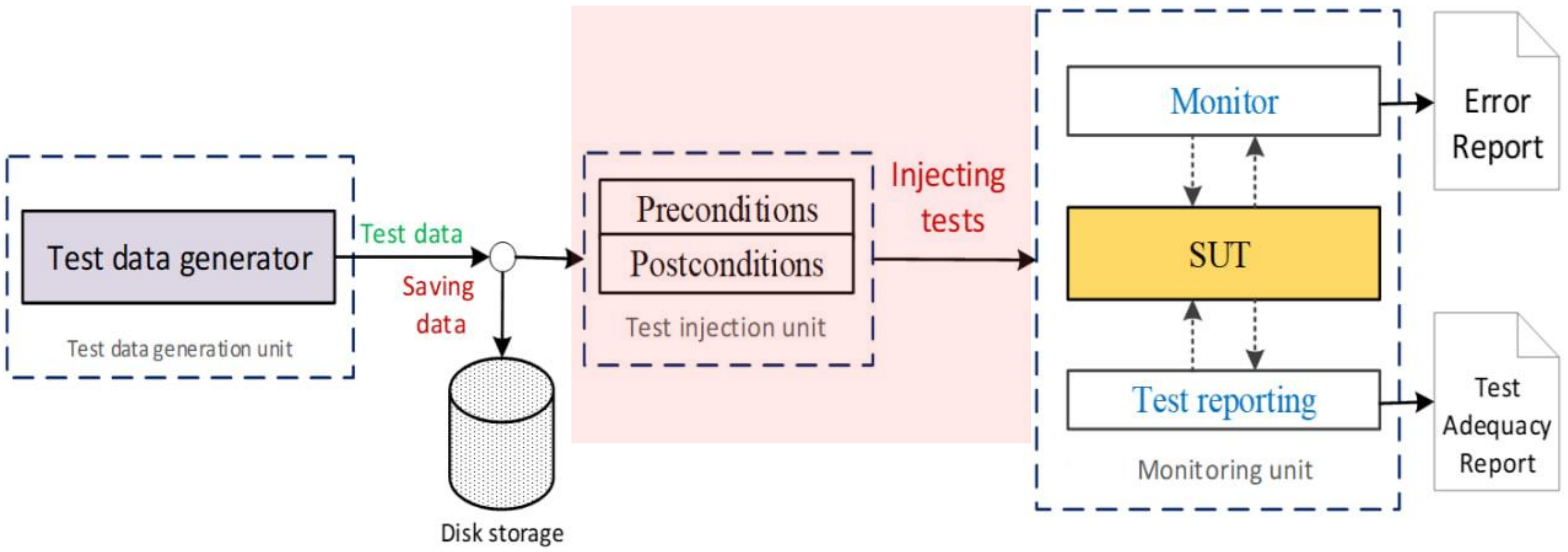
- Can you leverage the automated tool and design a practical solution to score a super hot deal?
- Explain
  - What is the challenge
  - What is the possible cases to handle and how?
    - In stock
    - Out of stock
    - Your shopping cart may be reset under what conditions…
  - How to add it into your shopping cart asap
  - How you are going to cooperate with the automated tool

GUI Testing Appendix:

Automatic test data/case generation for GUI Apps

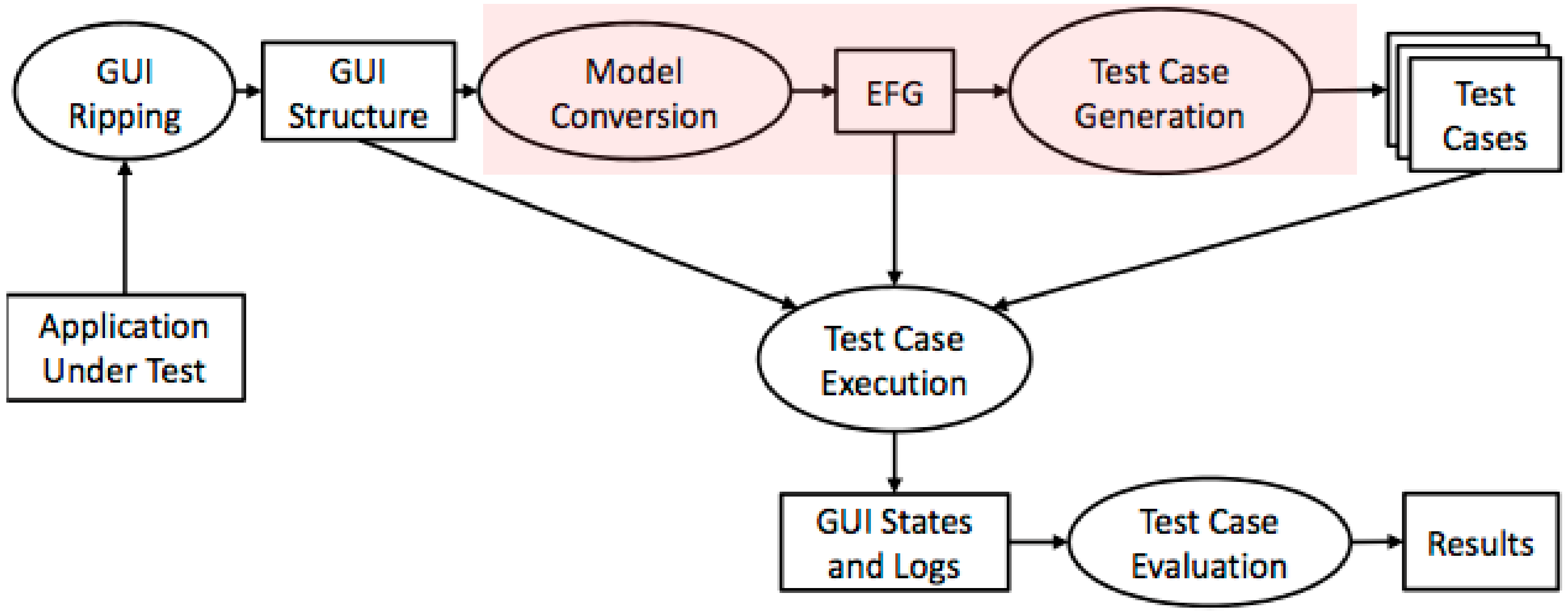
Test automation

- → **Test data generation**
- Test injection
- Monitoring and report



Automated GUI Test data generation

- **GUITAR**
  - A GUI testing framework (https://sourceforge.net/projects/guitar/)
  - Nguyen, B.N., Robbins, B., Banerjee, I. *et al.* **GUITAR: an innovative tool for automated testing of GUI-driven software**. *Autom Softw Eng* 21, 65–105 (2014). https://doi.org/10.1007/s10515-013-0128-9

