GEMINI SOLUTIONS

Testare manuala, Testare automata

Curs

Tipuri de testare

GEMINI SOLUTIONS

- Unit
- Functional
- Smoke
- Acceptance
- Usability
- Security
- Accessibility
- Recovery
- Scaling
- Performance
- Load
- Regression
- Requirement
- And others…

Tipuri de testare

GEMINI SOLUTIONS

- Unit
- Functional
- Smoke
- Acceptance
- Usability
- Security
- Accessibility
- Recovery
- Scaling
- Performance
- Load
- Regression
- Requirement
- And others…

Tipuri de testare

GEMINI SOLUTIONS

- Unit
- Functional
- Smoke
- Acceptance
- Usability
- Security
- Accessibility
- Recovery
- Scaling
- Performance
- Load
- Regression
- Requirement
- And others…

Unit testing

GEMINI SOLUTIONS

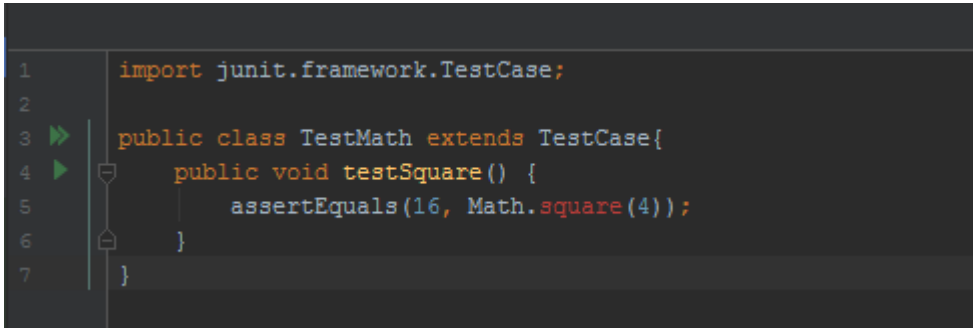
- Smallest testable part – method/function level assertions

Exemplu: Se cere o metoda care sa faca x^2, plecand de la test unitar.

Unit testing

GEMINI SOLUTIONS

Creating the unit test!



```
1  import junit.framework.TestCase;
2
3  public class TestMath extends TestCase{
4      public void testSquare() {
5          assertEquals(16, Math.square(4));
6      }
7  }
```

Unit testing

GEMINI SOLUTIONS

Method

```
1  public class Math {
2      public static int square(int x) {
3          return x;
4      }
5  }
6
```

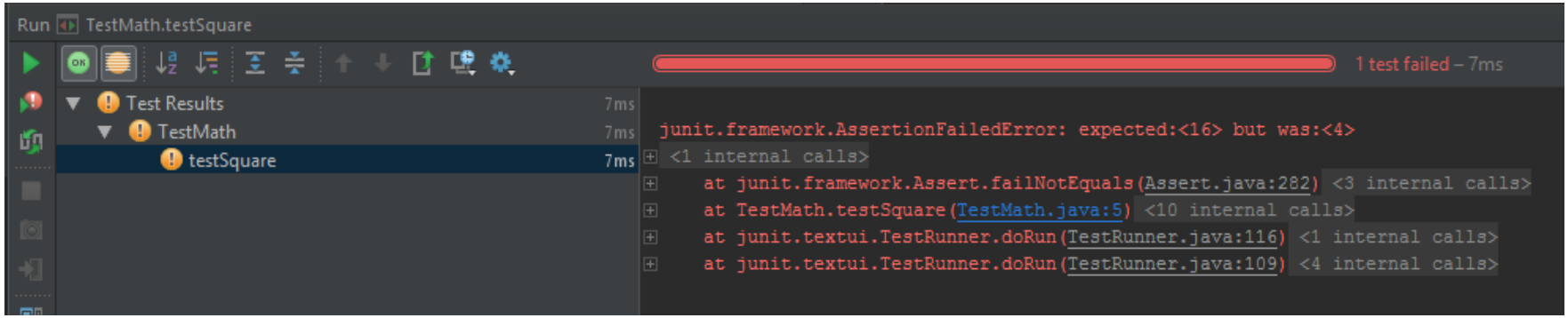
Unit test

```
1  import junit.framework.TestCase;
2
3  public class TestMath extends TestCase{
4      public void testSquare() {
5          assertEquals( expected: 16, Math.square( x: 4));
6      }
7  }
```

Unit testing

GEMINI SOLUTIONS

Test failed!



Unit testing

GEMINI SOLUTIONS

Fixing the problem!

