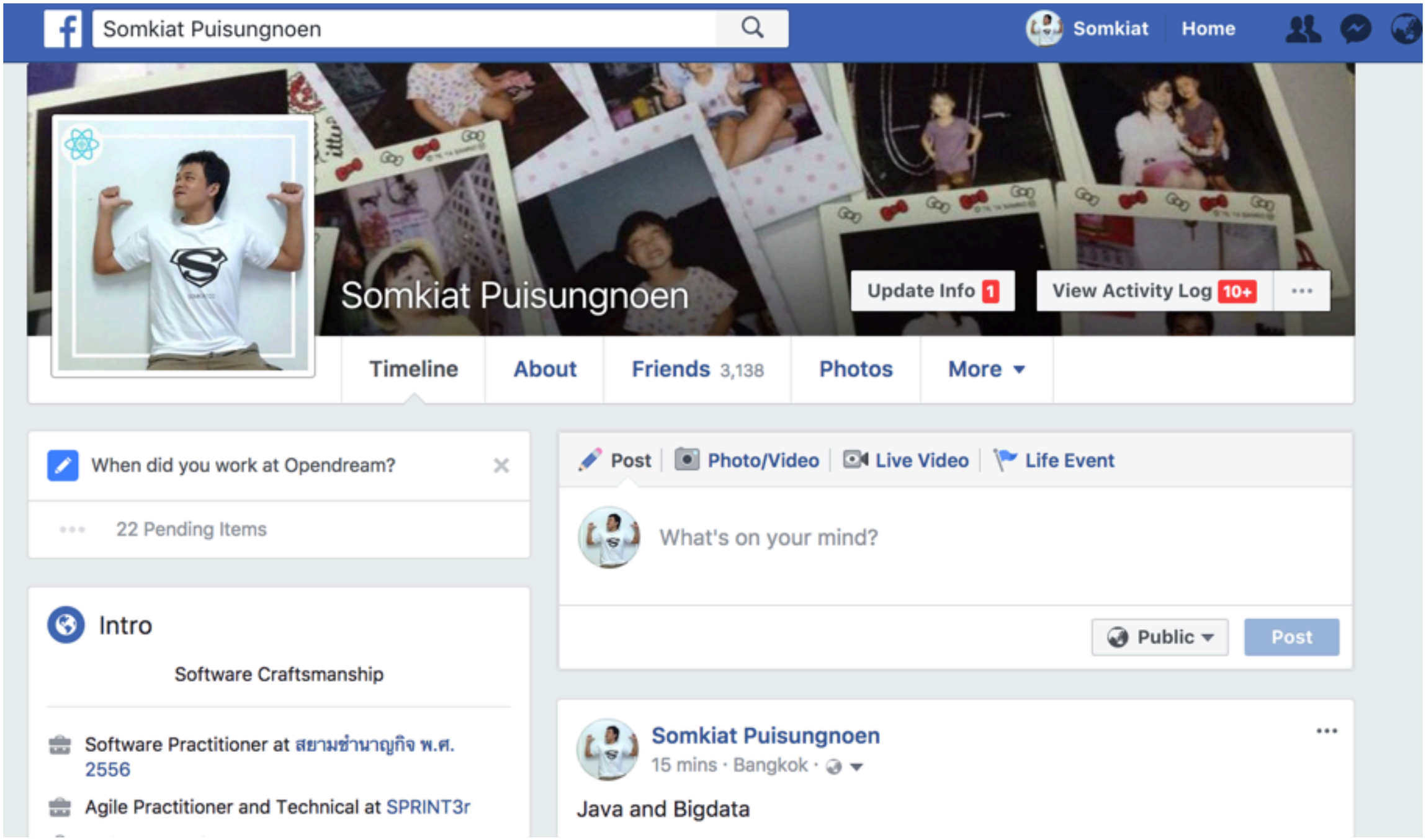
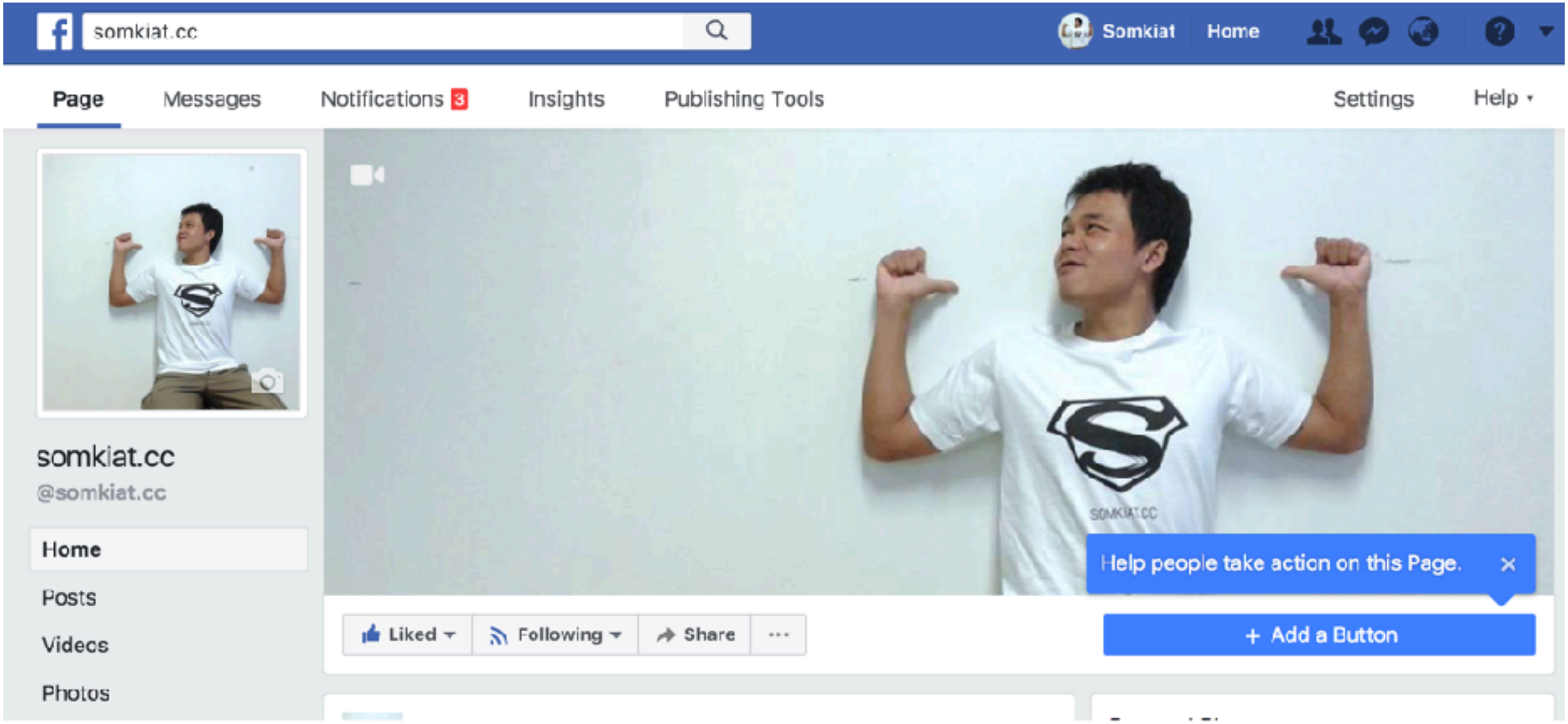
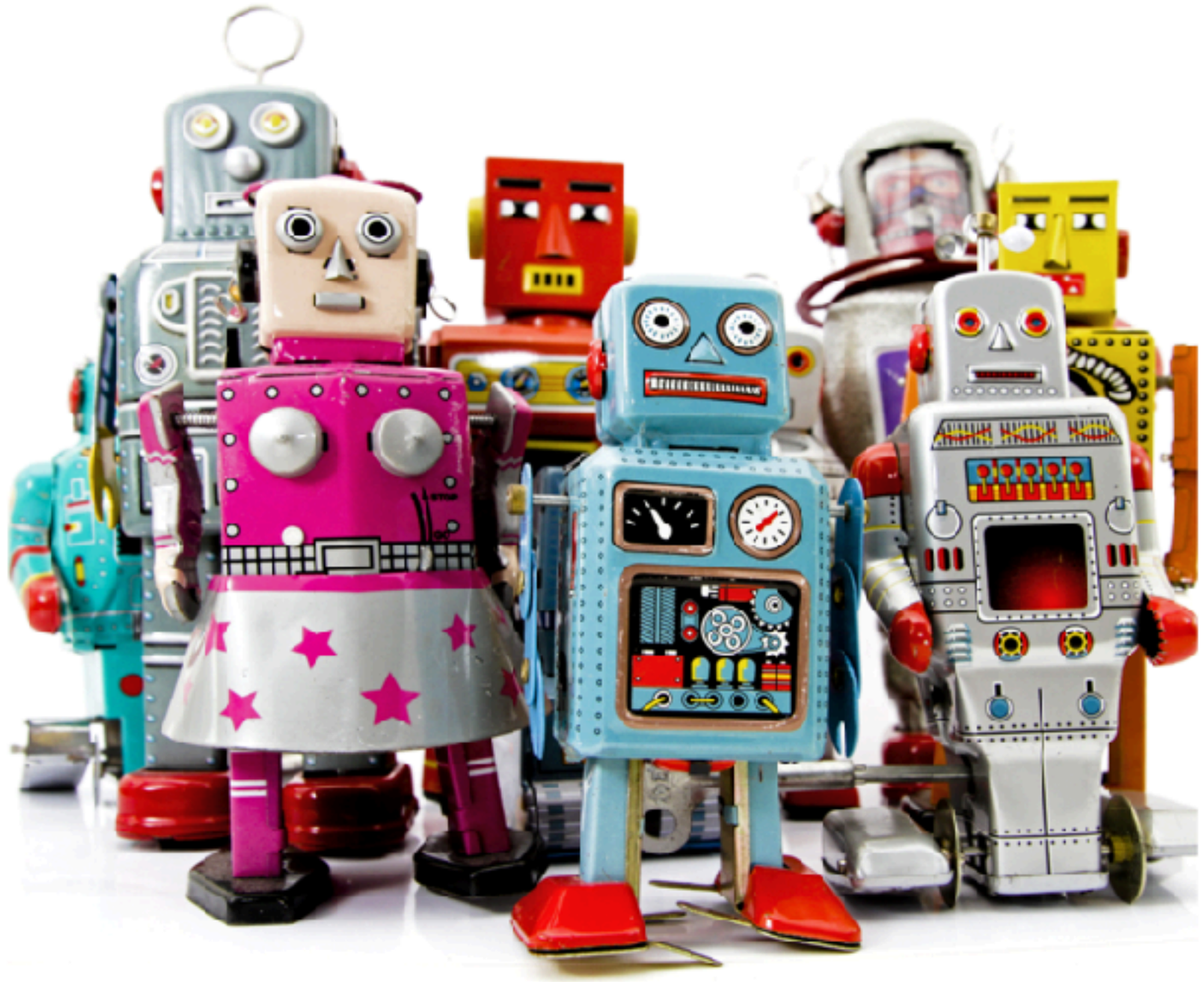
สยามชำนาญกิจ
SIAM CHAMNAN KIT

Robot Framework workshop





https://github.com/up1/course-robotframework



Robot Framework Workshop

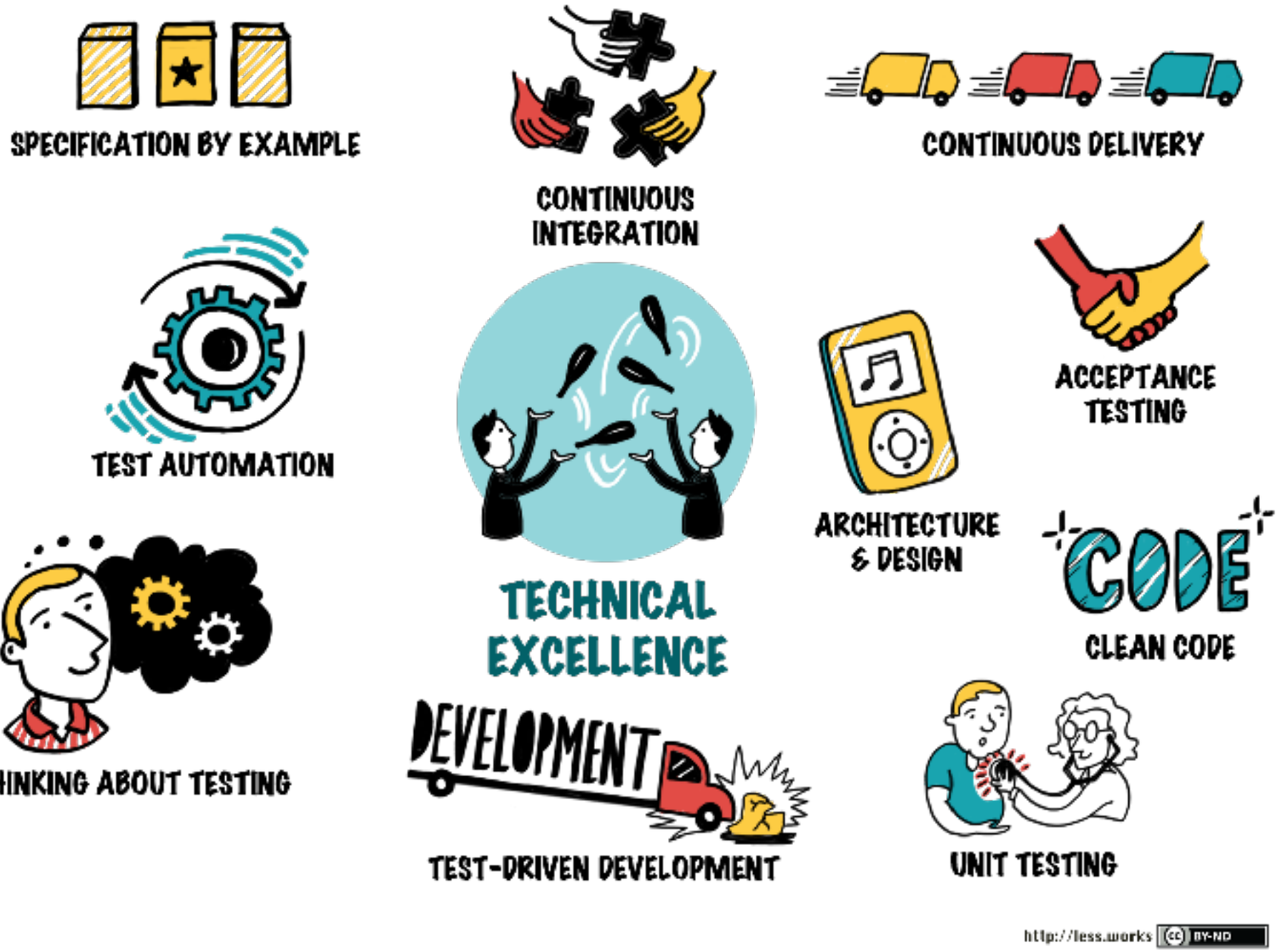
Agenda

- Acceptance Test-Driven Development (ATDD)
- Test strategies
- Introduction to Robot Framework
- Structure of test case
- Working with Web application
- Workshop