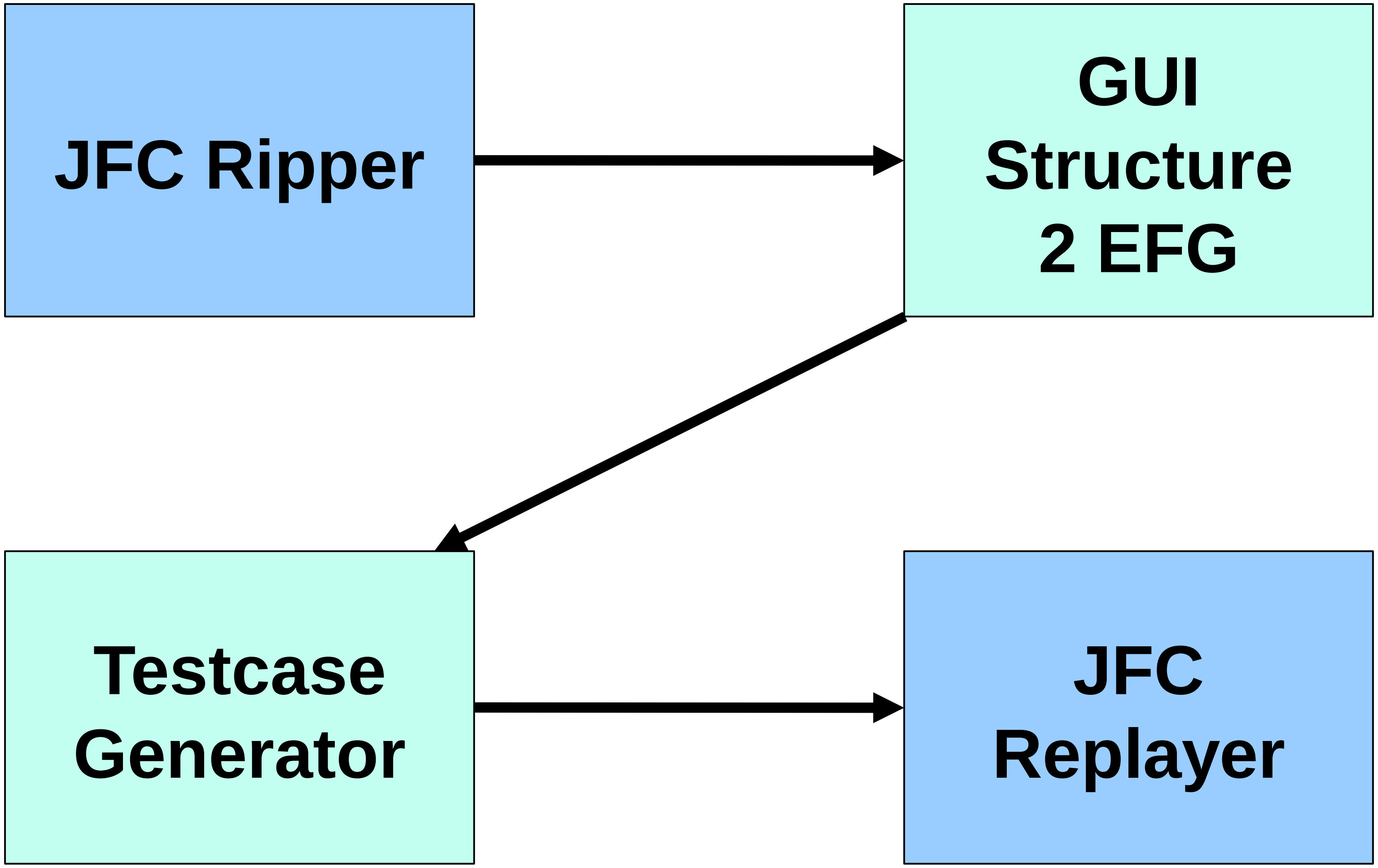


A Simple Testing Workflow

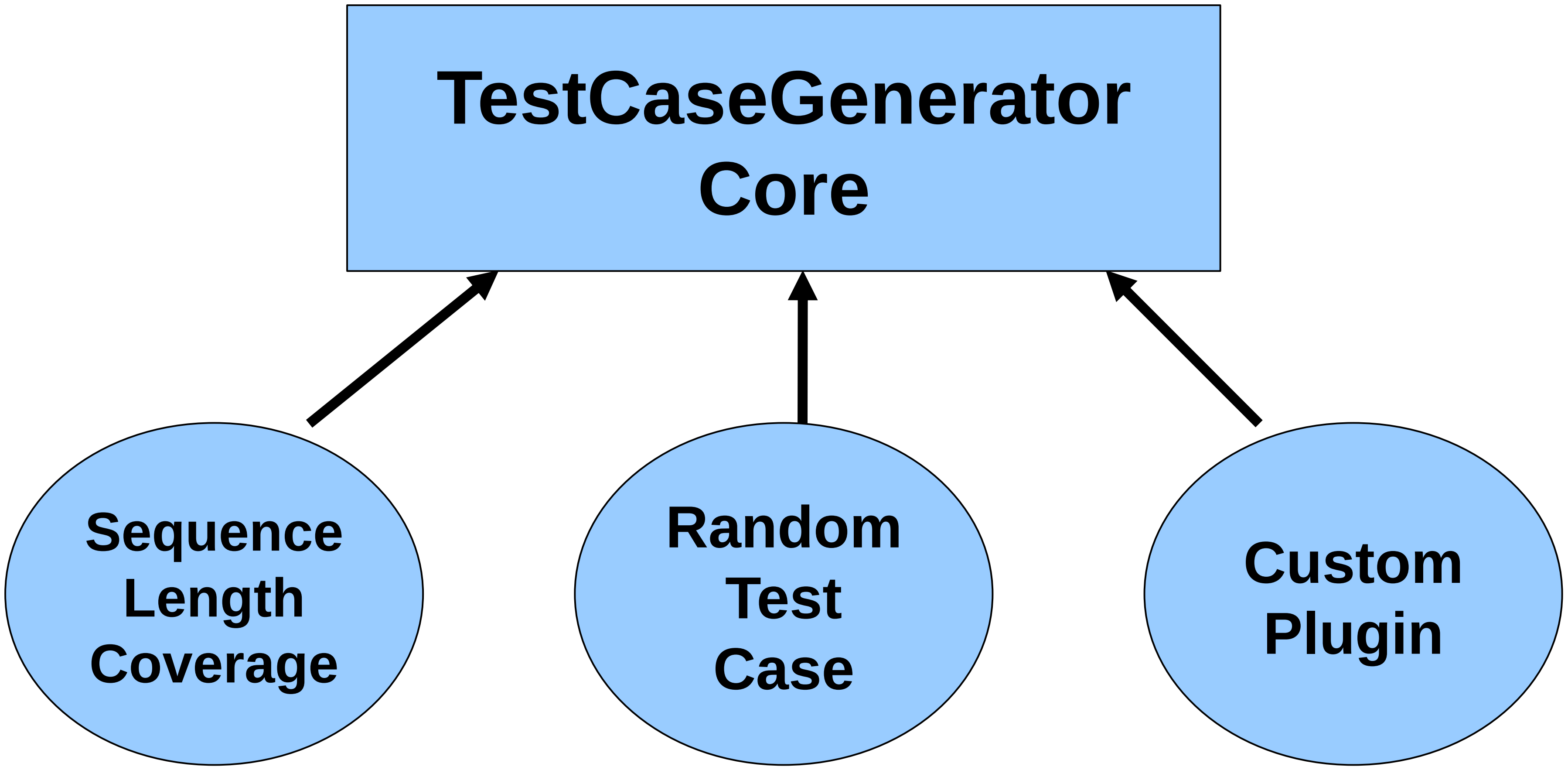
Automated GUI Test data generation

- GUITAR contains four parts:
  - The GUIRipper
    - Used to extract GUI information from a program.
  - The GUIStructure2Graph
    - Uses this output to build a traversable graph representation of the GUI elements.
  - The TestCaseGenerator
    - Creates an extensive set of test cases based on the graph.
  - The GUIReplayer
    - Runs the program as instructed by these tests.

GUITAR Recap



TCG Plugin Structure