Method

```
1  public class Math {
2      public static int square(int x) {
3          return x * x;
4      }
5  }
6
```

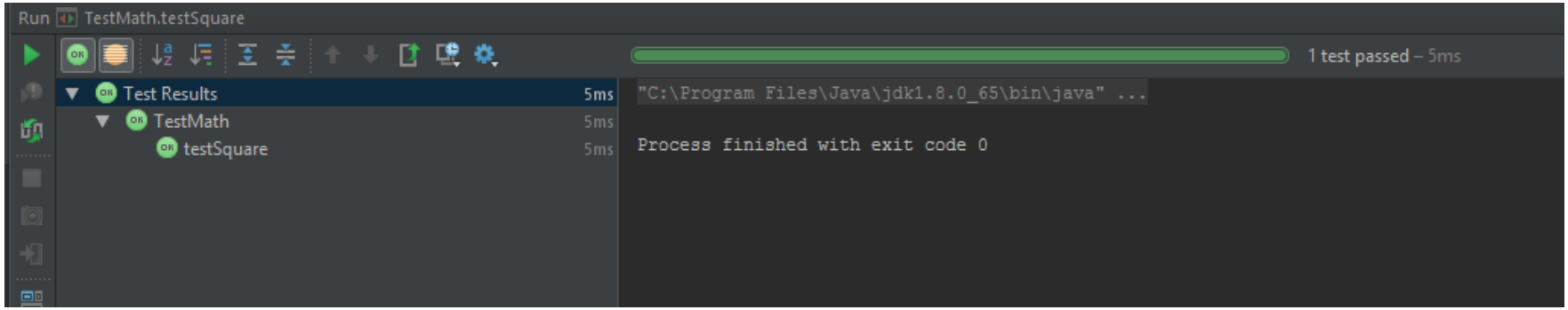
Unit test

```
1  import junit.framework.TestCase;
2
3  public class TestMath extends TestCase{
4      public void testSquare() {
5          assertEquals( expected: 16, Math.square( x: 4));
6      }
7  }
```

Unit testing

GEMINI SOLUTIONS

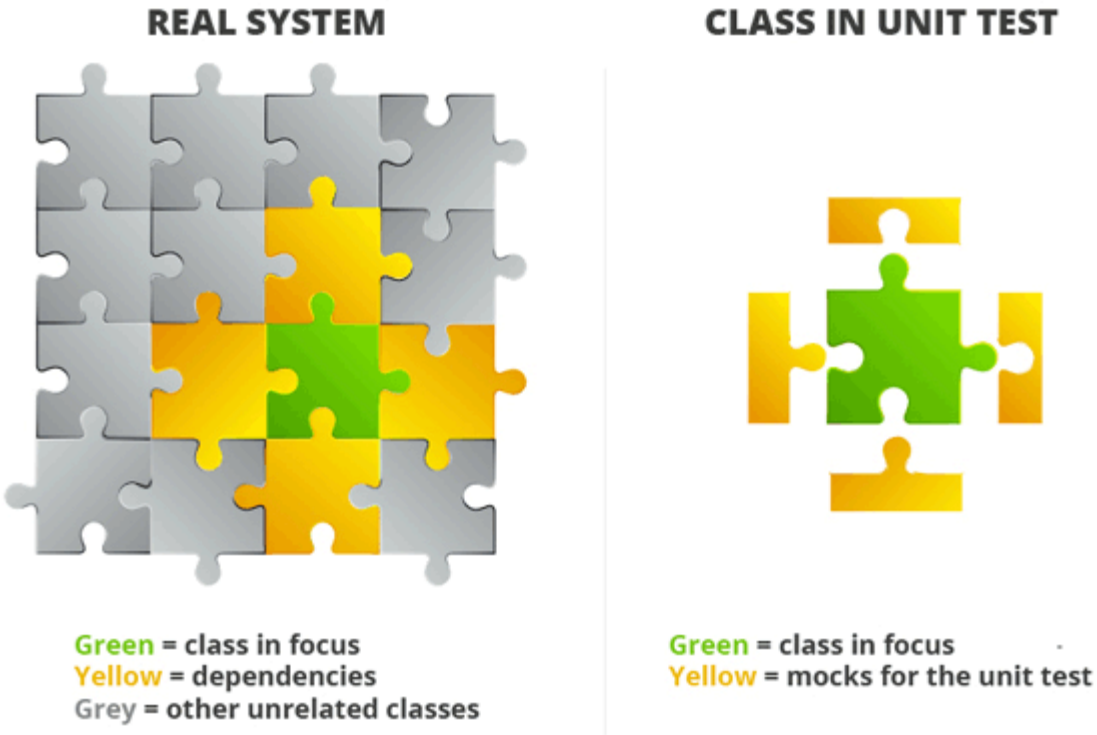
Test passed!



Unit testing

GEMINI SOLUTIONS

- Mocking



Unit testing

GEMINI SOLUTIONS

- Unit testing conclusions:
  - Pros:
    - Finds problems early
    - Easy refactor
    - Simplify integration
    - Better design
  - Cons:
    - Realism
    - Platform difference
    - Decision problem

Regression testing

GEMINI SOLUTIONS

Regression testing is performed when changes are made to the existing functionality of the software or if there is a bug fix in the software. Regression testing can be achieved through multiple approaches, if a *test all* approach is followed it provides certainty that the changes made to the software have not affected the existing functionalities, which are unaltered, in any way.