Agenda

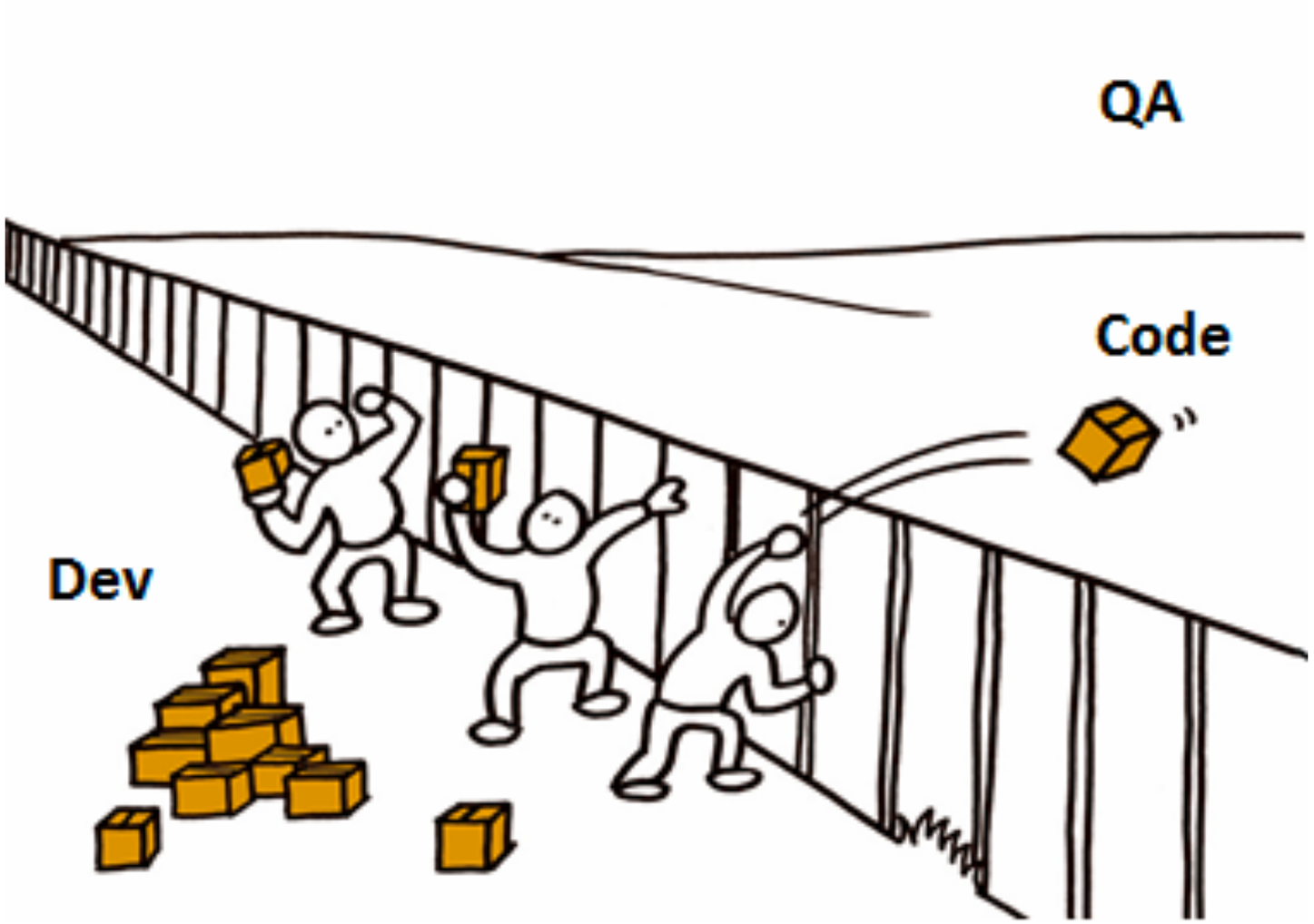
- Test life cycle of Robot Framework
- Better test case
- Better test structure with Page Object pattern
- Command lines
- Scaling test with parallel and distributed
- Design pipeline with automated test
- Workshop