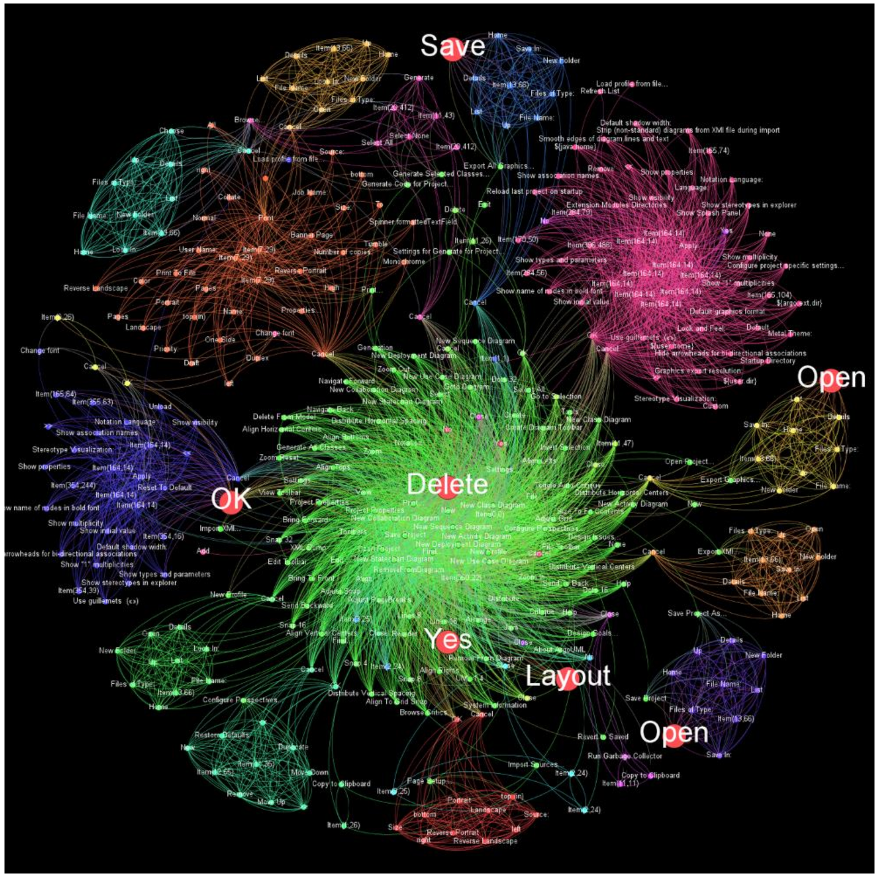
GUITAR Test Case Generator (TCG)

- Generates test cases given two inputs:
  - **Formal model** of the GUI in the form of a **Graph**
  - Graph traversal algorithm to simulate a user's possible interactions with the GUI
- Event Flow Graph (EFG)
- Bundled plugins
  - Sequence Length
  - Random

TCG Demo

- Test case generator arguments:
  - Output.dir
  - EFG
  - Plugin
  - Length
  - Max-number
- These arguments can be changed in the `TestCaseGenerator.properties` file

Sequence Length Plugin Demo



Looking at a sequence generated:

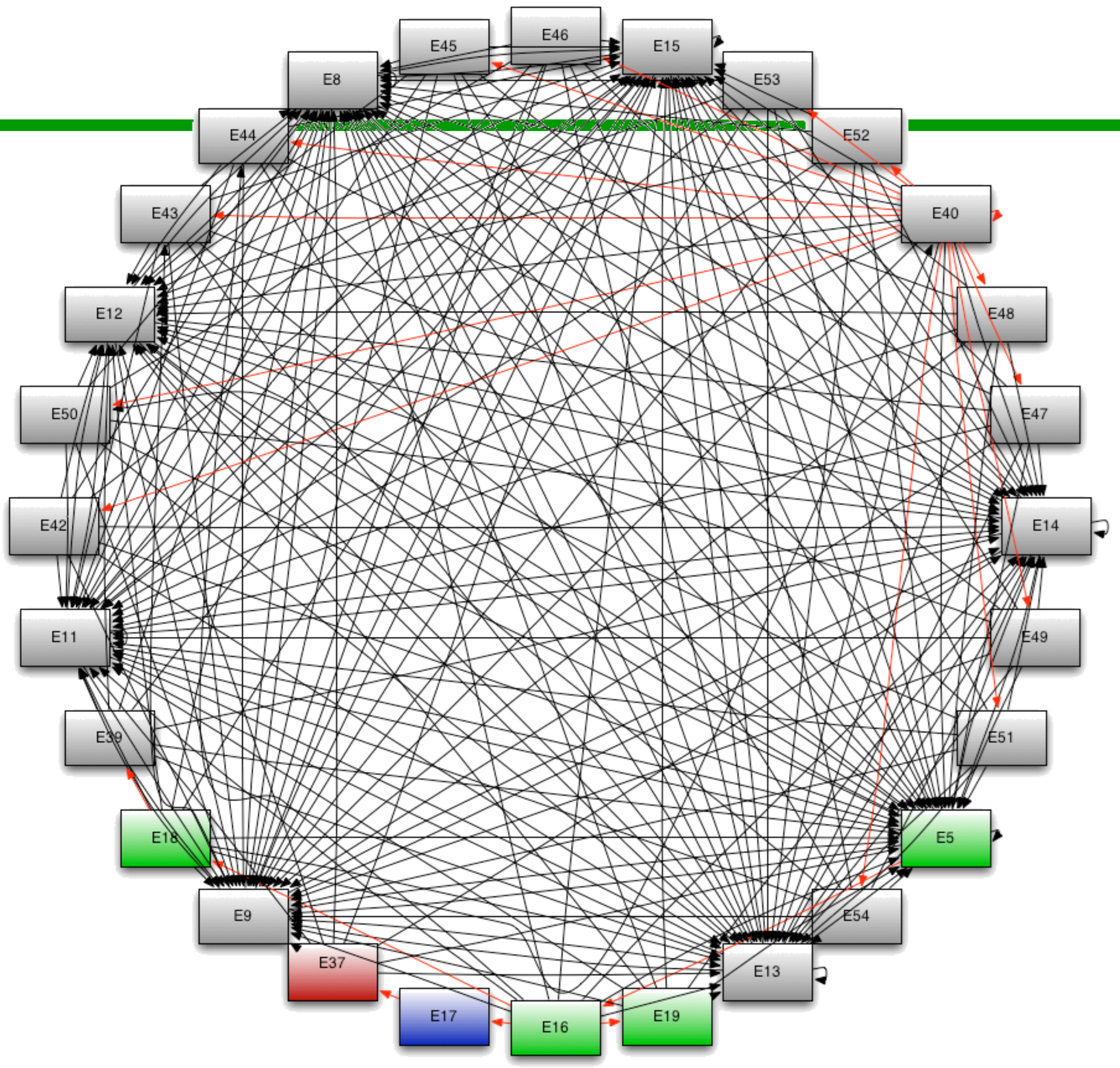
```
<?xml version="1.0" encoding="UTF-8"
      standalone="yes"?>
<TestCase>
   <Step>
      <EventId>e16</EventId>
      <ReachingStep>false</ReachingStep>
   </Step>
   <Step>
      <EventId>e18</EventId>
      <ReachingStep>false</ReachingStep>
   </Step>
   <Step>
      <EventId>e39</EventId>
      <ReachingStep>false</ReachingStep>
   </Step>
</TestCase>
```

WeightedRandom Plugin Demo

- Takes as input an EFG representing the graph structure and an EFG representing the weights of edges.

- Generates shortest path test cases for each potential starting event to every reachable event.

- Rest of tests generated by picking starting event and rest of path randomly, weighted by out edges.