- Know when things break
- A test for every build (and a build for every commit)
- Automate your build – continuous integration

Jenkins – Continuous integration

https://jenkins.io/

Manual Testing

GEMINI SOLUTIONS

- Coding Process with Manual Tests
  - Write code
  - Upload the code to a location
  - Build it
  - Running the code manually
  - Check Logs, Database, External Services, Values of variable name, Output on screen, etc …
  - If it doesn't work, repeat the above process

Automated Tests

GEMINI SOLUTIONS

- Coding Process with Automated Unit Tests
  - Write one or more test cases
  - Auto-compile and run to see the tests fail
  - Write code to pass the tests
  - Auto-compile and run
  - If tests fail > make appropriate modifications
  - If tests pass > repeat for next build
- Coding Process with Automated Functional Tests
  - Finish writing code (all unit tests pass)
  - Write a Functional Test using any tool
  - Auto-compile and run
  - If tests fail > make appropriate modifications
  - If tests pass > move ahead

Automated Tests vs Manual Tests

GEMINI SOLUTIONS

- Effort and Cost
  - Lets assume we have 10 test cases
  - Effort required to run all 10 manually = 20 min
  - Effort required to write unit tests for all 10 cases = 30 min
  - Effort required to run unit tests for all 10 cases = 1 min
  - Number of testing iterations = 5
  - Total manual testing time = 100 min
  - Total unit testing time = 35 min
  - Adding incremental Unit test cases is cheaper than adding incremental Manual Test Cases

Automated Tests vs Manual Tests

GEMINI SOLUTIONS

- Manual Testing is boring
  - Nobody wants to keep filling the same forms
  - There is nothing new to learn when one tests manually
  - People tend to neglect running manual tests
  - Nobody maintains a list of the tests required to be run if they are manual tests
- Automated Tests on the other hand are code
  - They are fun and challenging to write
  - One has to carefully think of design for reusability and coverage
  - They require analytical and reasoning skills
  - They represent contribution that is usable in the future

Automated Tests vs Manual Tests

GEMINI SOLUTIONS

- Manual Testing is not reusable
  - The effort required is the same each time
  - A tester cannot reuse a Manual Test
- Automated Tests are completely reusable
  - For the above sentence to be true there are required 2 IMPORTANT things: Code Repository and Continuous Integration Server
  - Once written the Automated Tests form a part of the codebase
  - They can be reused without any additional effort for the lifetime of the Project

Automated Tests vs Manual Tests

GEMINI SOLUTIONS

- Manual Testing provide limited Visibility
  - Only the developer testing the code can see the results (more or less …)
- Automated Tests provide global Visibility
  - All team members can login and see Test Results
  - No additional effort required by any of the team members to see tge software works!

Summary

GEMINI SOLUTIONS

- Manual Tests take more effort and cost compared to Automated Test
- Manual Testing is boring (but sometimes necessary)
- Automated Tests are reusable
- Manual Tests provide limited Visibility
- Automated Tests can have varying scopes and can test single units of code by Mocking the dependencies
- Automated tests may require less complex setup and teardown

Summary

GEMINI SOLUTIONS

- Automated Testing ensures you don’t miss out running a test
- Automated Testing can actually enforce a drive clean design decisions
- Automated Tests have Training value
- Automated Tests do not create clutter in code/console/logs

Best Practices

You must use a unit testing framework

You must have an auto-build process, a CI server, auto-testing upon commits

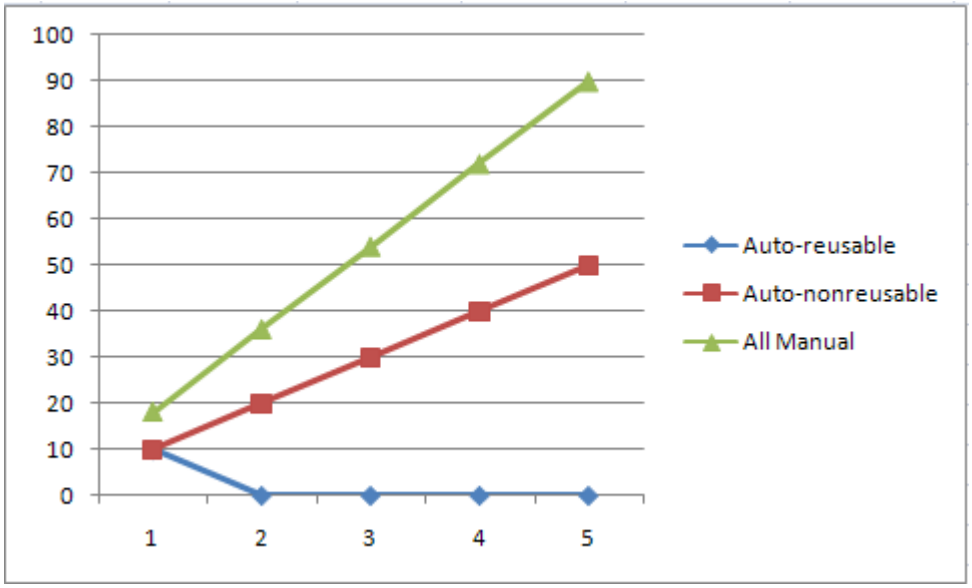
Unit Tests are locally during the day and upon commit by CI Server