Technical Excellence

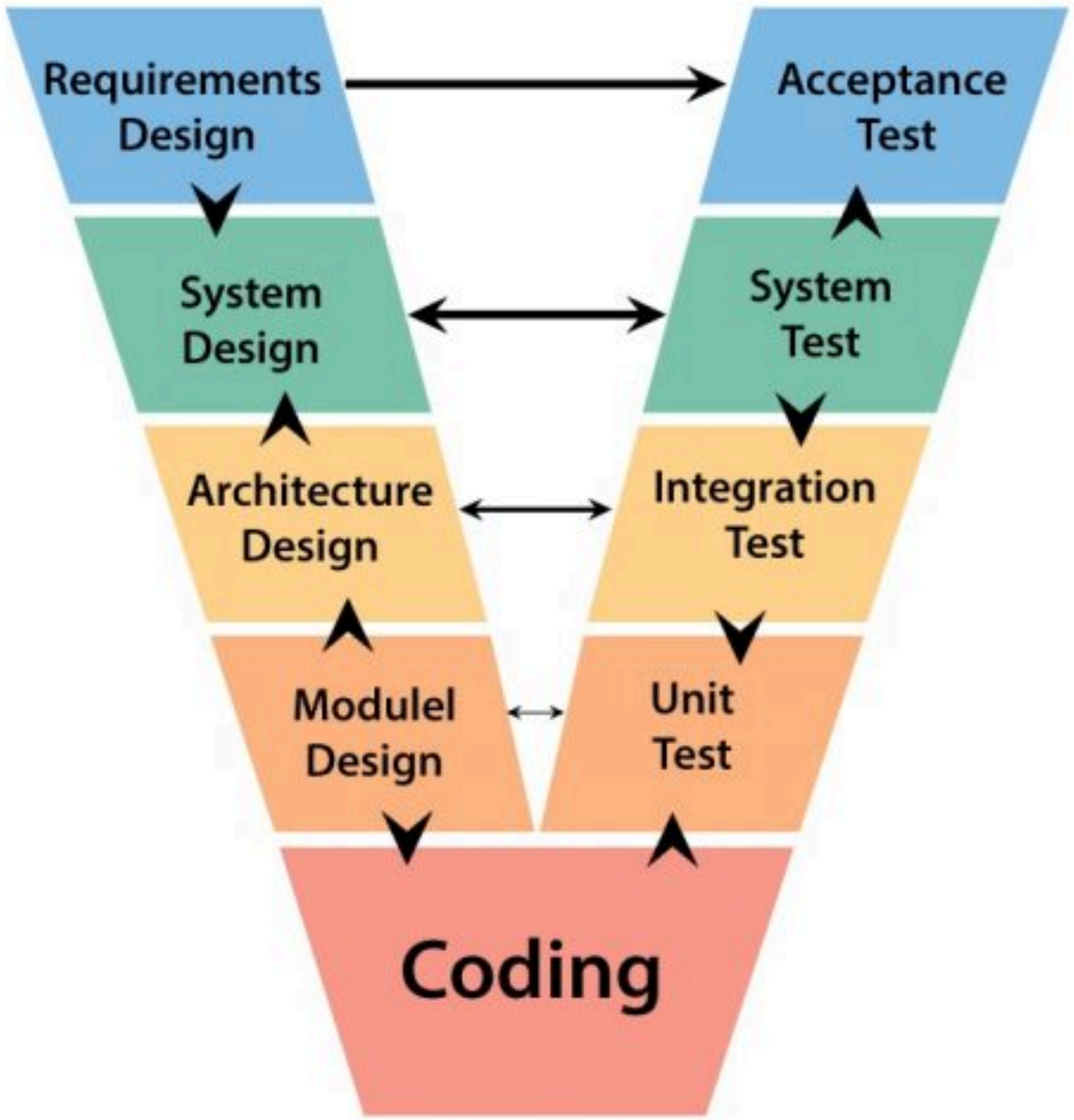


https://less.works/less/technical-excellence

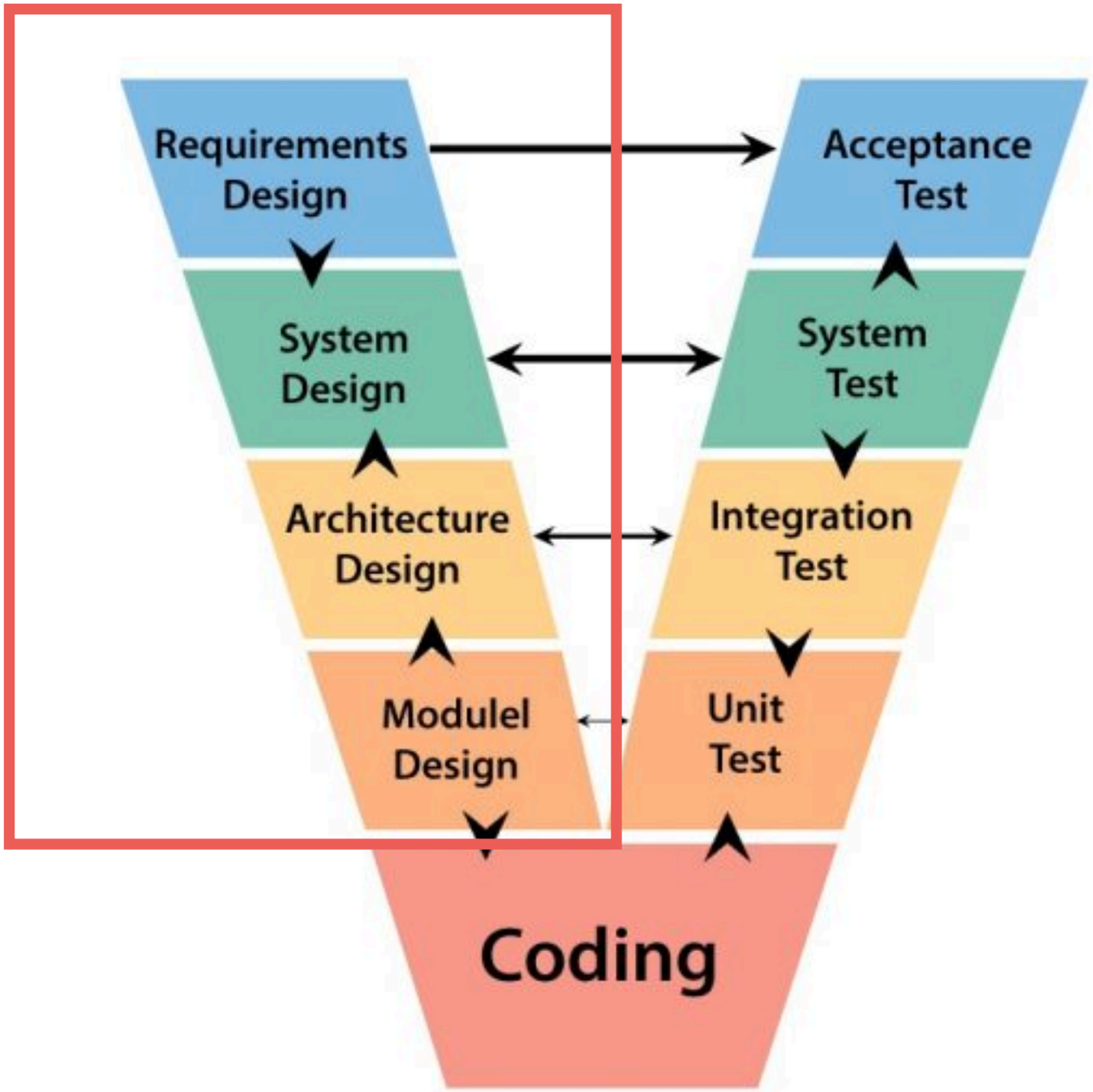
Software Development



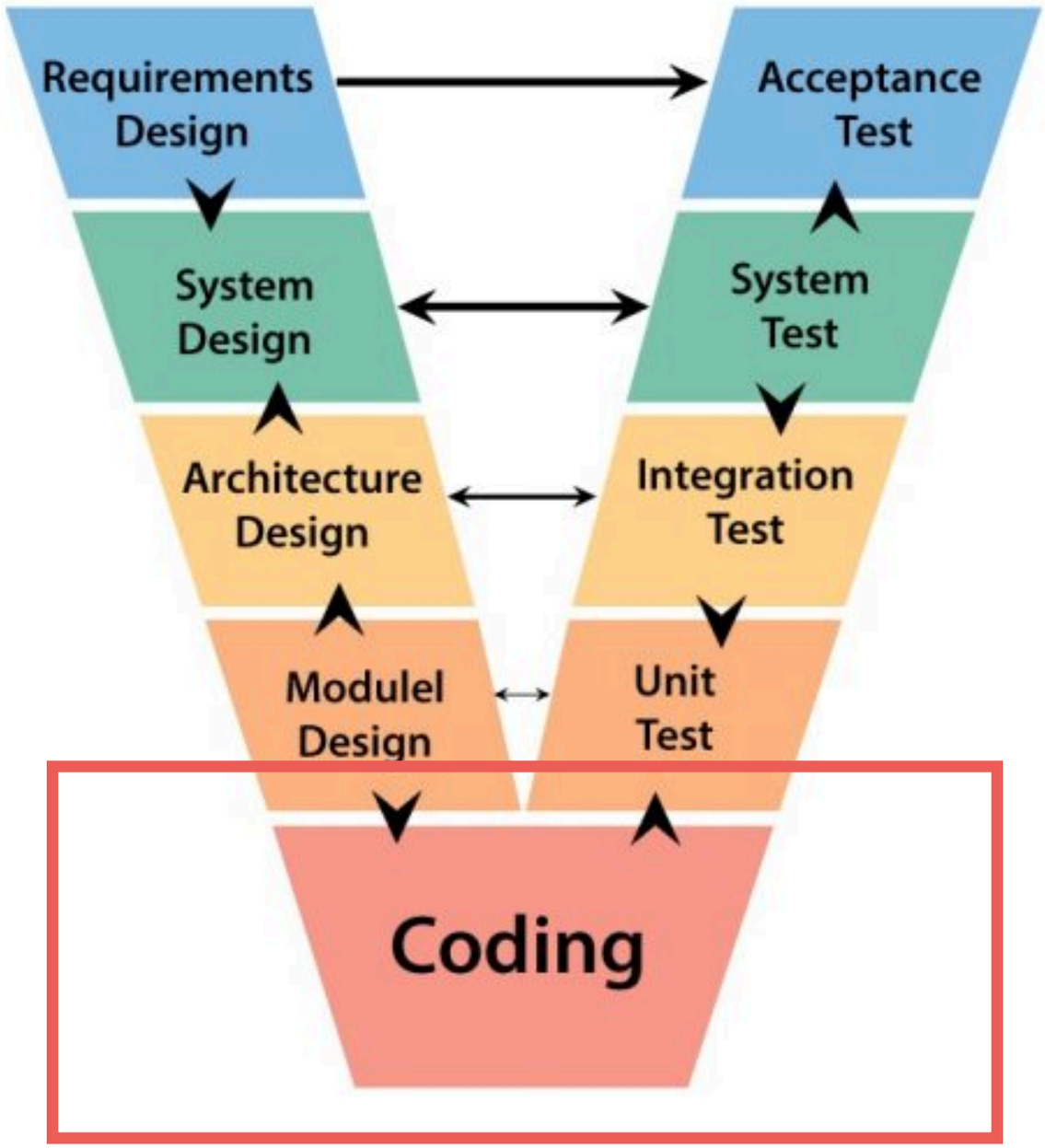
V Model or Waterfall Model



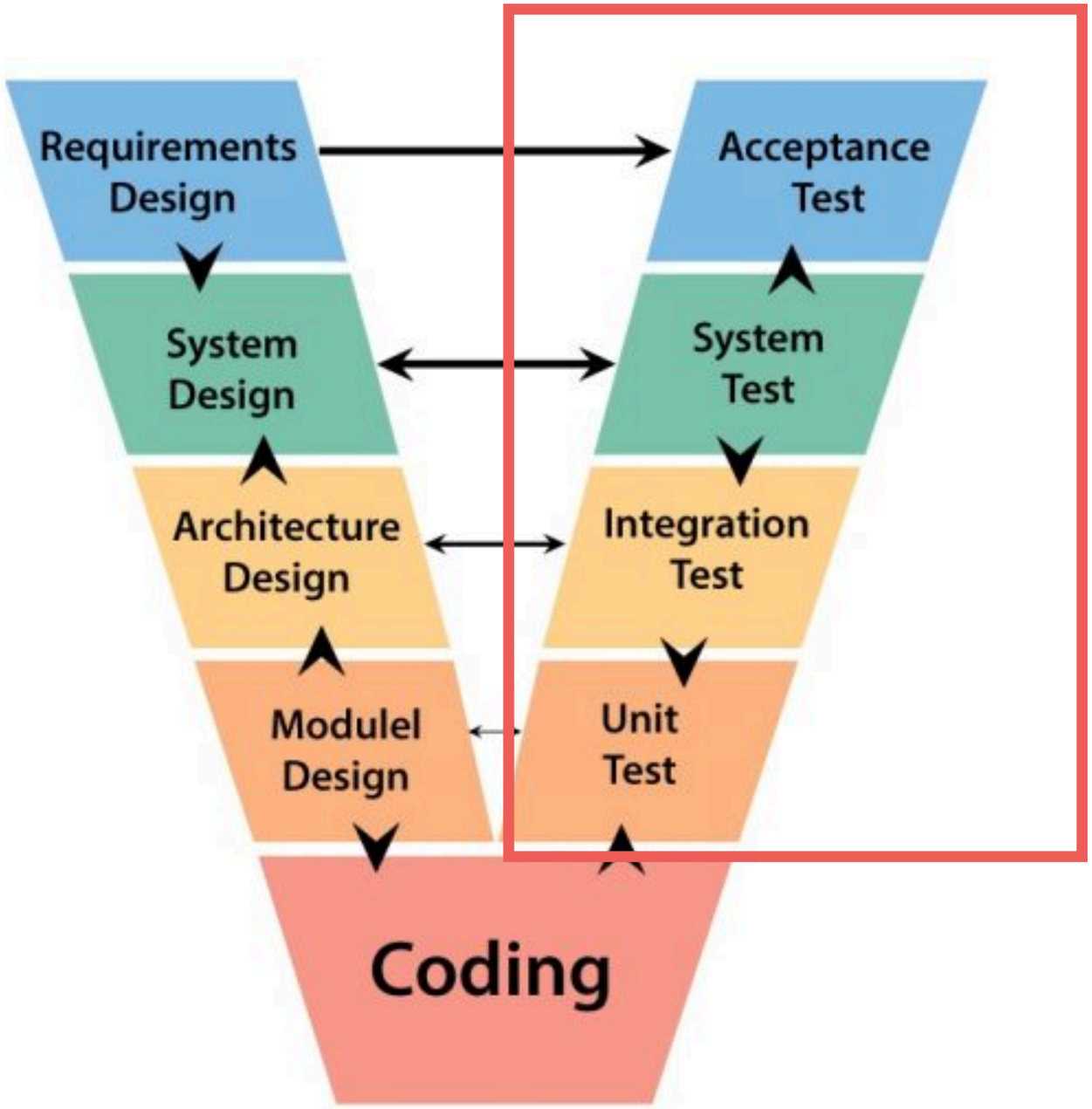
Verification phase



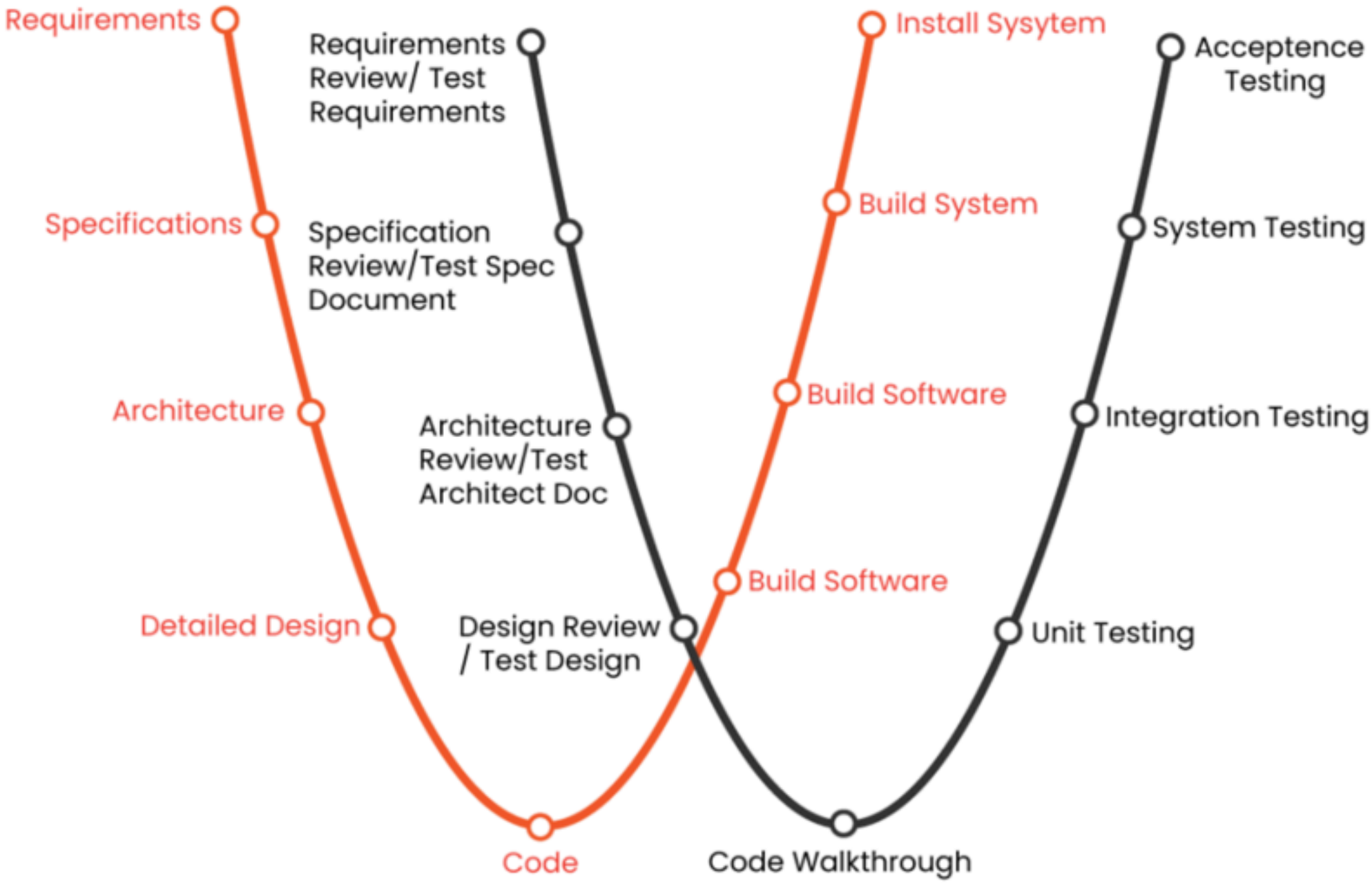
Coding phase



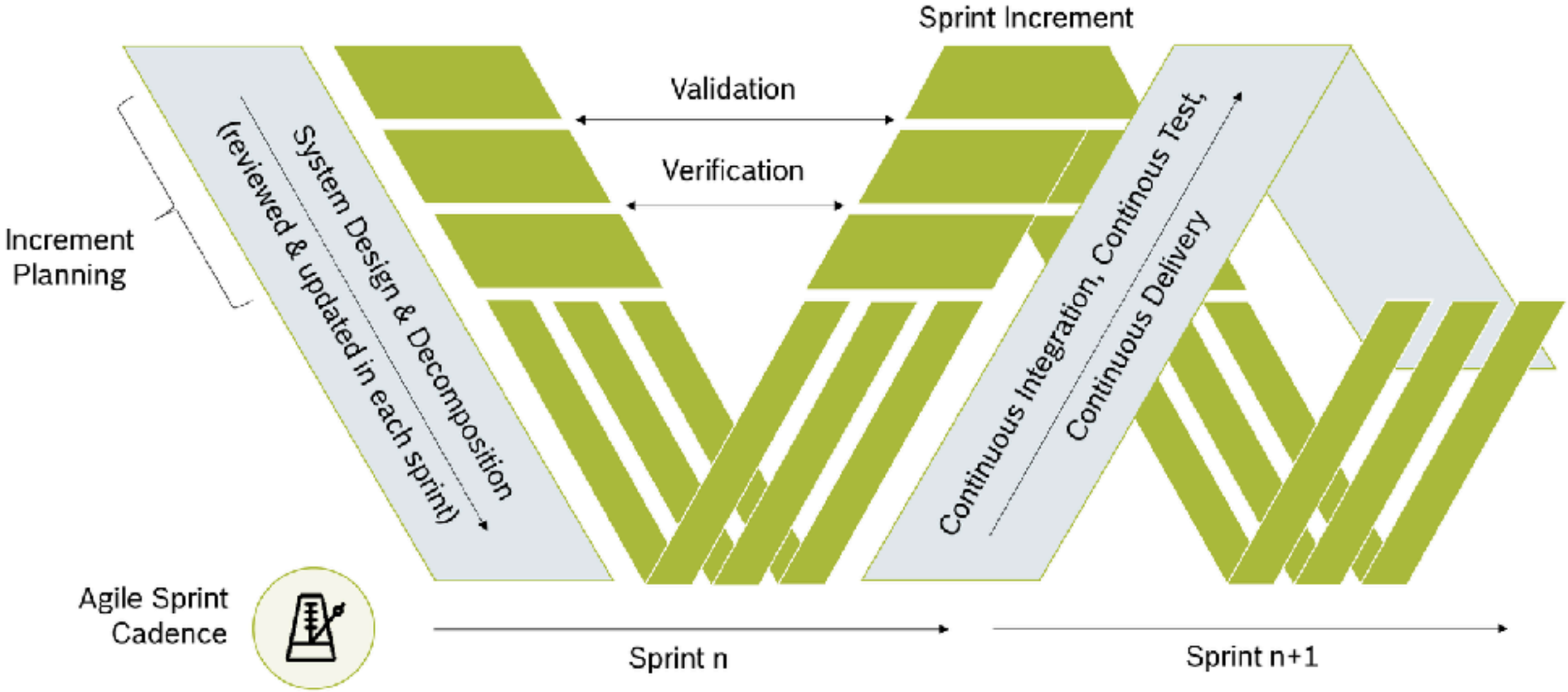
Validation phase



W Model

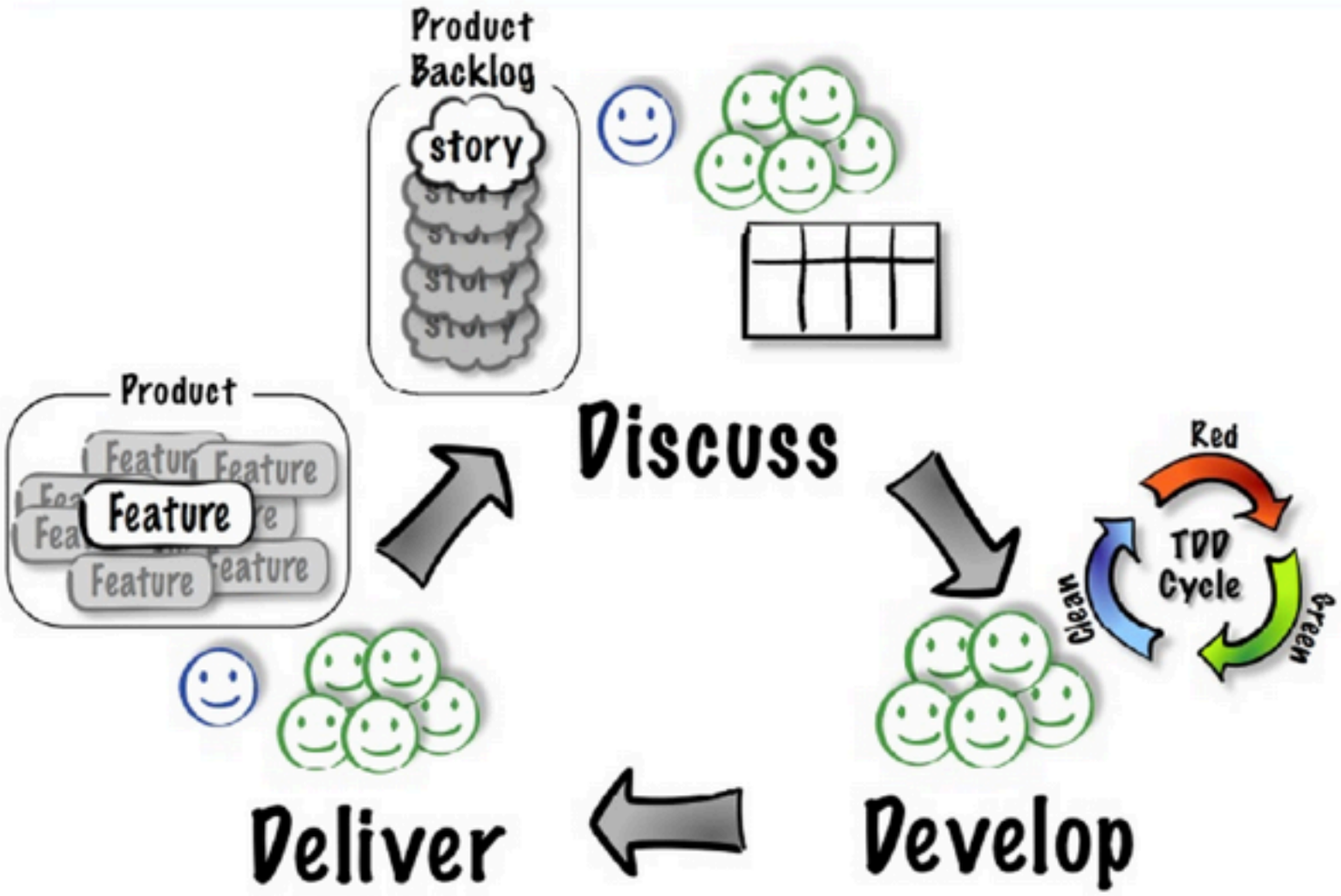


Iterative and Incremental



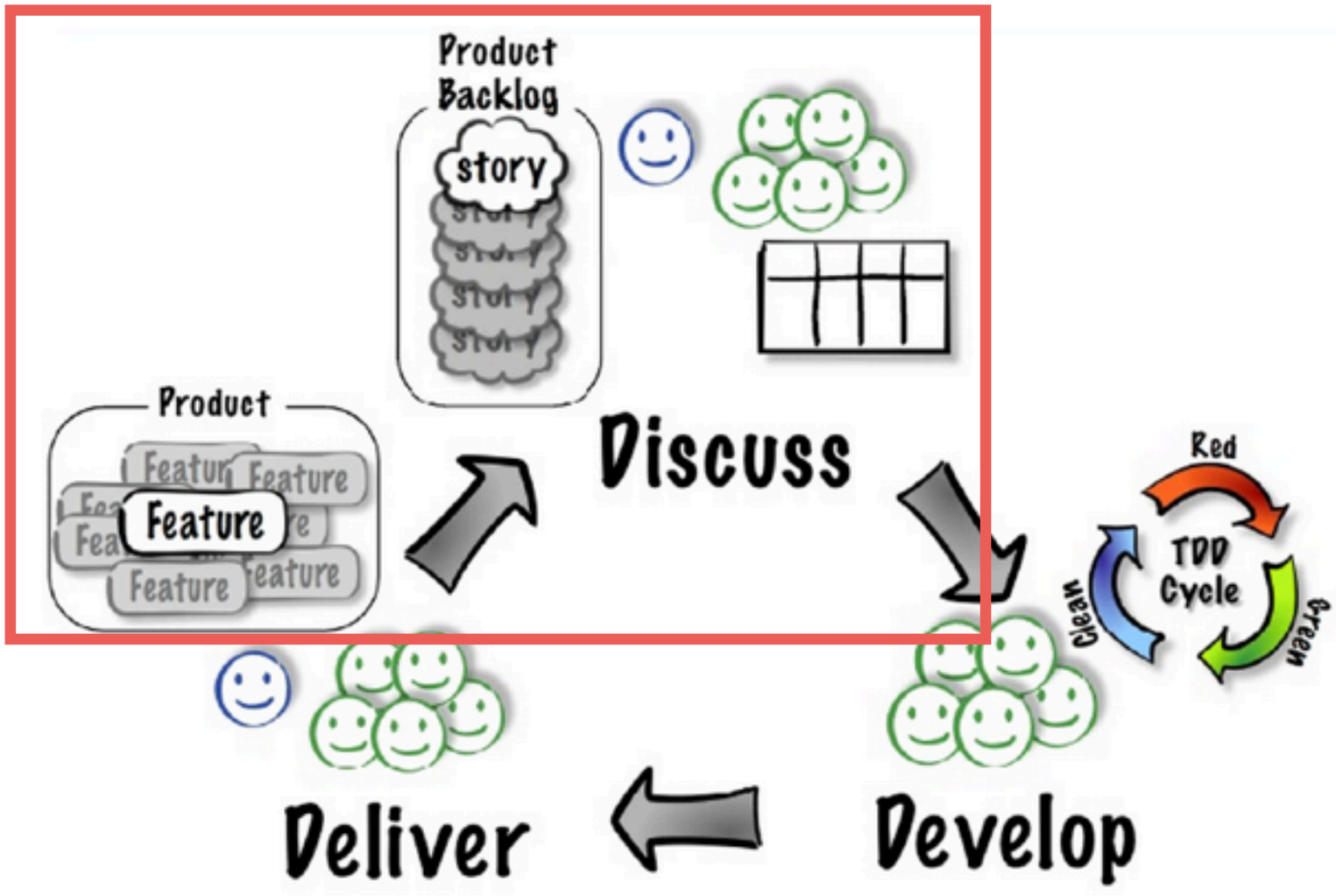
Acceptance Test-Driven Development (ATDD)

Acceptance Test-Driven Development (ATDD)



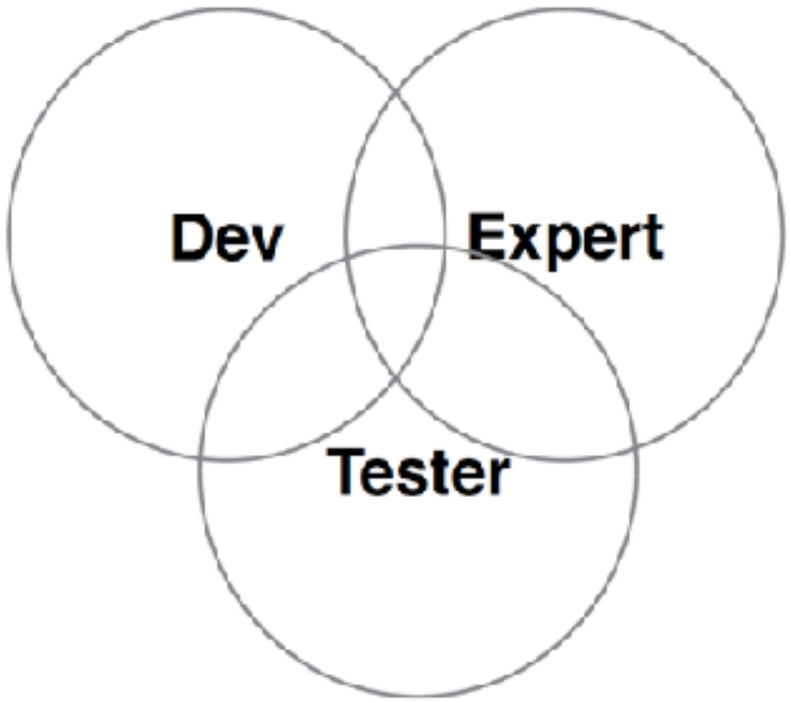
Acceptance Test-Driven Development (ATDD)