🟩 Expand Event

🟦 Restricted Focus Event

🟥 Terminal Event

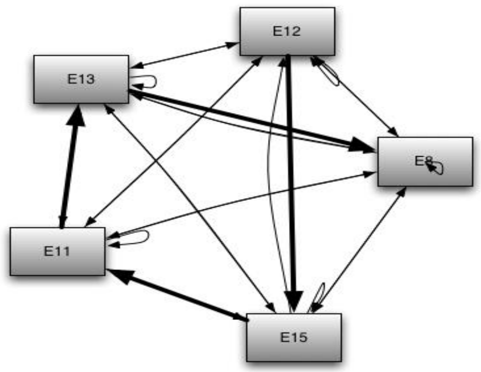
⬜ System Interaction Event

Edge, Weight: 1 →

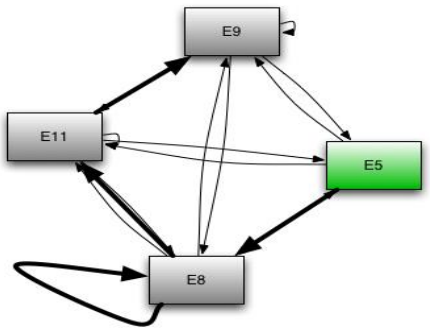
Edge, Weight: 2 →

Example Test Cases

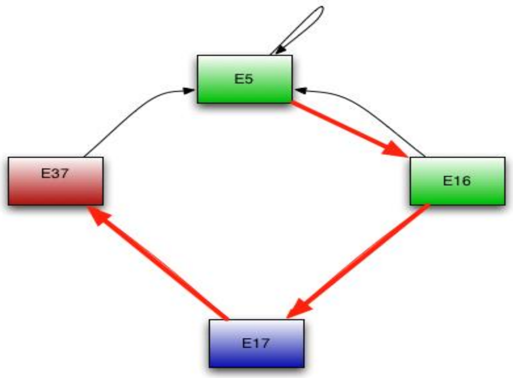
t_random0.tst



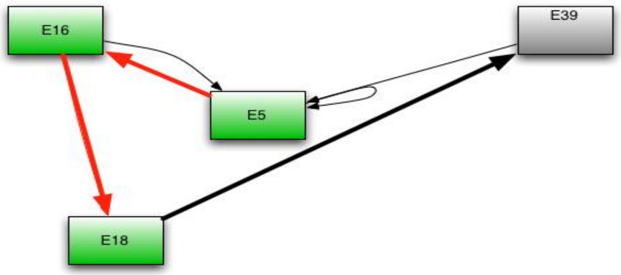
t_random9.tst

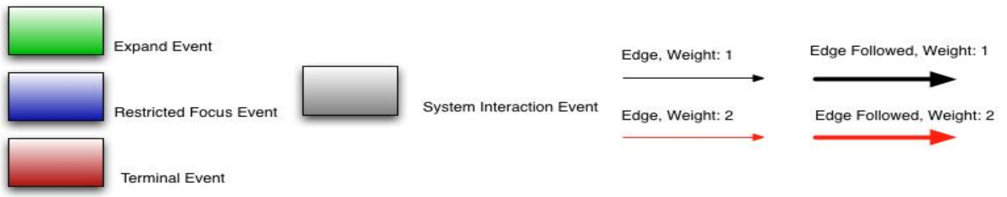


t_shortestPath_e37.tst

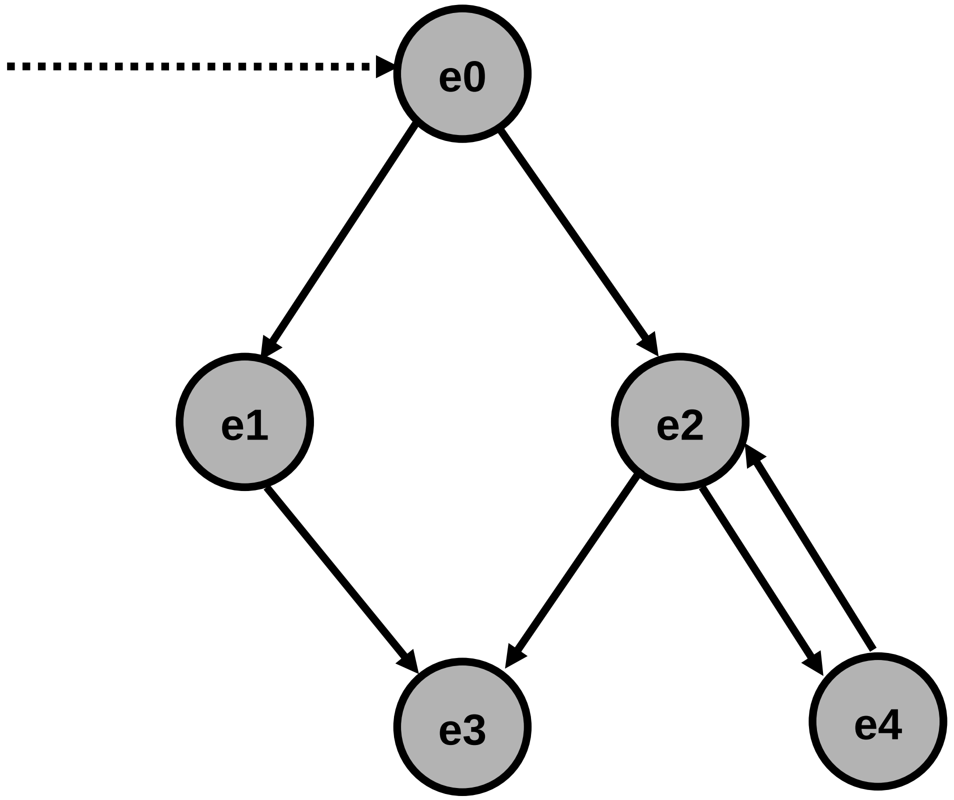


t_shortestPath_e39.tst

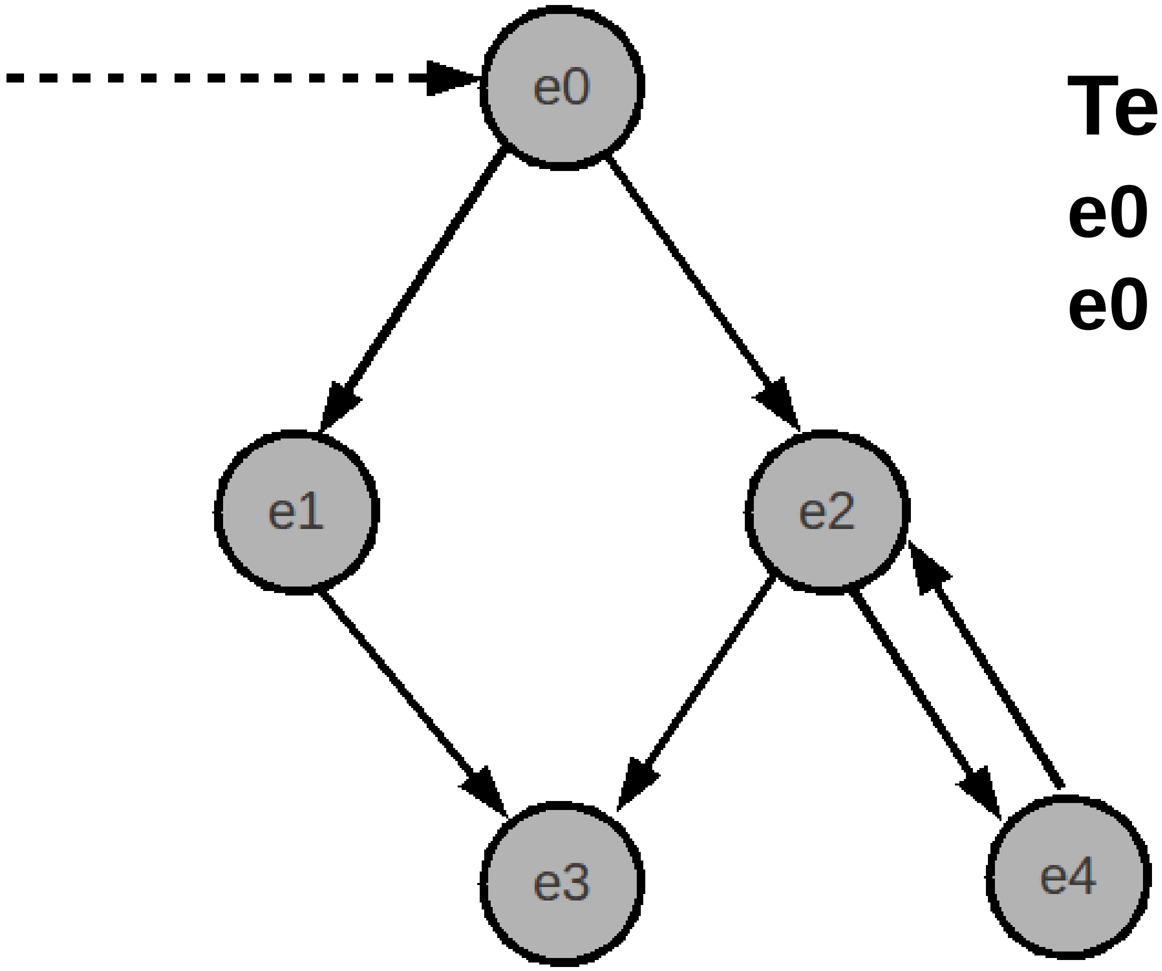




An EFG



A Graph Traversal (SequenceLengthCoverage)