Over a period of time you may want to have a CI Server run tests selectively

Tests must be committed along with code

Organize the tests properly

If you do not commit Tests they are not reusable and the reduced effort advantage is lost



Automation frameworks

Curs

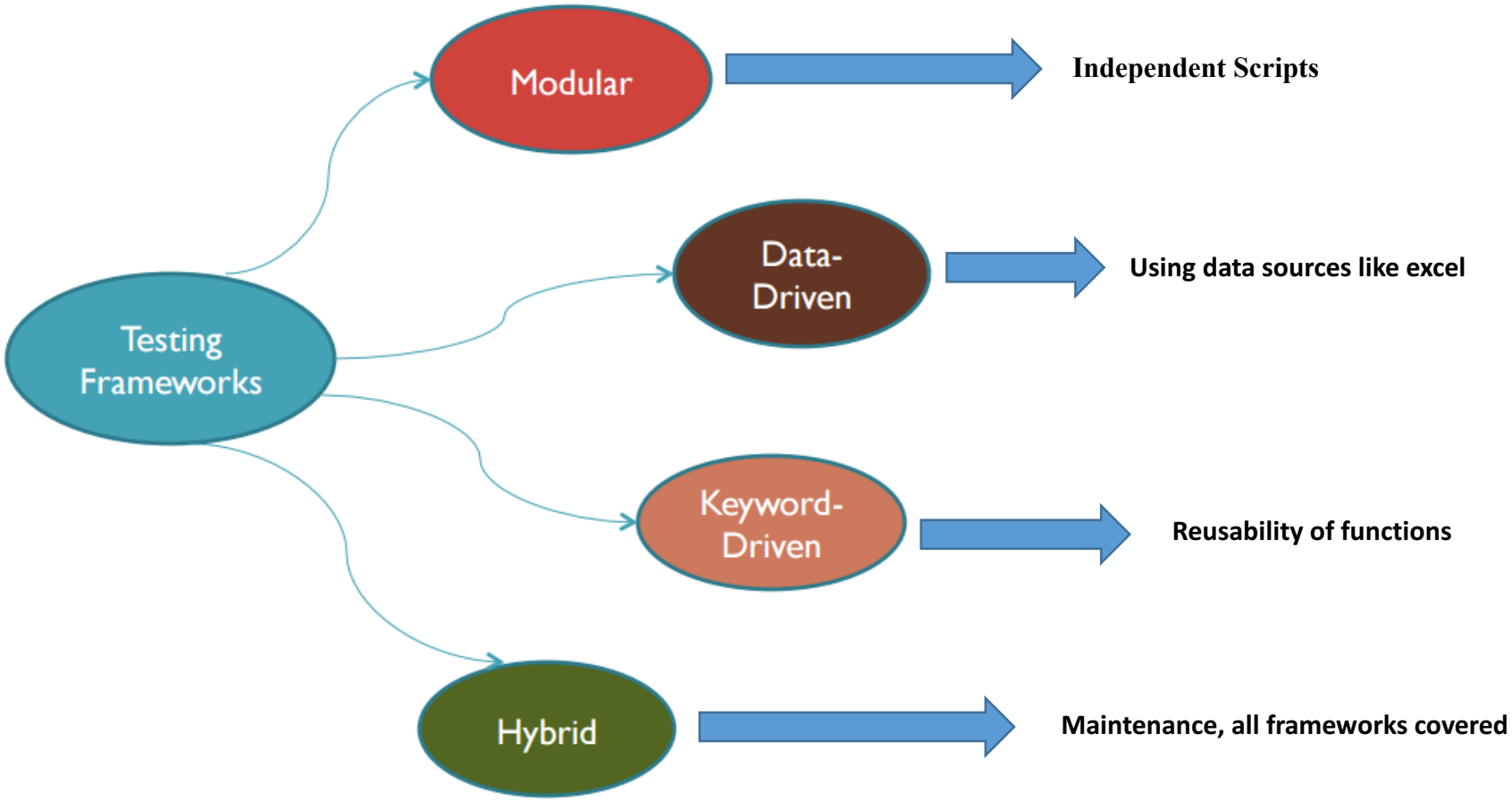
What is a test automation framework?