Discussion process



Discussion process

Whole team approach
Share understanding
Clarify solution
Concrete example/data



Decompose or Slicing feature

Feature

Feature 1

Feature 2

Feature 3

Decompose or Slicing feature

Feature **Flow/User Story**

Feature 1	Flow 1.1
Feature 2	Flow 1.2
Feature 3	Flow 1.3

Decompose or Slicing feature

Feature	Flow/User Story	Task	
Feature 1	Flow 1.1	Task 1.1.1	Task 1.1.4
Feature 2	Flow 1.2	Task 1.1.2	Task 1.1.5
Feature 3	Flow 1.3	Task 1.1.3	Task 1.1.6

Example

Login process

Login page

https://demo-login-workshop.vercel.app

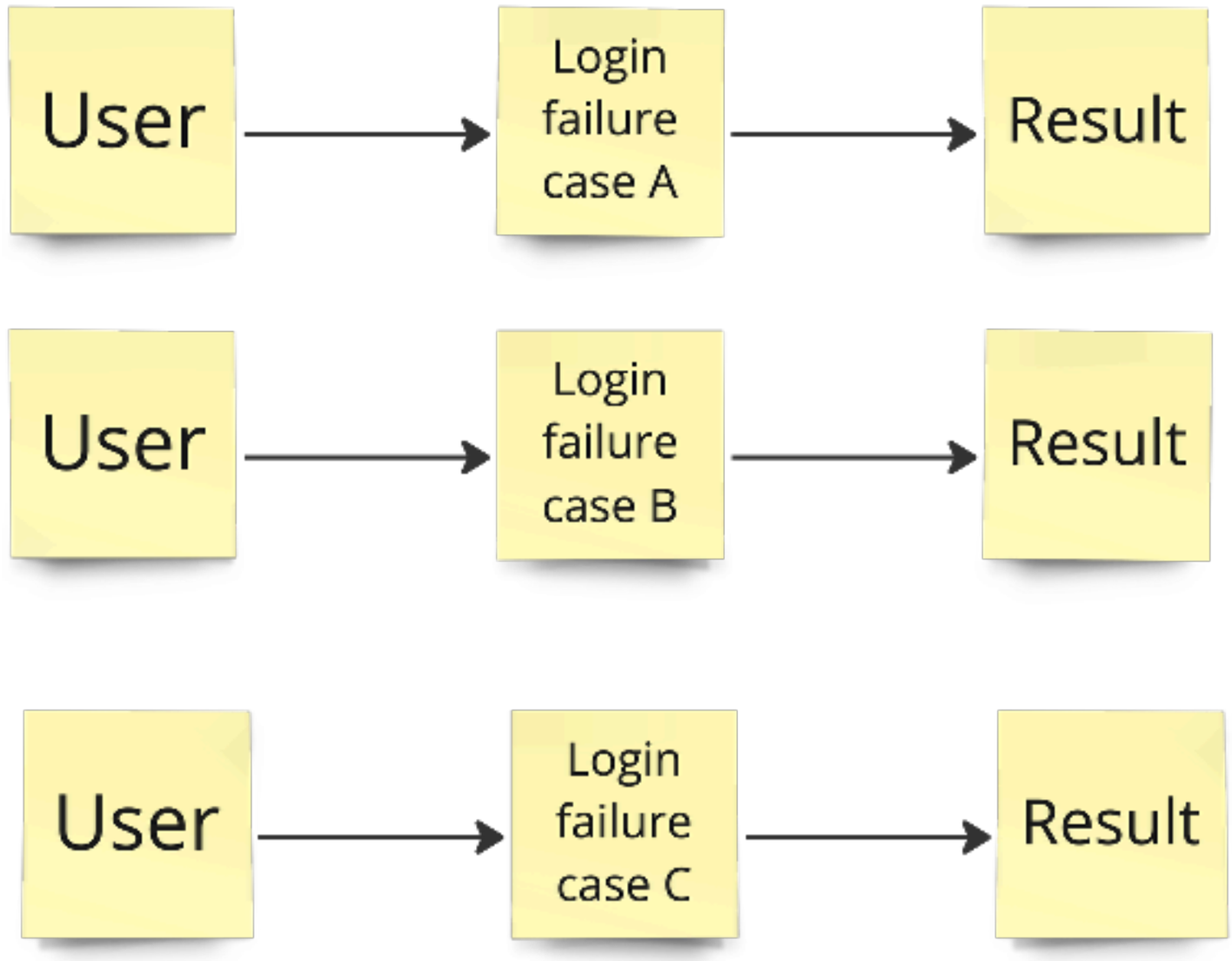
username=**demo**
password=**mode**

Result page

Welcome and Error page

Slicing process

Slicing process for fail case



Test Case =
Business logic + Example Data

Test Cases

Case name	Input		Expected Result
	Username	Password	
Login success	demo	mode	Show welcome page
Login fail case A	demo	mode2	Show error page
Login fail case B	demo2	mode	Show error page
Login fail case C	demo2	mode2	Show error page

More requirement !!

Password Complexity

Password complexity policies are designed to deter brute force attacks by increasing the number of possible passwords. When password complexity policy is enforced, new passwords must meet the following guidelines:

- The password doesn't contain the account name of the user.
- The password is at least eight characters long.
- The password contains characters from three of the following four categories:
  - Latin uppercase letters (A through Z)
  - Latin lowercase letters (a through z)
  - Base 10 digits (0 through 9)
  - Non-alphanumeric characters such as: exclamation point (!), dollar sign ($), number sign (#), or percent (%).

Passwords can be up to 128 characters long. Use passwords that are as long and complex as possible.

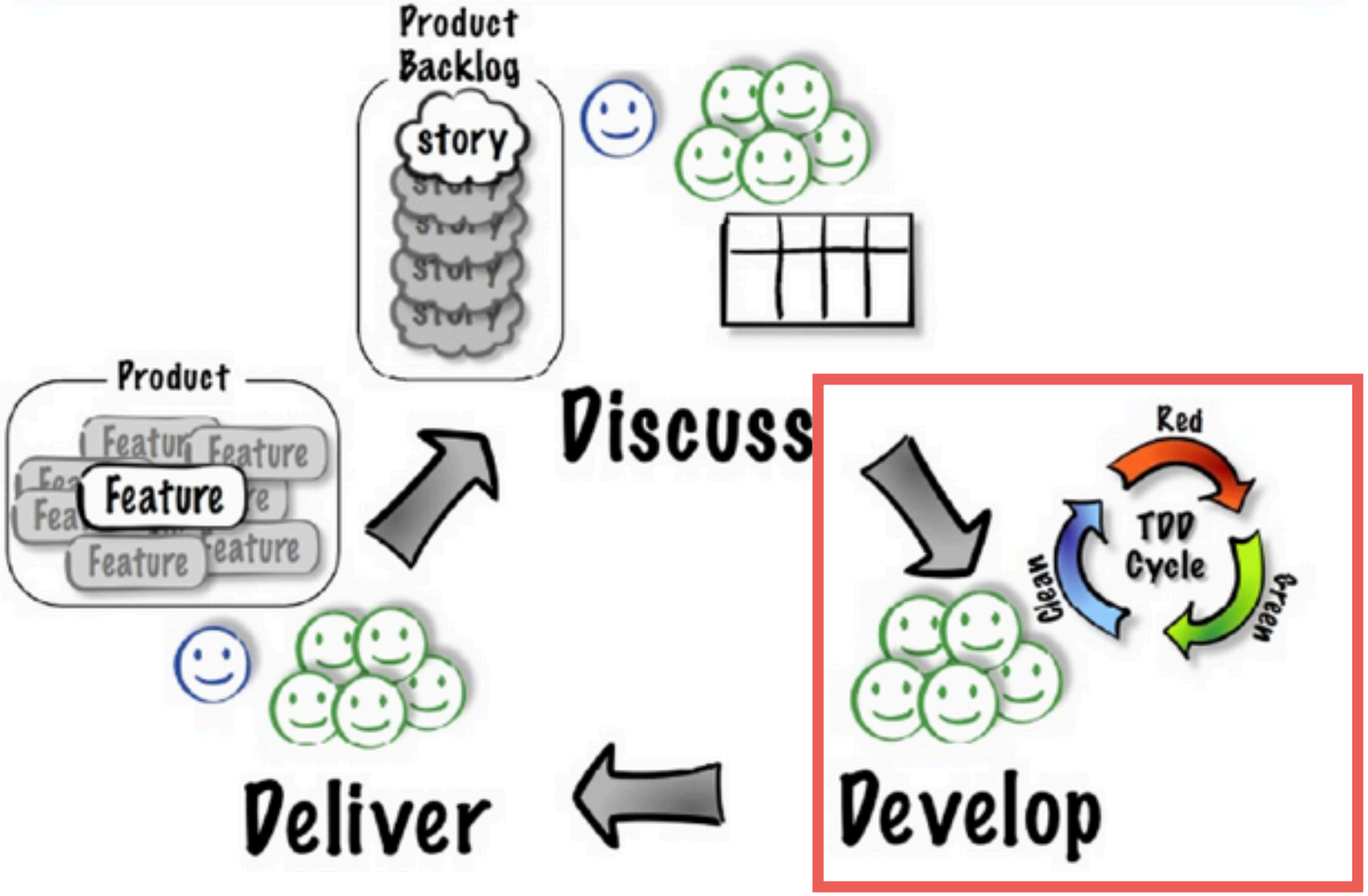
https://learn.microsoft.com/en-us/sql/relational-databases/security/password-policy?view=sql-server-ver16

Test Cases ?

Case name	Input		Expected Result
	Username	Password	
Login success	demo	mode	Show welcome page
Login fail case A	demo	mode2	Show error page
Login fail case B	demo2	mode	Show error page
Login fail case C	demo2	mode2	Show error page

Acceptance Test-Driven Development (ATDD)

Development process = Coding + Testing



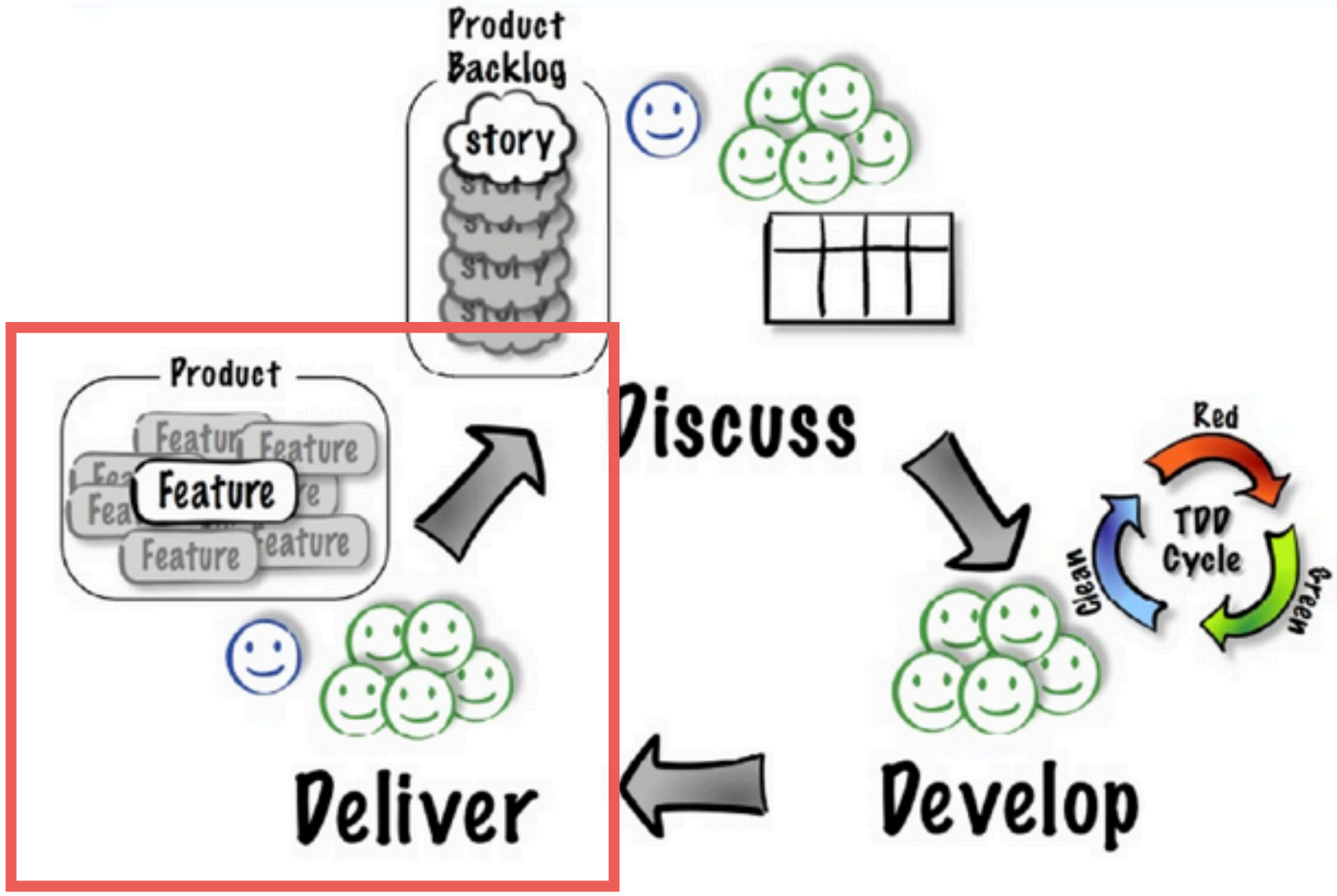
Development process

Implementation with example/data
All examples are passed

Done = Coded + Tested

Acceptance Test-Driven Development (ATDD)