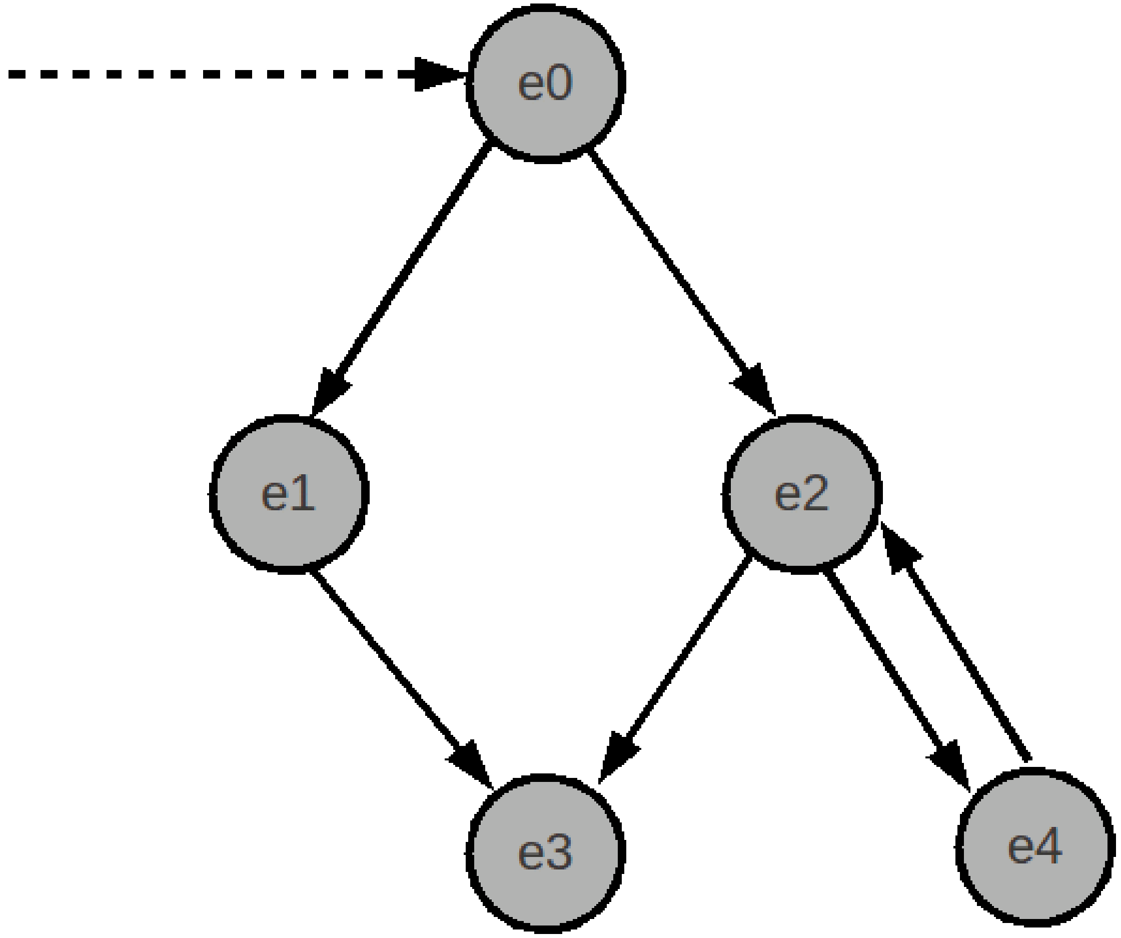


Testcases:

e0 → e2 → e4

e0 → e2 → e3

Another Graph Traversal (Random)



Testcases:

e0 → e1 → e3

e0 → e2 → e4

Plugin Goals

TestCaseGenerator uses this plugin mechanism due to different goals in graph traversal:

Speed: Complete coverage plugins may be infeasible for larger graphs

Completeness: For smaller GUIs, all possible test cases may be preferred

Focus: Specific types of test cases, e.g. "All test cases with no cycles"

Plugin Implementation

- GUITAR TestCaseGenerator plugins are based off the TCPlugin abstract class

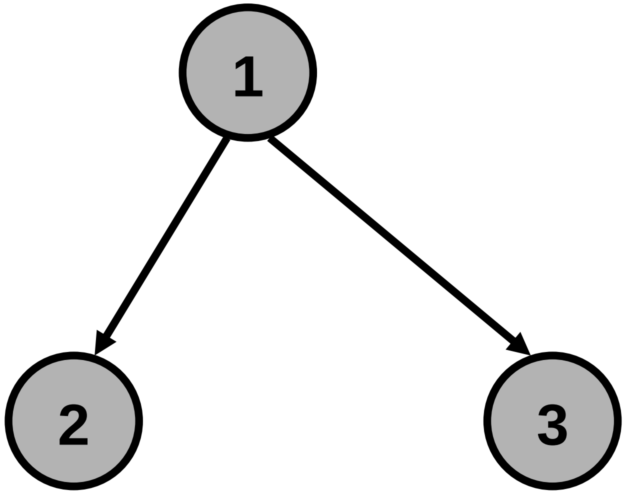
public abstract class TCPlugin

*** generate(EFG efg, String outputDir, int nMaxNumber)**
*** getConfiguration()**
*** isValidArgs()**

*** writeToFile(String tCName, LinkedList<EventType> path)**
*** initialize()**
*** getPathToRoot(EventType event)**

Plugin Implementation (ctd)

- TCG events are broken down into three types:
  - Successors
  - Predecessors
  - Initial Events