GEMINI SOLUTIONS

- A test automation framework is an application that allows you to write a series of tests without worrying about the constrains or limitations of the underlying test tools.
- How to choose the correct framework
  - Technology (Java, Python, Perl …)
  - Continuous Integration
  - IDE
  - Data sources
  - Logging
  - Reporting

Types of Automation Testing Frameworks

GEMINI SOLUTIONS



Automation Frameworks

GEMINI SOLUTIONS

- Selenium WebDriver Framework
- Robot Framework
- Sikuli
- Serenity
- SpecFlow BDD Automation
- Cucumber (BDD)
- Golem Automated Testing Framework
- Gauge

Selenium

GEMINI SOLUTIONS

- Selenium is a Functional Automation tool for Web applications.
- Selenium is an open source tool (No cost Involved in it).
- Selenium supports the languages like HTML, Java, PHP, Perl, Python, Ruby and C#.
- It supports the browsers like IE, Mozilla Firefox, Safari, Google Chrome and Opera.
- It supports the operating systems like Windows, Linux and Mac.
- It is very flexible when compared to QTP and other functional tools, because it supports multiple languages.

Selenium

GEMINI SOLUTIONS

- Components:
  - Selenium is mainly built on 3 components:
    1. Selenium IDE
    2. Selenium RC
    3. Selenium Grid

Selenium IDE

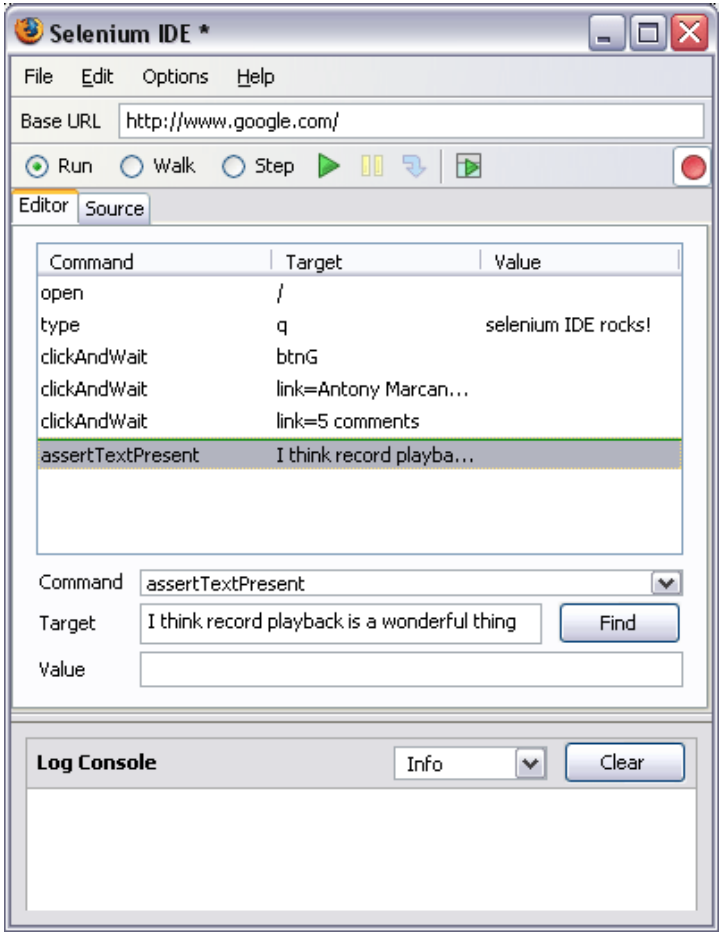
IDE stands for Integrated Development Environment.

Which is used for Record and Play back the scripts.

It is an Add on for Mozilla Firefox, which means we can download the Selenium IDE from Mozilla Firefox and we can Record and Run the scripts in Mozilla Firefox only.

Selenium IDE is accountable for user actions.

We can Run the Recorded scripts against other browsers by using Selenium RC.