Delivery process

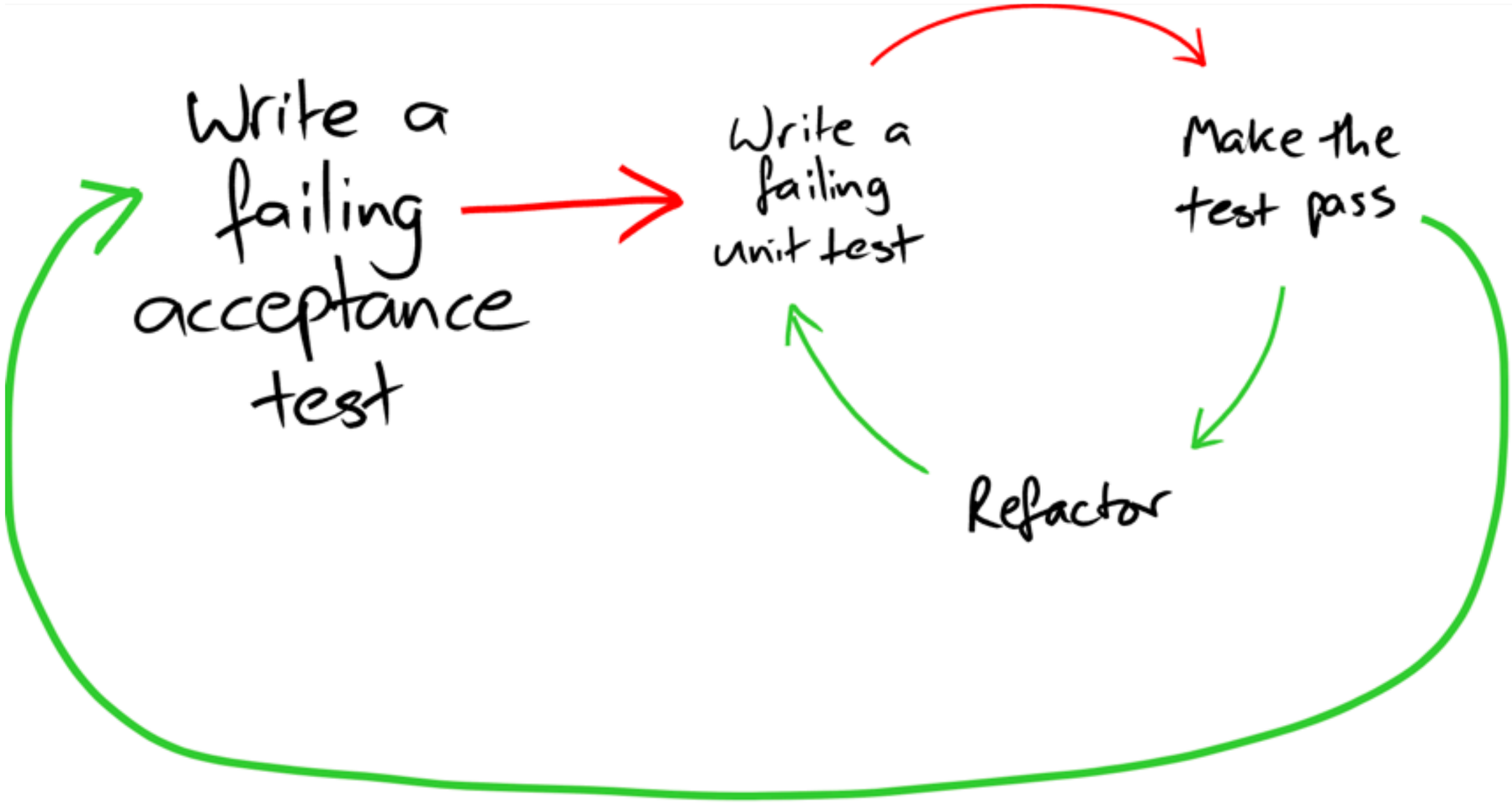


Delivery process

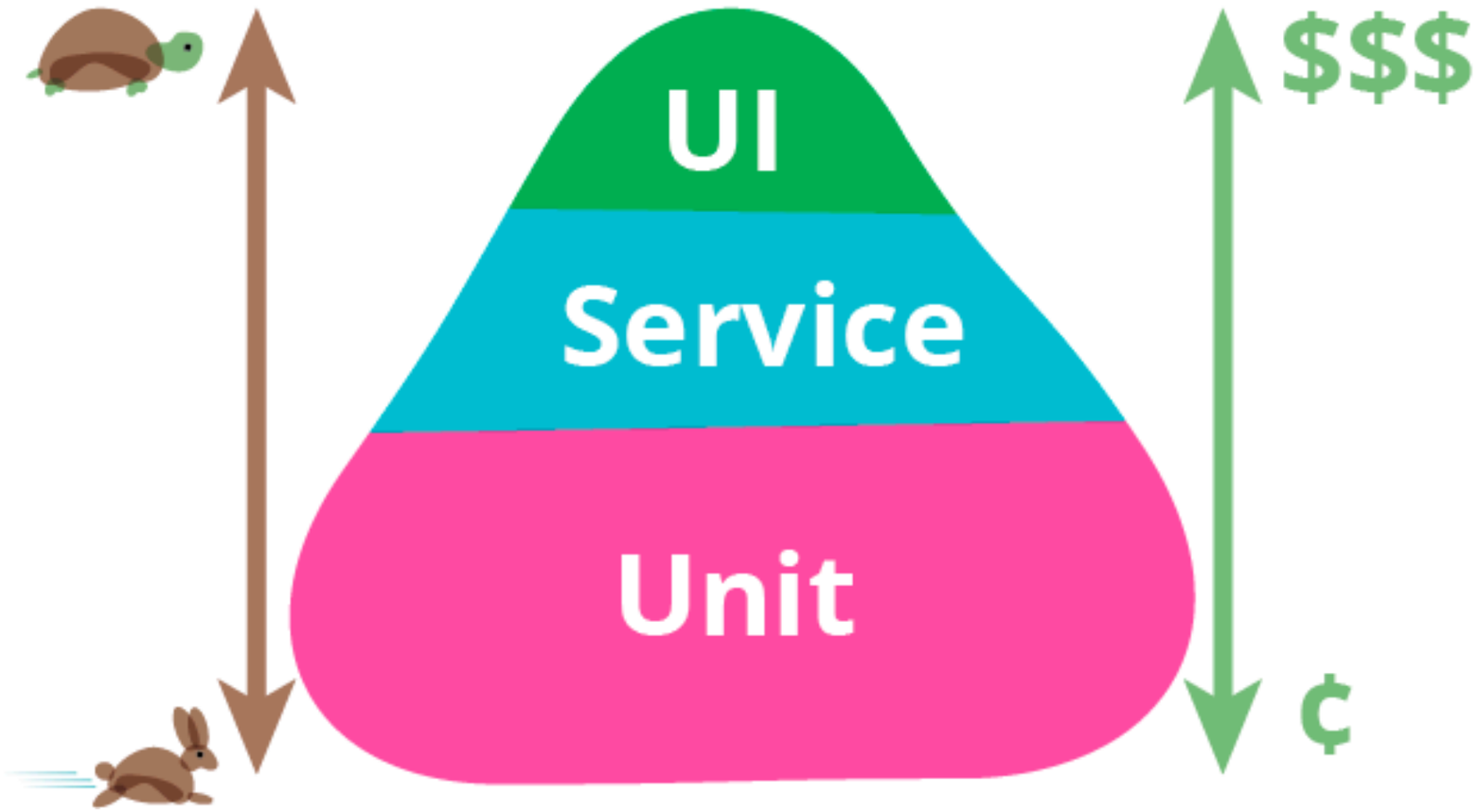
Features are demonstrated to all stakeholders
All examples are passed (new + existed)

Feedback as input to the next discussion

Overall process

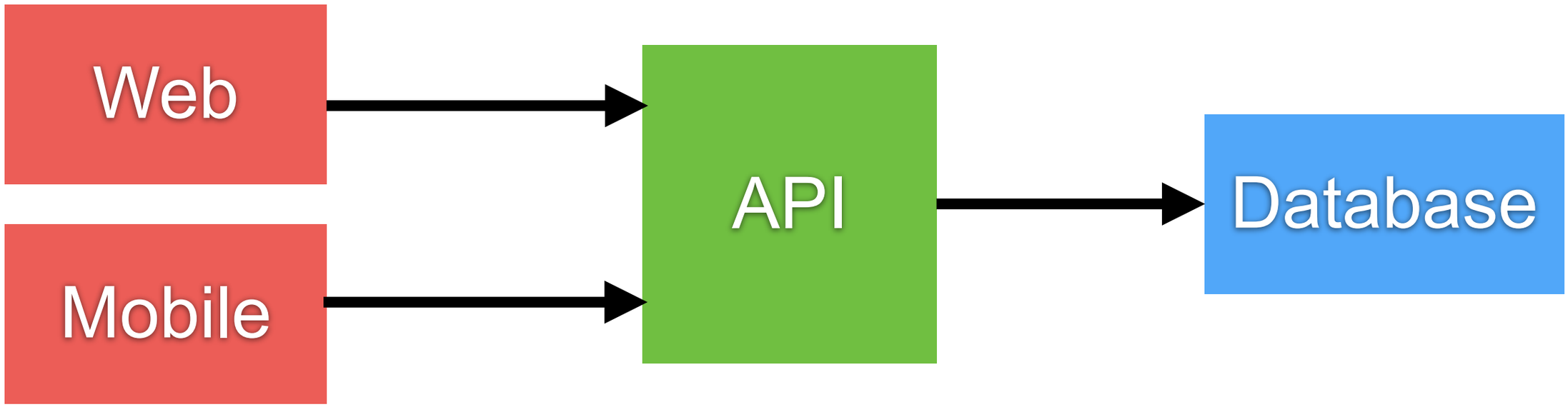


Test Strategies

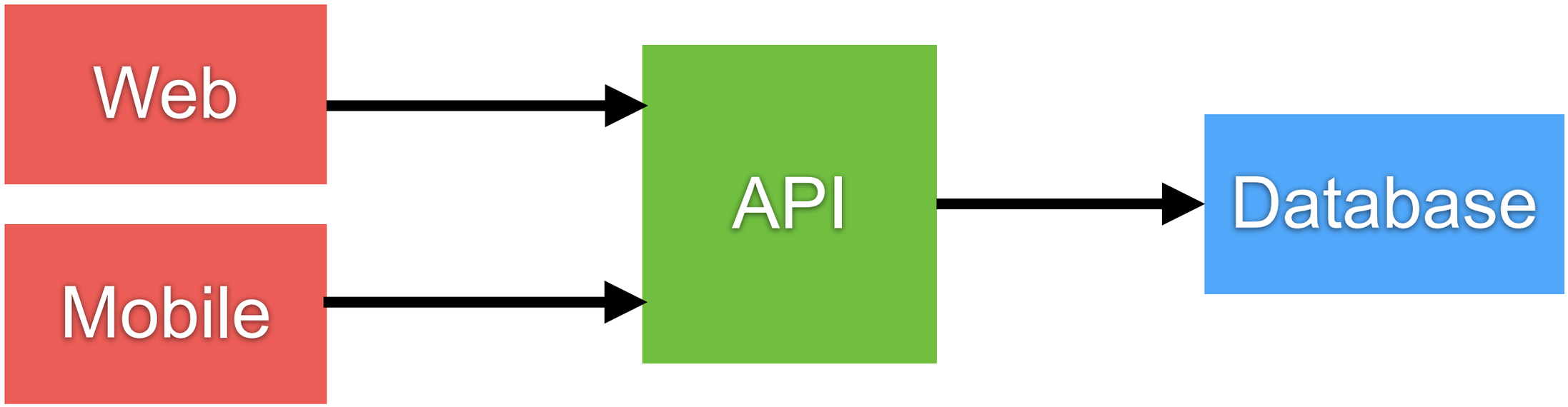


Why -> What -> How to test ?

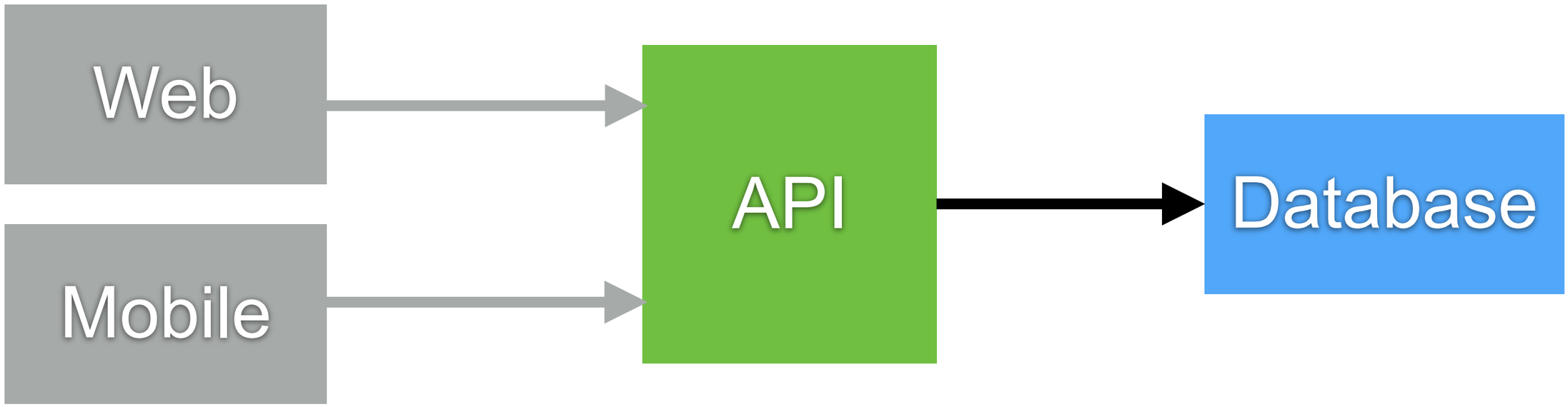
Architecture



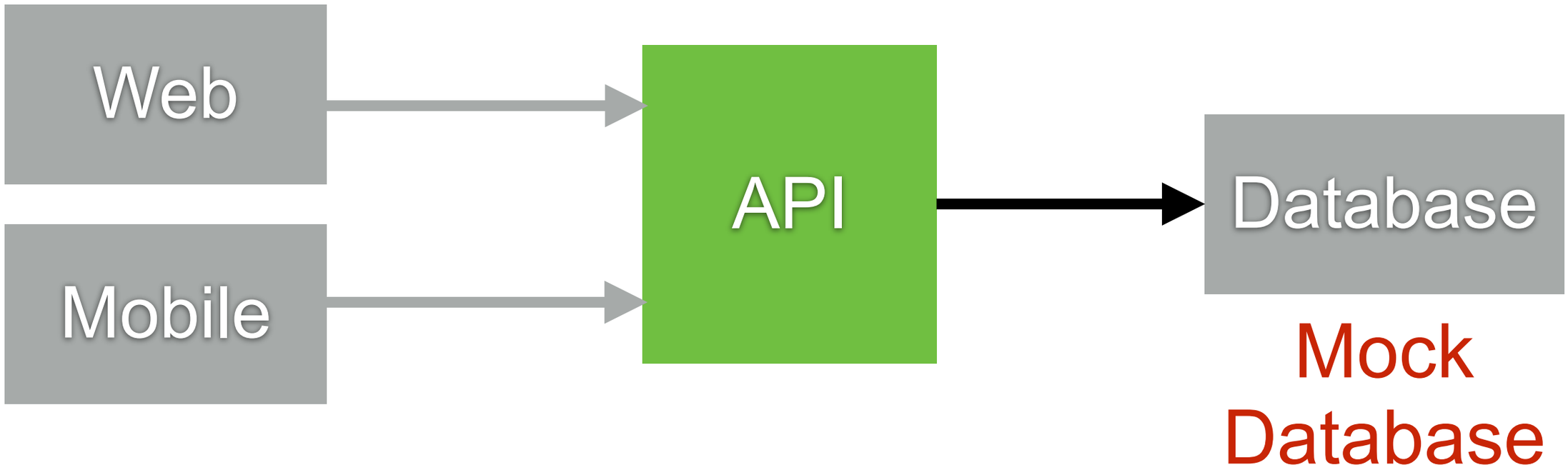
Test ?



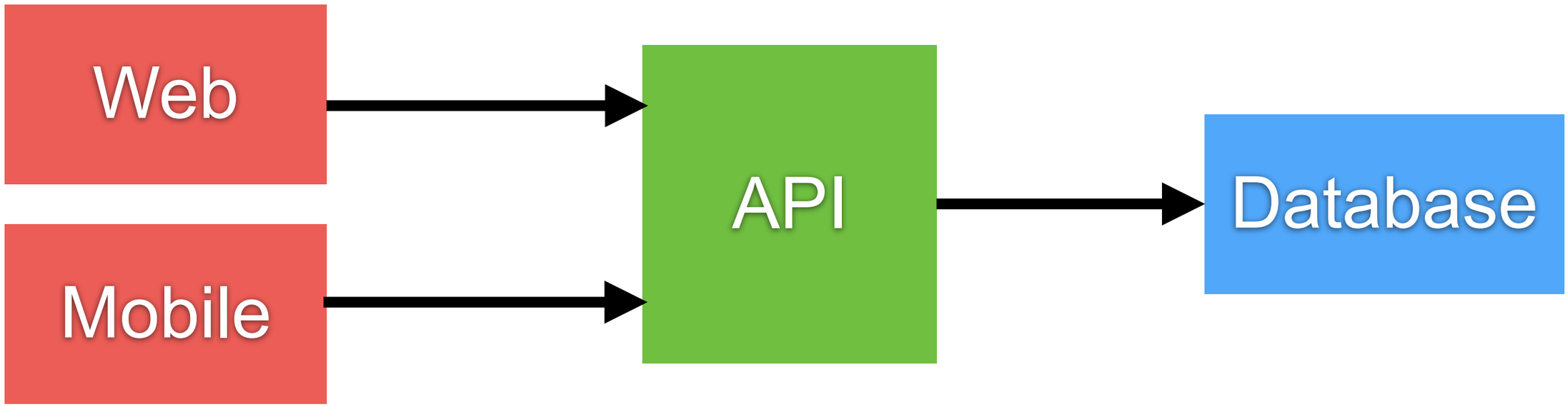
API Testing ?



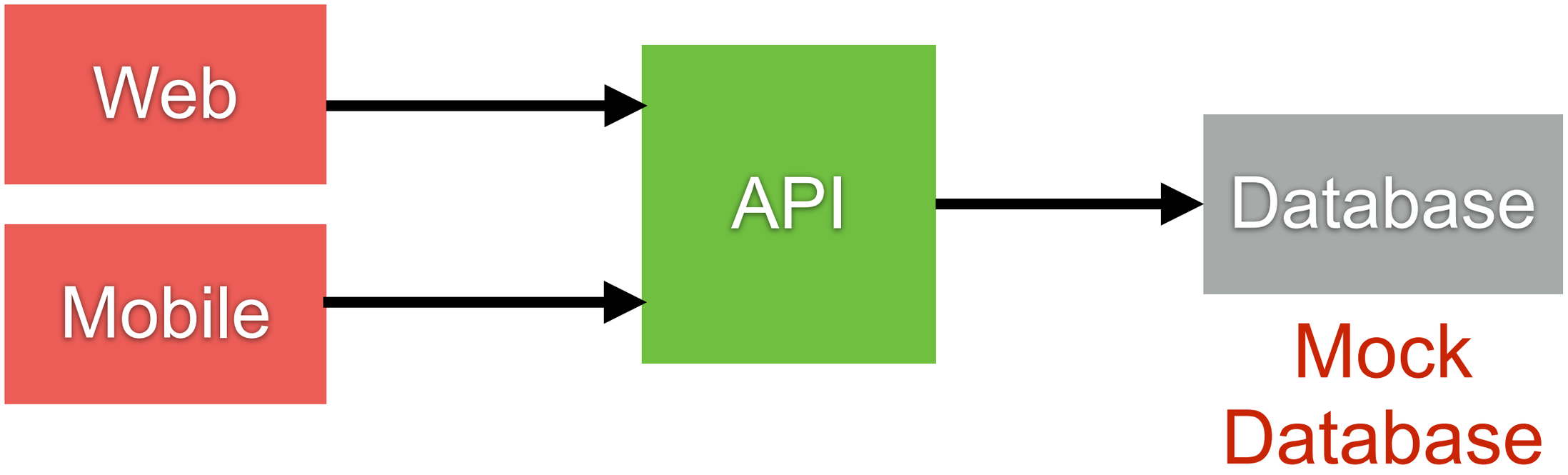
API Testing ?