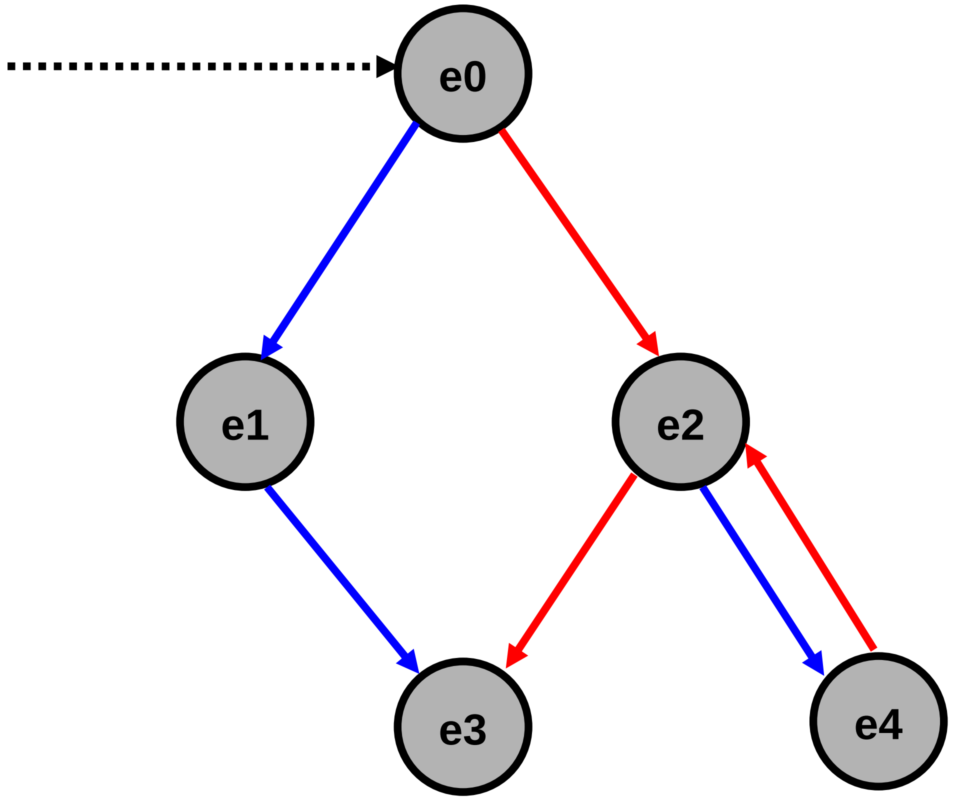
Relevant GUITAR Constants

NO_EDGE: There is no relationship between the two events

FOLLOW_EDGE: Typical edges in a graph

REACHING_EDGE: Edges used to reach an event

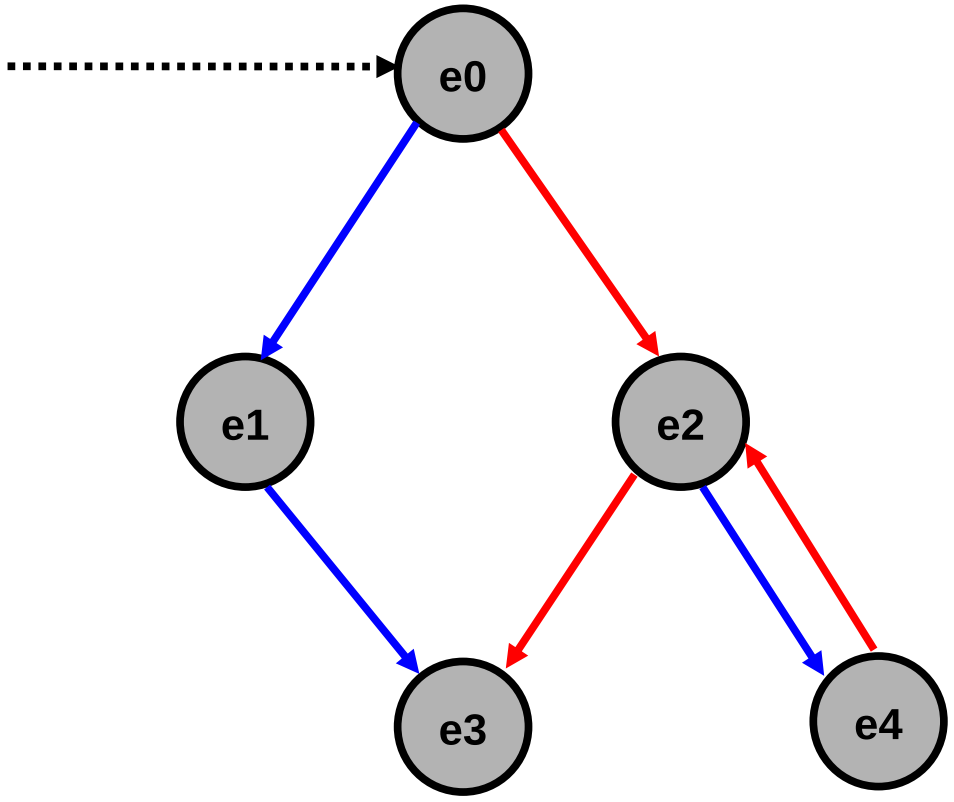
initialize()



— FOLLOW_EDGE

— REACHING_EDGE

initialize()