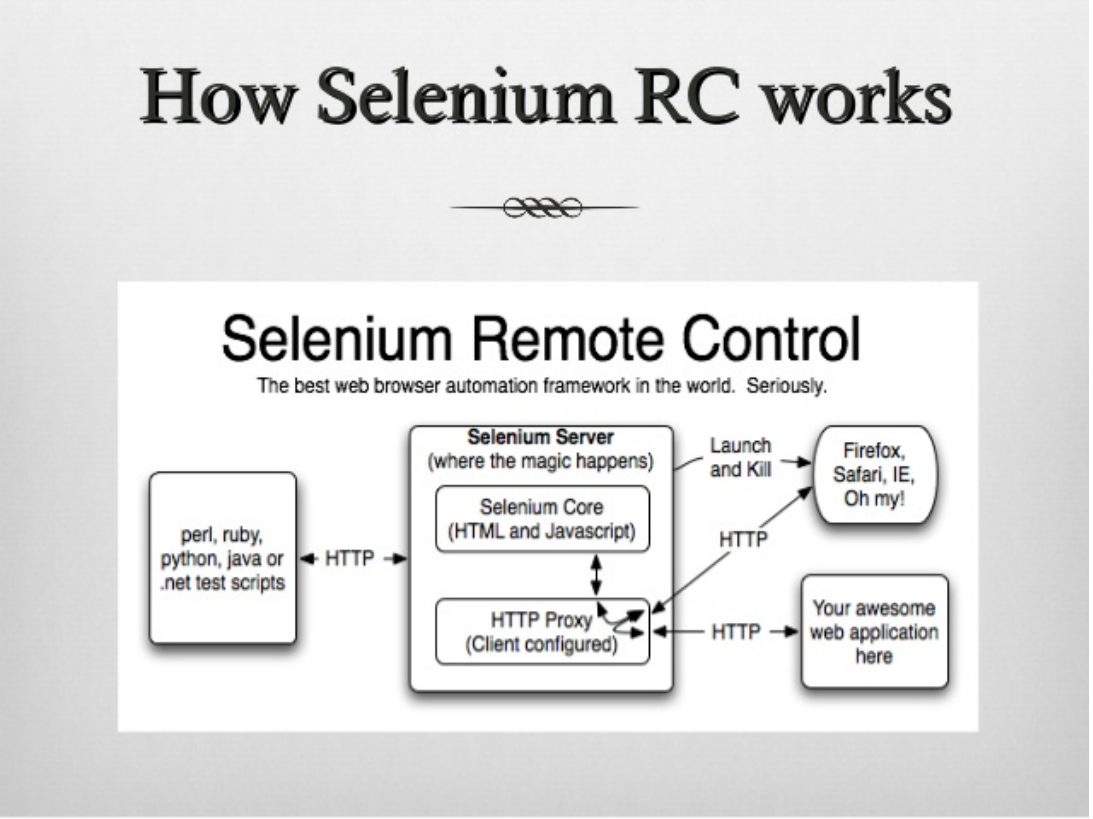
Selenium RC

RC stands for Remote Control.

It is a Server and launches the Browser.

It acts as a API and Library of Selenium.

We need to configure the Selenium RC with the supported language, then we can automate the application.

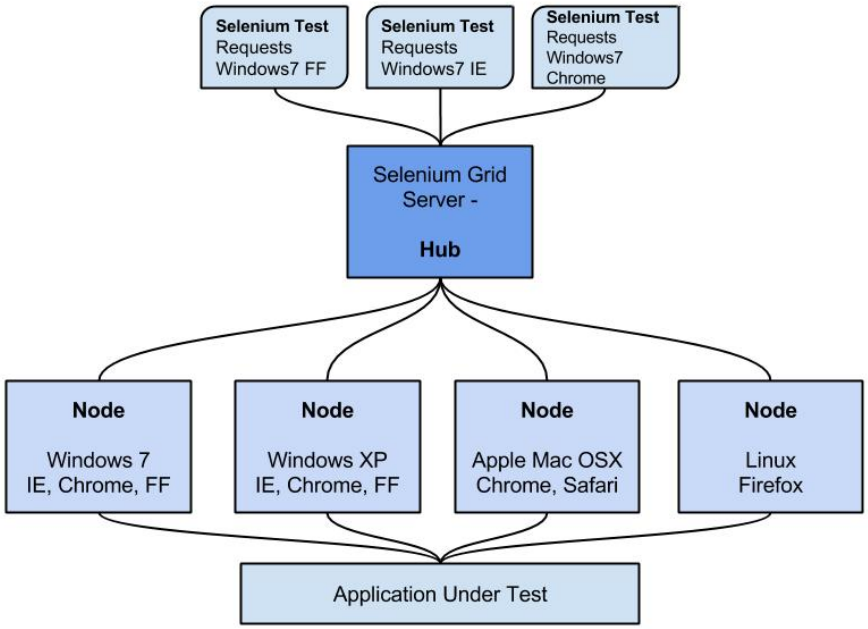


Selenium Grid

Selenium Grid is used for launching the multiple browsers with supported operating system in parallel.

We can run the scripts on different browsers in parallel.

It allows you to easily run multiple tests in parallel, on multiple machines, in a heterogeneous environment.



Selenium samples

GEMINI SOLUTIONS

- Setting up the driver:

```
@Before
public void setUp() throws Exception {
    driver = new FirefoxDriver();
    FirefoxProfile profile = new FirefoxProfile();
    profile.setPreference("intl.accept_languages", "gr");
    grdriver = new FirefoxDriver(profile);
    randomEmail = (new Random()).nextInt( bound: 9900) + 100;
    contorNumberOfVideos = 0;
    this.homeUrl = "https://stage.irewind.com";
    this.newsLetterUrlNew = "https://stage.irewind.com/public/video/0f8de6230eeadfbeb5d6c77ccddb598d";
    this.testURL = "https://stage.irewind.com/go/code/ddcbc4358102903df8d7d6e00e8fb695";
    this.newsLetterUrlOld = "https://stage.irewind.com/public/video/618f2627e1892d0f389aa1ec95df4c4b";
    this.emailUrl = "https://gmail.com";
    this.presaleUrl = "https://stage.irewind.com/go/presale/channel/218";
    driver.manage().window().maximize();
    driver.manage().timeouts().implicitlyWait( l: 5L, TimeUnit.SECONDS);
    grdriver.manage().window().maximize();
    grdriver.manage().timeouts().implicitlyWait( l: 5L, TimeUnit.SECONDS);
}

@After
public void tearDown() throws Exception {
    driver.quit();
    grdriver.quit();
}
```