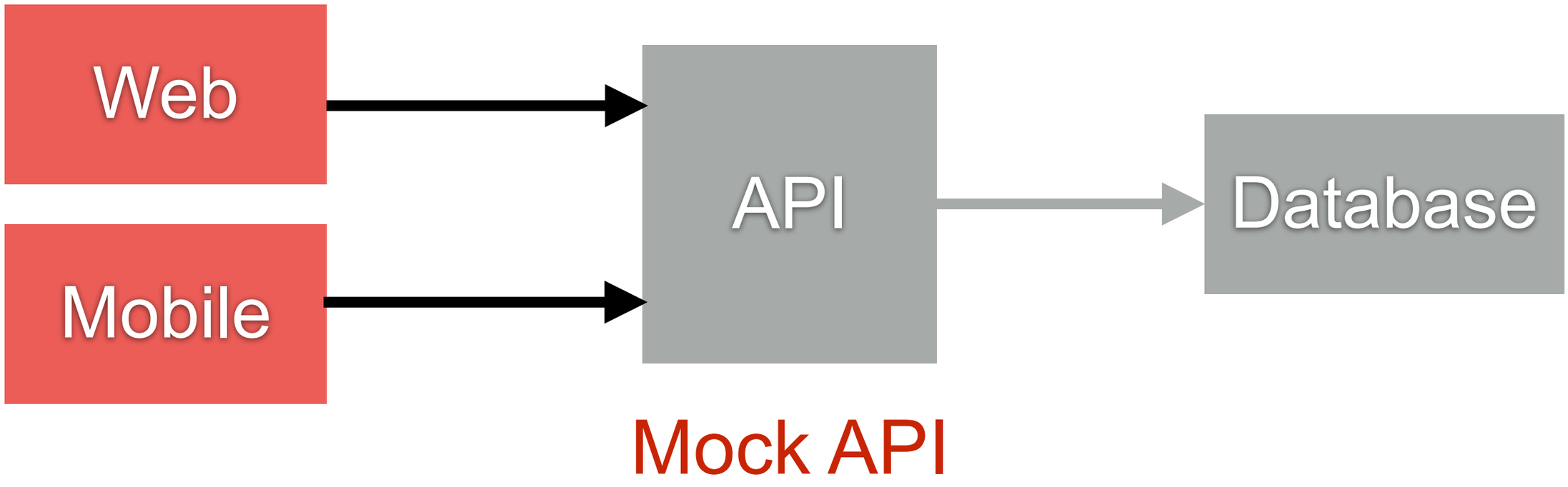
UI Test ?



UI Test ?

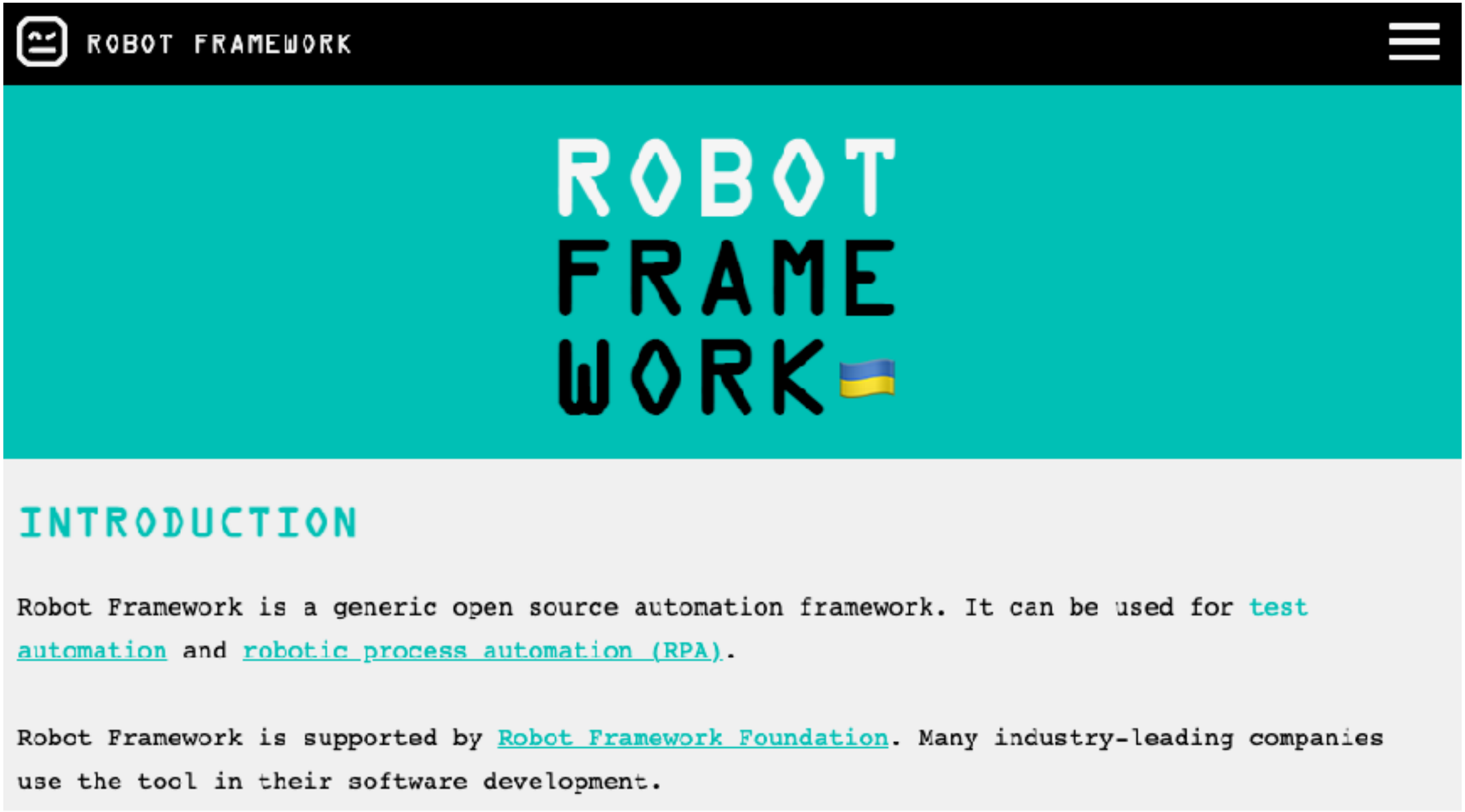


UI Test ?



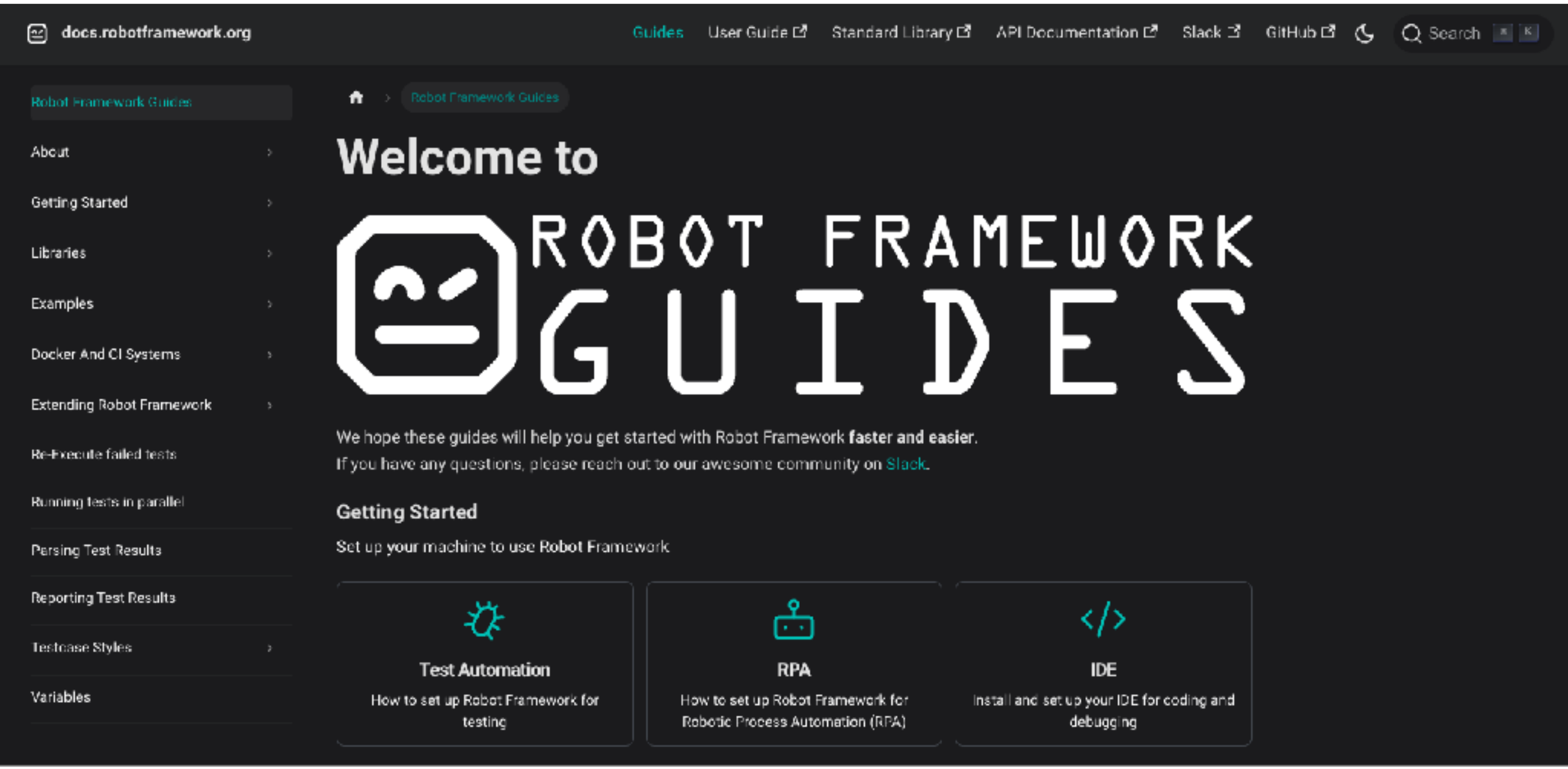
Introduction with Robot Framework

Robot Framework



https://robotframework.org/

Guides for beginner



https://docs.robotframework.org/docs

User Guide

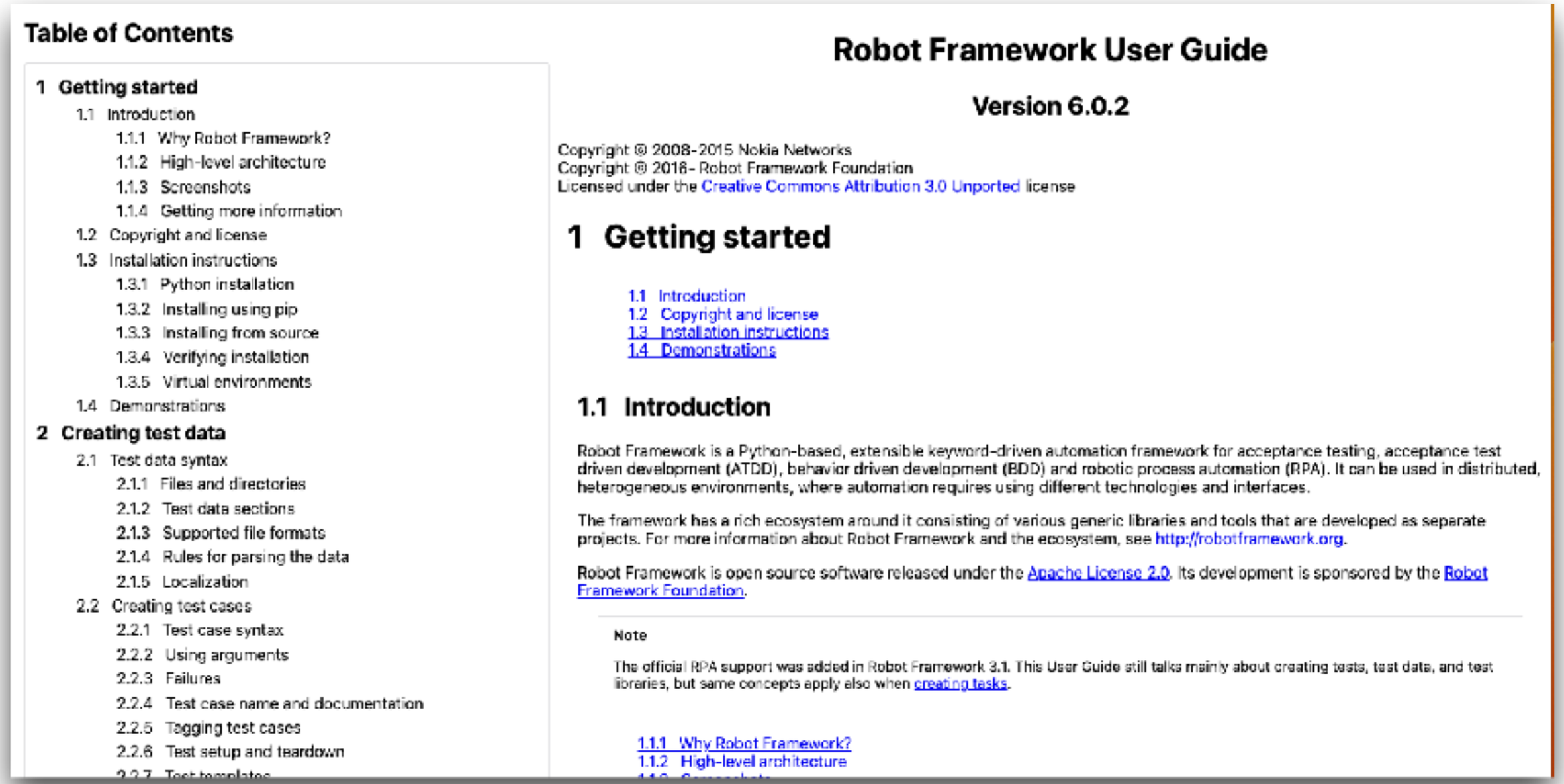


Table of Contents

1 Getting started
- 1.1 Introduction
  - 1.1.1 Why Robot Framework?
  - 1.1.2 High-level architecture
  - 1.1.3 Screenshots
  - 1.1.4 Getting more information
- 1.2 Copyright and license
- 1.3 Installation instructions
  - 1.3.1 Python installation
  - 1.3.2 Installing using pip
  - 1.3.3 Installing from source
  - 1.3.4 Verifying installation
  - 1.3.5 Virtual environments
- 1.4 Demonstrations

2 Creating test data
- 2.1 Test data syntax
  - 2.1.1 Files and directories
  - 2.1.2 Test data sections
  - 2.1.3 Supported file formats
  - 2.1.4 Rules for parsing the data
  - 2.1.5 Localization
- 2.2 Creating test cases
  - 2.2.1 Test case syntax
  - 2.2.2 Using arguments
  - 2.2.3 Failures
  - 2.2.4 Test case name and documentation
  - 2.2.5 Tagging test cases
  - 2.2.6 Test setup and teardown
  - 2.2.7 Test templates

Robot Framework User Guide

Version 6.0.2

Copyright © 2008-2015 Nokia Networks
Copyright © 2016- Robot Framework Foundation
Licensed under the Creative Commons Attribution 3.0 Unported license

1 Getting started

1.1 Introduction
1.2 Copyright and license
1.3 Installation instructions
1.4 Demonstrations

1.1 Introduction

Robot Framework is a Python-based, extensible keyword-driven automation framework for acceptance testing, acceptance test driven development (ATDD), behavior driven development (BDD) and robotic process automation (RPA). It can be used in distributed, heterogeneous environments, where automation requires using different technologies and interfaces.

The framework has a rich ecosystem around it consisting of various generic libraries and tools that are developed as separate projects. For more information about Robot Framework and the ecosystem, see http://robotframework.org.

Robot Framework is open source software released under the Apache License 2.0. Its development is sponsored by the Robot Framework Foundation.