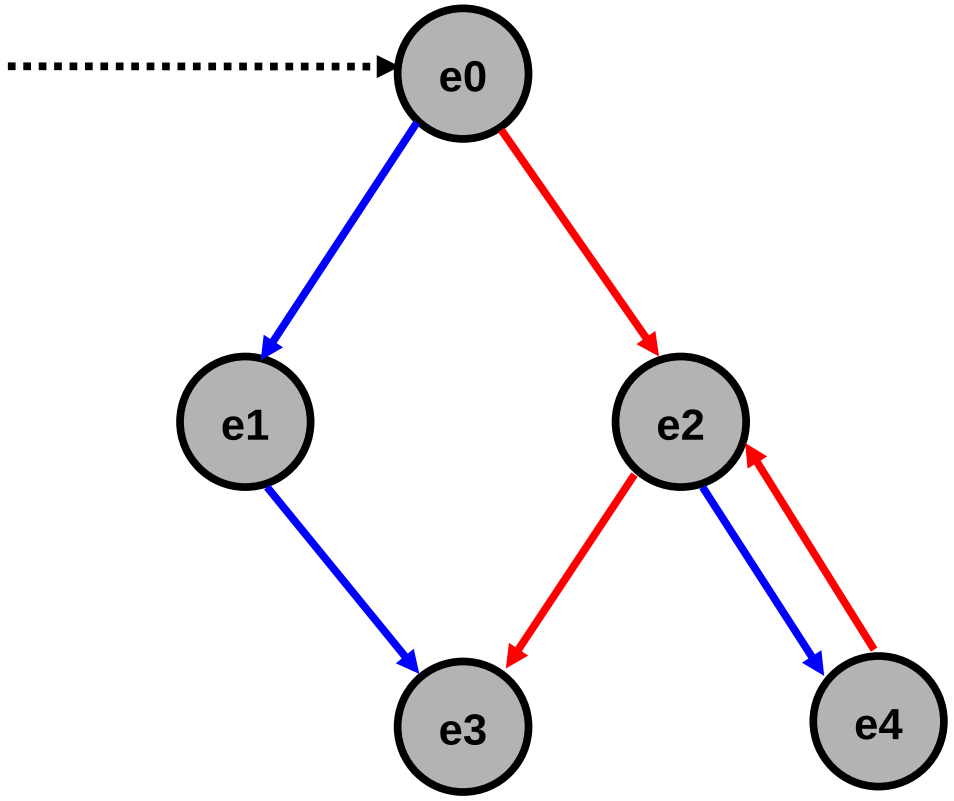


initialEvents: {e0}

preds: {e0 → Ø}
{e1 → Ø}
{e2 → e0, e4}
{e3 → e2}
{e4 → Ø}

succs: {e0 → e1, e2}
{e1 → e3}
{e2 → e3, e4}
{e3 → Ø}
{e4 → e2}

getPathToRoot(EventType event)



Paths to Root -

e0 : e0

e1 : Ø

e2 : e0 → e2

e3 : e0 → e2 → e3

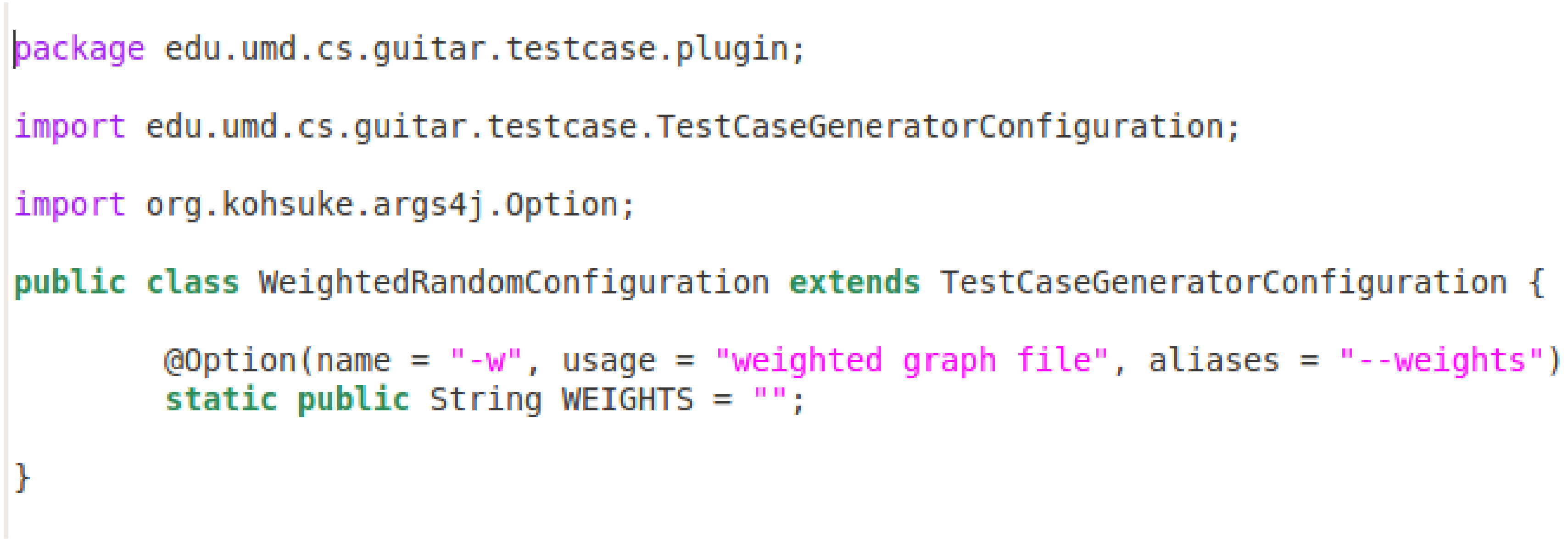
e4 : Ø

Configuration

- In many cases, additional arguments are needed for a TestCaseGenerator plugin.
- Plugins can override a “getConfiguration” method in the TCPlugin interface.
- The configuration specifies which additional arguments are to be parsed from the command line.

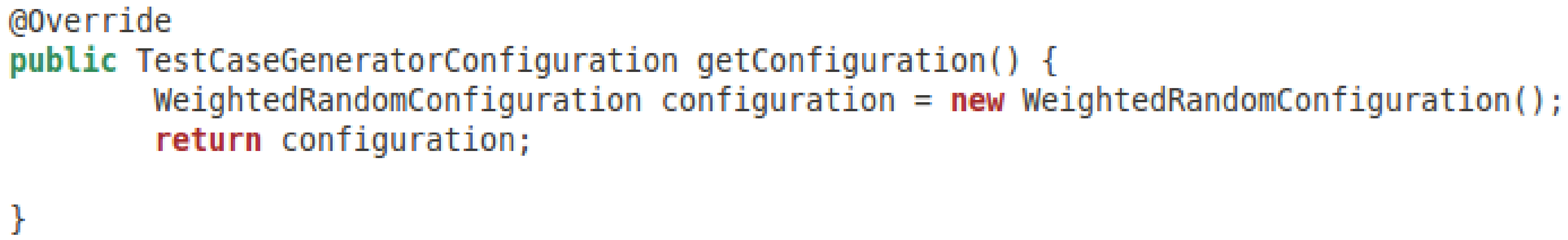
Example: Weighted Random Configuration

Configuration Class File:



```java
package edu.umd.cs.guitar.testcase.plugin;

import edu.umd.cs.guitar.testcase.TestCaseGeneratorConfiguration;

import org.kohsuke.args4j.Option;

public class WeightedRandomConfiguration extends TestCaseGeneratorConfiguration {

        @Option(name = "-w", usage = "weighted graph file", aliases = "--weights")
        static public String WEIGHTS = "";

}
```

Implementation of getConfiguration() in WeightedRandom.java:



```java
@Override
public TestCaseGeneratorConfiguration getConfiguration() {
        WeightedRandomConfiguration configuration = new WeightedRandomConfiguration();
        return configuration;

}
```

Relevant Java Imports

`org.kohsuke.args4j`

- Useful command line parser that contains
- An “Option” object, which acts as a **field/setter** that receives a command line **switch value.**