Robot Framework

GEMINI SOLUTIONS

- http://robotframework.org/
- Generic test automation framework
  - Utilizes the keyword-driven testing approach
  - Suitable for both “normal” test automation and ATDD (Acceptance test driven development)
- Implemented with Python
  - Runs also on Jython (JVM) and IronPython (.NET)
  - Can be extended natively using Python or Java
  - Other languages supported via a remote interface
- Open source
  - Hosted on GitHub, Apache 2 license
  - Sponsored by Nokia Networks
  - Rich ecosystem and very active community

Robot Framework

GEMINI SOLUTIONS

- Simple Keyword-driven syntax

```
*** Test Cases ***
Valid Login
    Open Browser To Login Page
    Input Username    demo
    Input Password    mode
    Submit Credentials
    Welcome Page Should Be Open
    [Teardown]    Close Browser
```

Robot Framework

GEMINI SOLUTIONS

- Data-Driven tests

```
*** Test Cases ***      USER NAME         PASSWORD
Invalid Username        invalid           ${VALID PWD}
Invalid Password        ${VALID USER}     invalid
Both Invalid            invalid           whatever
Empty Username          ${EMPTY}          ${VALID PWD}
Empty Password          ${VALID USER}     ${EMPTY}
Both Empty              ${EMPTY}          ${EMPTY}
```

Robot Framework

GEMINI SOLUTIONS

- High Level Keywords

```
*** Keywords ***
Open Browser To Login Page
    Open Browser    ${LOGIN URL}    ${BROWSER}
    Maximize Browser Window
    Set Selenium Speed    ${DELAY}
    Login Page Should Be Open

Login Page Should Be Open
    Title Should Be    Login Page

Input Username
    [Arguments]    ${username}
    Input Text    username_field    ${username}

Input Password
    [Arguments]    ${password}
    Input Text    password_field    ${password}
```

Robot Framework

GEMINI SOLUTIONS

- Variables
  - Easy to create:

```
*** Variables ***
${BROWSER}          Firefox
${HOST}             localhost:7272
${LOGIN URL}        http://${HOST}/
${WELCOME URL}      http://${HOST}/welcome.html
```

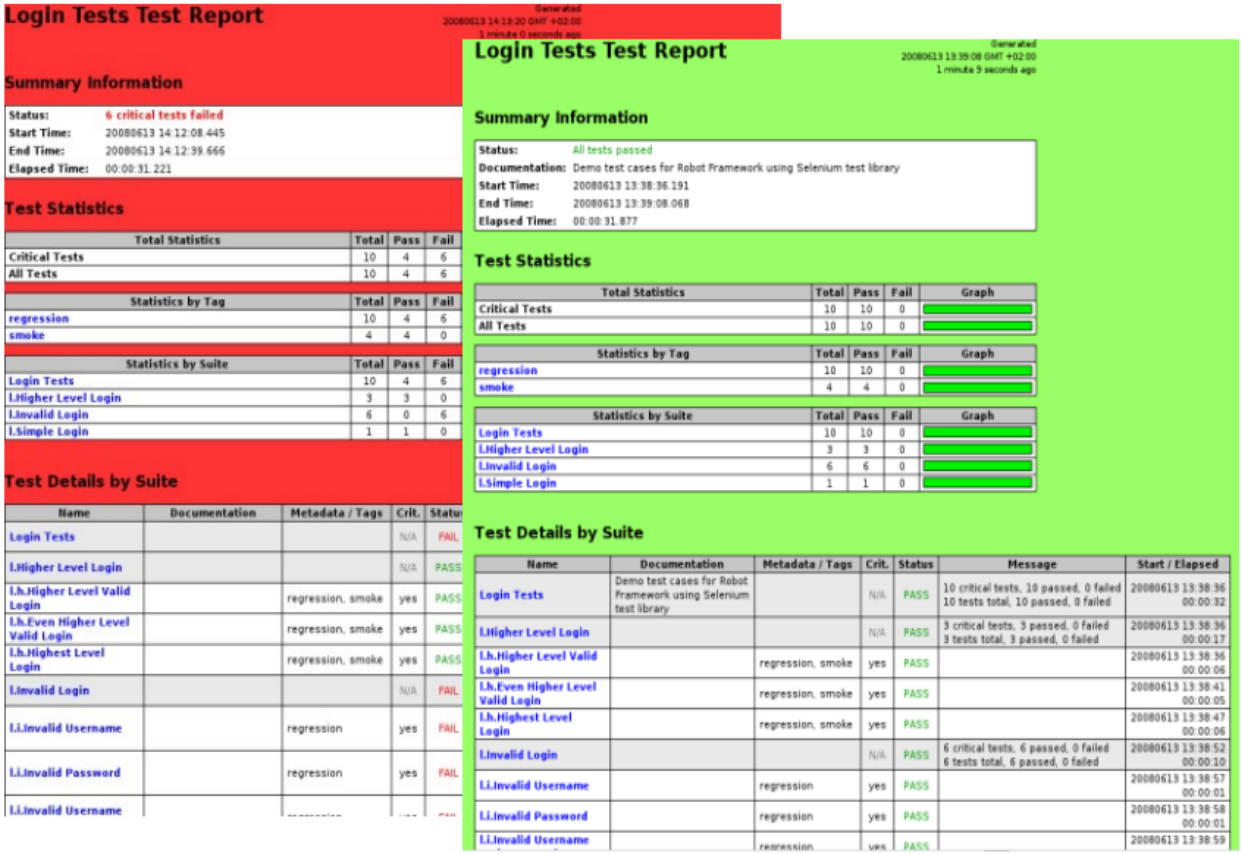
  - Override from the command line:

```
--variable BROWSER:IE
```