Note

The official RPA support was added in Robot Framework 3.1. This User Guide still talks mainly about creating tests, test data, and test libraries, but same concepts apply also when creating tasks.

1.1.1 Why Robot Framework?
1.1.2 High-level architecture

https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html

Build-in Library

LIBRARIES | BUILT-IN | TOOLS

Libraries and tools that are bundled with the framework. Libraries provide the actual automation and testing capabilities to Robot Framework by providing keywords.

Filter by tag

Name	Description	Tags
Builtin	Provides a set of often needed generic keywords. Always automatically available without imports.	LIBRARY
Collections	Provides a set of keywords for handling Python lists and dictionaries.	LIBRARY
DateTime	Library for date and time conversions.	LIBRARY
Dialogs	Provides means for pausing the execution and getting input from users.	LIBRARY
Libdoc	Generate keyword documentation for test libraries and resource files.	TCOL

https://robotframework.org/?tab=builtin#resources

3-parties Library

LIBRARIES | BUILT-IN | TOOLS

Separately developed external libraries that can be installed based on your needs. Creating your own libraries is a breeze. For instructions, see creating test libraries in Robot Framework User Guide.

Filter by tag

Name	Description	Stars	Tags
SeleniumLibrary	Web testing library that uses popular Selenium tool internally.	1233	WEB, SELENIUM
RPA framework	Collection of open-source libraries and tools for Robotic Process Automation (RPA), designed to be used both with Robot Framework and Python.	805	RPA
HTTP RequestsLibrary (Python)	HTTP level testing using Python Requests internally.	444	HTTP
Browser Library	A modern web testing library powered by Playwright. Aiming for speed, reliability and visibility.	378	WEB
AppiumLibrary	Android and iOS testing. Uses Appium internally.	345	MOBILE

https://robotframework.org/?tab=libraries#resources

Software Requirement

Software Requirement

python™

3+

Visual Studio Code

Check

$pip -V

$pip install robotframework

Check

$pip list

```
redis                                 4.5.3
requests                              2.29.0
robotframework                        6.1a1
robotframework-appiumlibrary          2.0.0
robotframework-pythonlibcore          4.1.2
robotframework-seleniumlibrary        6.0.0
selenium                              4.8.2
setuptools                            58.0.4
```

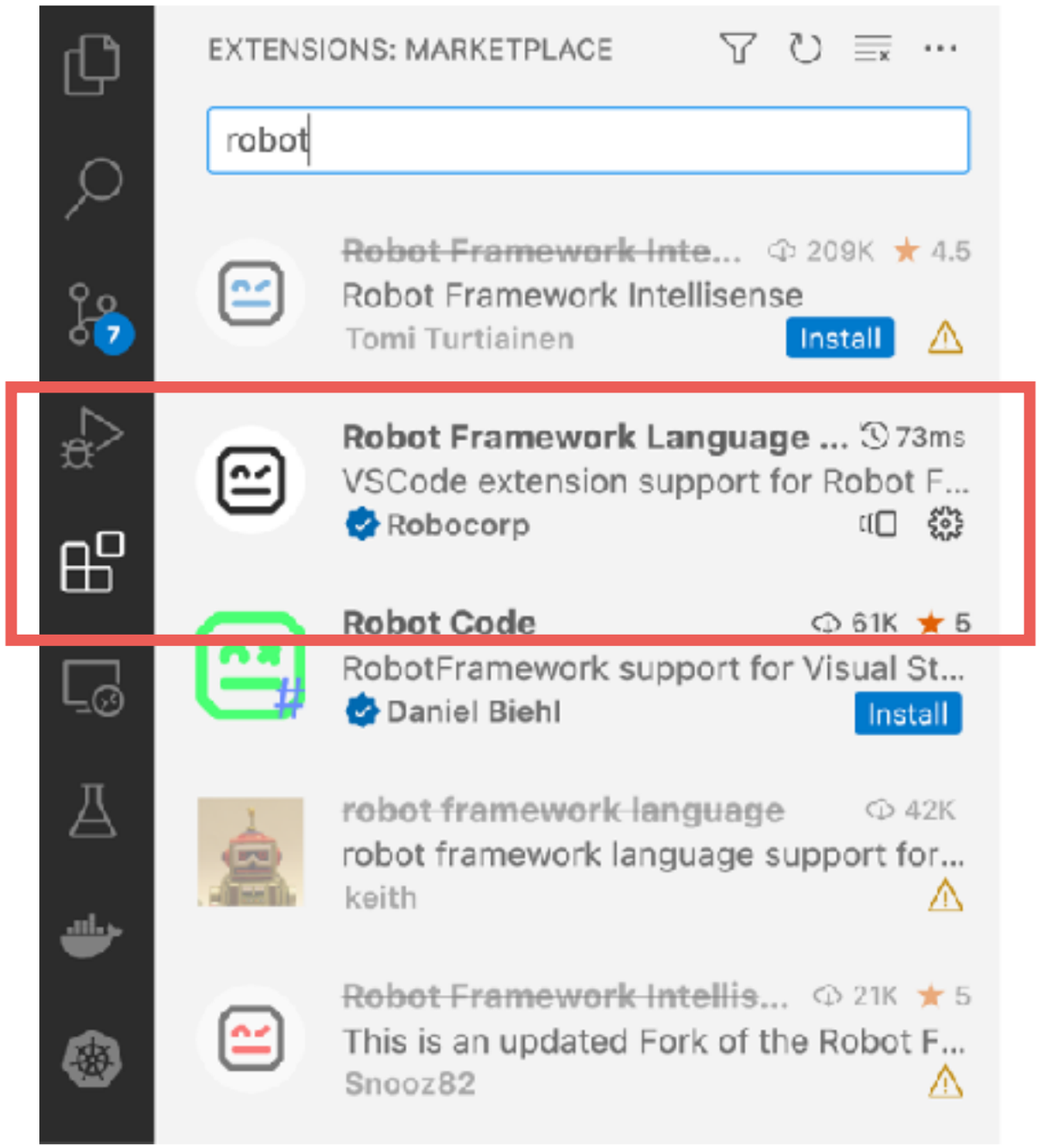
Check Robot Framework

$robot

```
[ ERROR ] Expected at least 1 argument, got 0.

Try --help for usage information.
```

VS Code Extension



Let’ start

Write First Test Case

Test Structure

hello.robot

```
*** Settings ***

*** Variables ***

*** Test Cases ***

*** Keywords ***
```

Sections

Section Name	Used for
Settings	Import libraries, resource files and variable files Define metadata for test suites and test cases
Variables	Define variables
Test Cases	Create test cases
Tasks	Create tasks
Keywords	Create user keywords

https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#test-data-sections

Test Cases

Case name	Input		Expected Result
	Username	Password	
Login success	demo	mode	Show welcome page
Login fail case A	demo	mode2	Show error page
Login fail case B	demo2	mode	Show error page
Login fail case C	demo2	mode2	Show error page

Run test

$robot hello.robot

UI testing with Robot Framework

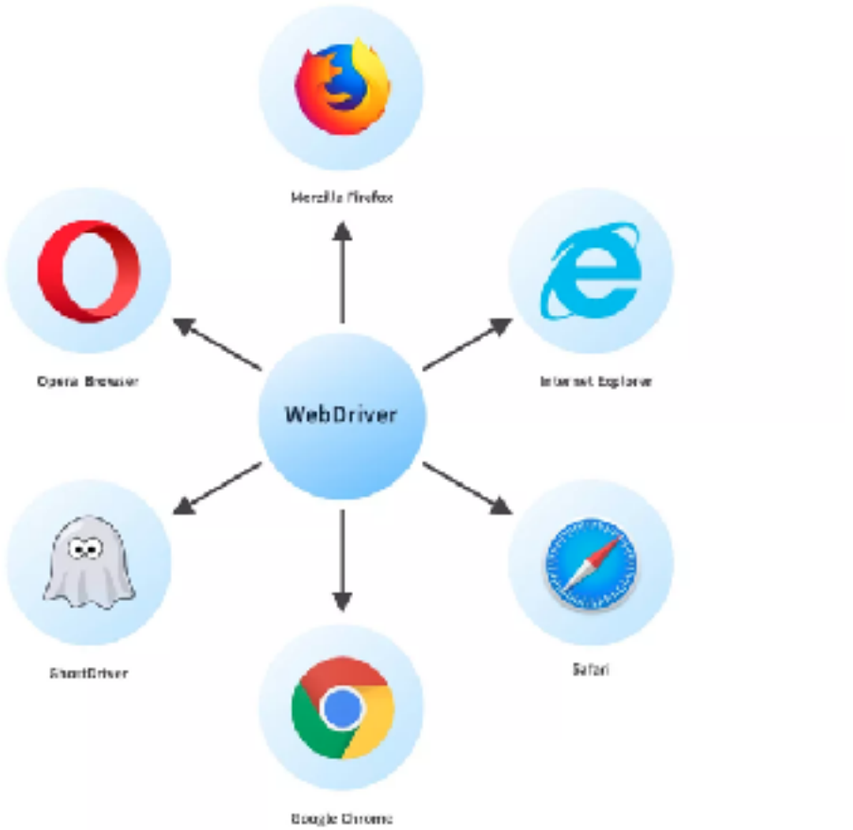
UI Testing (Web)

Selenium Library
Browser Library

https://robotframework.org/?tab=libraries#resources

Selenium Library

Use selenium project
Support multiple web browsers

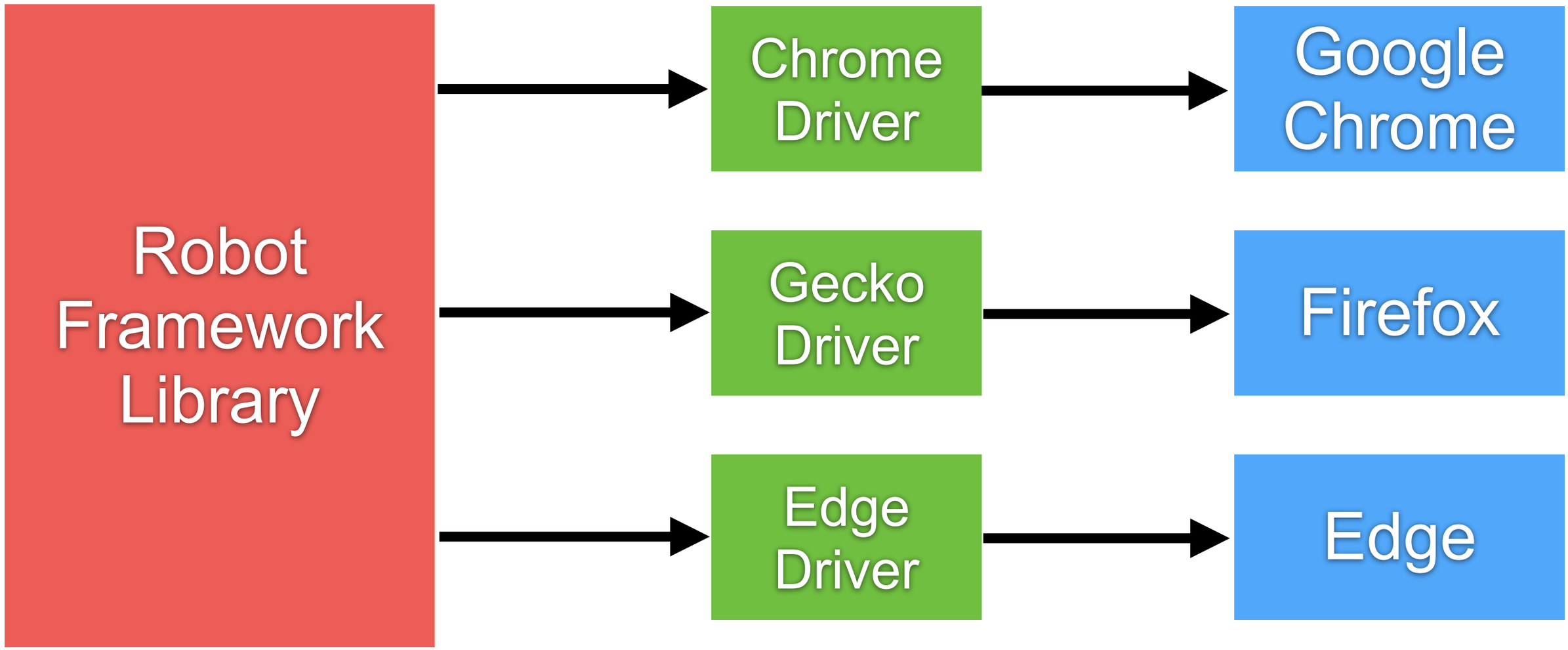


https://github.com/robotframework/SeleniumLibrary

Installation

$pip install -U robotframework-seleniumlibrary
$pip list

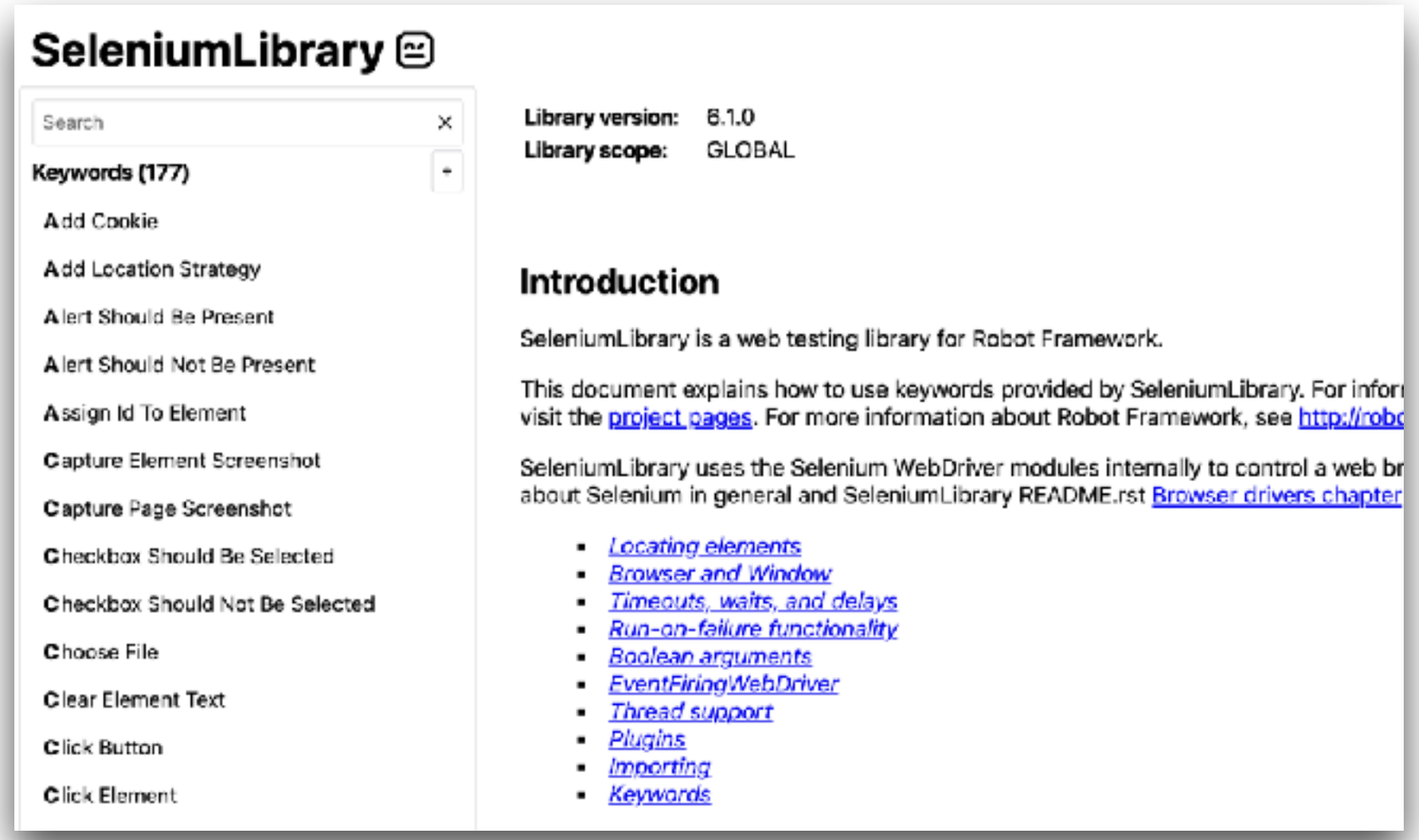
Selenium Library



https://github.com/robotframework/SeleniumLibrary

Selenium Library

List of keywords



https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html

Workshop with First test case

$robot hello.robot

Web Element Locators !!

How to access ?