Robot Framework

GEMINI SOLUTIONS

- Reports



Login Tests Test Report

Generated
20080613 14:13:20 GMT +02:00
1 minute 0 seconds ago

Summary Information

Status:	6 critical tests failed
Start Time:	20080613 14:12:08.445
End Time:	20080613 14:12:39.666
Elapsed Time:	00:00:31.221

Test Statistics

Total Statistics	Total	Pass	Fail
Critical Tests	10	4	6
All Tests	10	4	6

Statistics by Tag	Total	Pass	Fail
regression	10	4	6
smoke	4	4	0

Statistics by Suite	Total	Pass	Fail
Login Tests	10	4	6
I.Higher Level Login	3	3	0
I.Invalid Login	6	0	6
I.Simple Login	1	1	0

Test Details by Suite

Name	Documentation	Metadata / Tags	Crit.	Statu
Login Tests			N/A	FAIL
I.Higher Level Login			N/A	PASS
I.h.Higher Level Valid Login		regression, smoke	yes	PASS
I.h.Even Higher Level Valid Login		regression, smoke	yes	PASS
I.h.Highest Level Login		regression, smoke	yes	PASS
I.Invalid Login			N/A	FAIL
I.i.Invalid Username		regression	yes	FAIL
I.i.Invalid Password		regression	yes	FAIL
I.i.Invalid Username		regression	yes	FAIL

Login Tests Test Report

Generated
20080613 13:39:08 GMT +02:00
1 minute 9 seconds ago

Summary Information

Status:	All tests passed
Documentation:	Demo test cases for Robot Framework using Selenium test library
Start Time:	20080613 13:38:36.191
End Time:	20080613 13:39:08.068
Elapsed Time:	00:00:31.877

Test Statistics

Total Statistics	Total	Pass	Fail	Graph
Critical Tests	10	10	0	
All Tests	10	10	0	

Statistics by Tag	Total	Pass	Fail	Graph
regression	10	10	0	
smoke	4	4	0	

Statistics by Suite	Total	Pass	Fail	Graph
Login Tests	10	10	0	
I.Higher Level Login	3	3	0	
I.Invalid Login	6	6	0	
I.Simple Login	1	1	0	

Test Details by Suite

Name	Documentation	Metadata / Tags	Crit.	Status	Message	Start / Elapsed
Login Tests	Demo test cases for Robot Framework using Selenium test library		N/A	PASS	10 critical tests, 10 passed, 0 failed 10 tests total, 10 passed, 0 failed	20080613 13:38:36 00:00:32
I.Higher Level Login			N/A	PASS	3 critical tests, 3 passed, 0 failed 3 tests total, 3 passed, 0 failed	20080613 13:38:36 00:00:17
I.h.Higher Level Valid Login		regression, smoke	yes	PASS		20080613 13:38:36 00:00:06
I.h.Even Higher Level Valid Login		regression, smoke	yes	PASS		20080613 13:38:41 00:00:05
I.h.Highest Level Login		regression, smoke	yes	PASS		20080613 13:38:47 00:00:06
I.Invalid Login			N/A	PASS	6 critical tests, 6 passed, 0 failed 6 tests total, 6 passed, 0 failed	20080613 13:38:52 00:00:10
I.i.Invalid Username		regression	yes	PASS		20080613 13:38:57 00:00:01
I.i.Invalid Password		regression	yes	PASS		20080613 13:38:58 00:00:01
I.i.Invalid Username		regression	yes	PASS		20080613 13:38:59

That's all folks!

GEMINI SOLUTIONS

Feedback & Learnings & Questions

Knowledge is the Most Beautiful of Awards