Web Element Location

Identifier
id
name
XPath
CSS Selector
Class name

Web Element Location

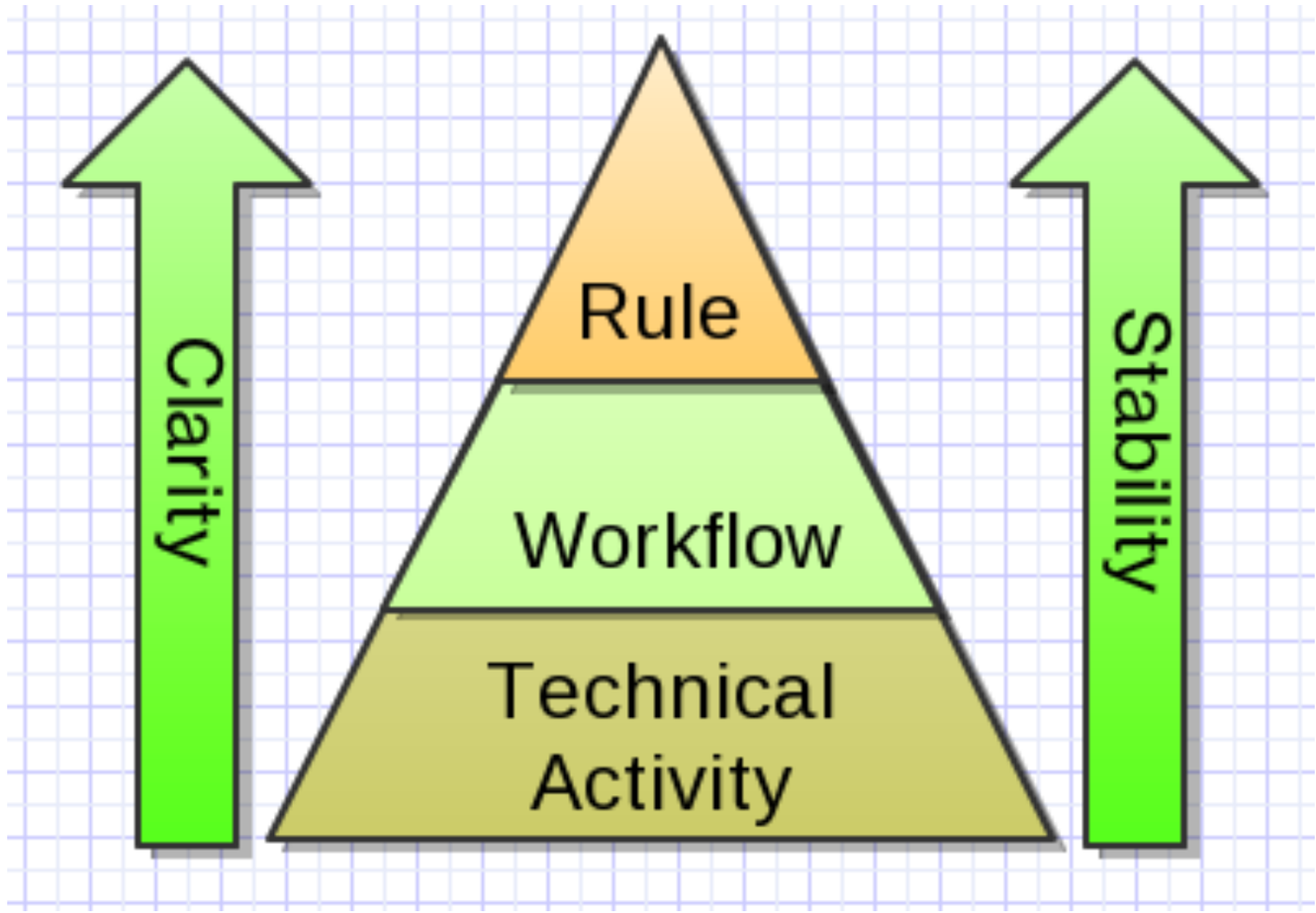
Strategy	Match based on	Example
id	Element `id`.	`id:example`
name	`name` attribute.	`name:example`
identifier	Either `id` or `name`.	`identifier:example`
class	Element `class`.	`class:example`
tag	Tag name.	`tag:div`
xpath	XPath expression.	`xpath://div[@id="example"]`
css	CSS selector.	`css:div#example`
dom	DOM expression.	`dom:document.images[5]`
link	Exact text a link has.	`link:The example`
partial link	Partial link text.	`partial link:he ex`
sizzle	Sizzle selector deprecated.	`sizzle:div.example`
data	Element `data-*` attribute	`data:id:my_id`
jquery	jQuery expression.	`jquery:div.example`
default	Keyword specific default behavior.	`default:example`

https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html

Bad locator === Flaky test

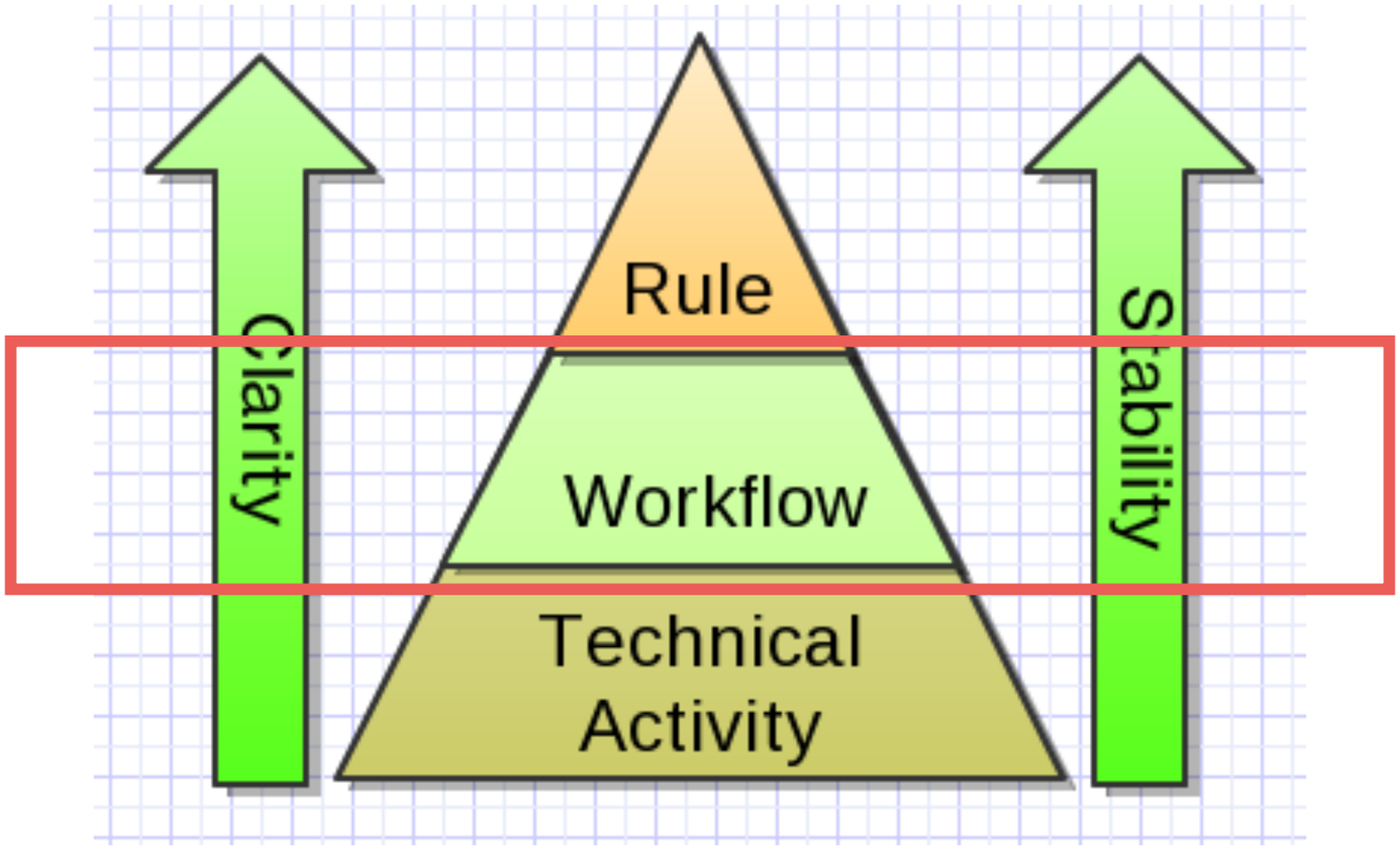
https://testing.googleblog.com/2016/05/flaky-tests-at-google-and-how-we.html

3 levels of UI Automation Test



https://gojko.net/2010/04/13/how-to-implement-ui-testing-without-shooting-yourself-in-the-foot-2/

User Interface workflow



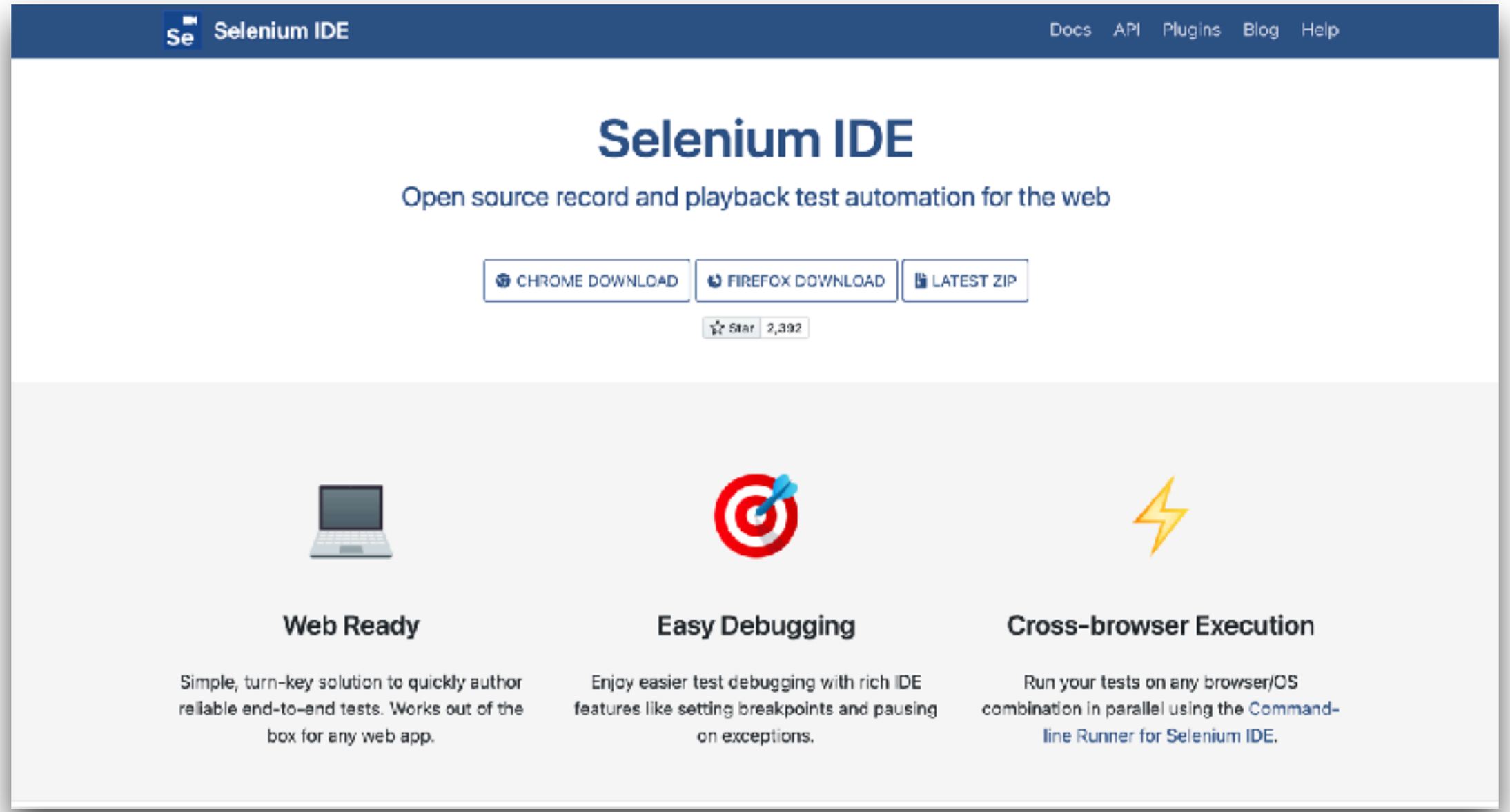
https://gojko.net/2010/04/13/how-to-implement-ui-testing-without-shooting-yourself-in-the-foot-2/

Recording Tools

Recording Tools

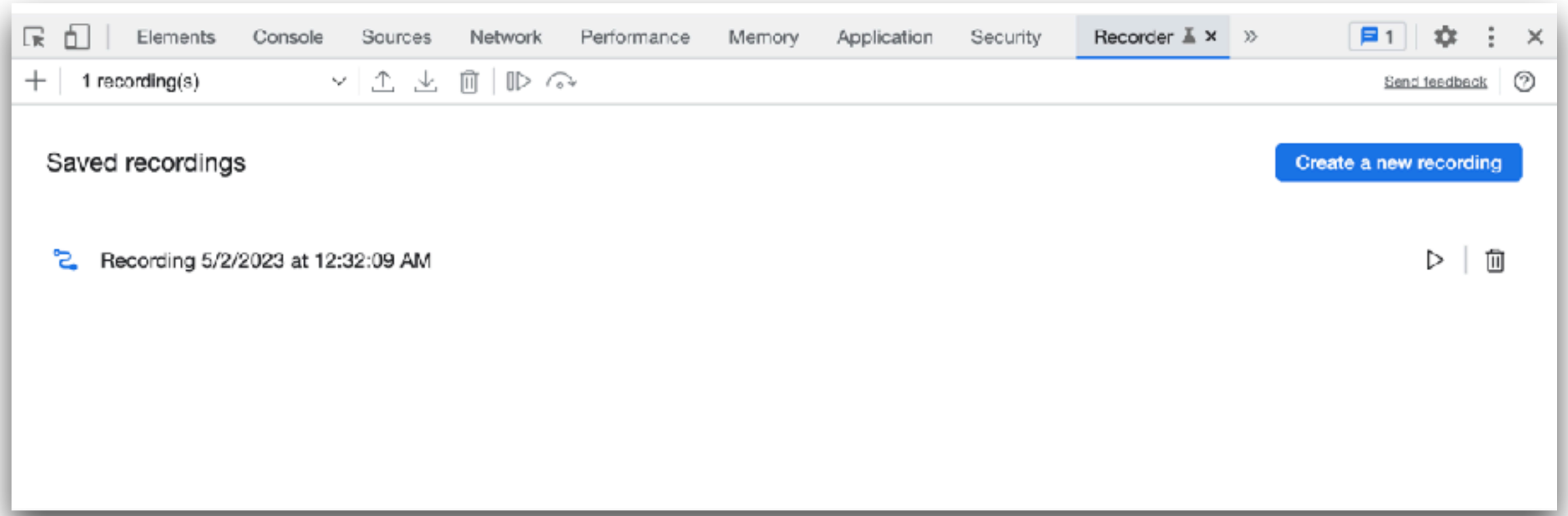
Selenium IDE
Google Chrome Recorder

Selenium IDE



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

Google Chrome Recorder



https://developer.chrome.com/docs/devtools/recorder/

Arguments in Keyword

https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#creating-user-keywords

Arguments in Keyword

Keyword arguments
Embedding arguments in keyword

https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#creating-user-keywords

Keyword arguments

```
*** Settings ***
Library    SeleniumLibrary

*** Test Cases ***
Login fail case with wrong username
    Fill in    demo    mode

*** Keywords ***
Fill in
    [Arguments]    ${username}    ${password}
    Input Text    id=username_field    ${username}
    Input Text    id=password_field    ${password}
```

Embedding arguments

```
*** Settings ***
Library    SeleniumLibrary

*** Test Cases ***
Login fail case with wrong username
    Fill in with user='demo' and password='mode'

*** Keywords ***
Fill in with user='${username}' and password='${password}'
    Input Text    id=username_field    ${username}
    Input Text    id=password_field    ${password}
```

Grouping test case with Tag

https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#tagging-test-cases

Use Tag in test case

```
*** Test Cases ***
Login fail case with wrong username
    [Tags]  feature01   done
    Fill in with user='demo2' and password='mode'

Login fail case with wrong password
    [Tags]  feature01   done
    Fill in with user='demo' and password='mode2'

Login fail case with wrong username and password
    [Tags]  feature01   testing
    Fill in with user='demo2' and password='mode2'
```

$robot -i testing hello.robot

$robot -e testing hello.robot

Using Variables

https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#variables

Variables

Scalar => ${var}

List => @{var}

Dictionary => &{var}

Environment variable => %{ENV_VAR}

Build-in Variables

${SPACE}
${SPACE*5}

${EMPTY}

https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#built-in-variables

Use variables from command line

Try to change value in test case ?

```
*** Keywords ***
Open Login Page
    Open Browser  https://demo-login-workshop.vercel.app/
    ...  browser=chrome
```

Change URL and browser ?

https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#built-in-variables

Use variables from command line

Create new variables

```
*** Variables ***
${URL}    https://demo-login-workshop.vercel.app/
${BROWSER}    chrome

*** Keywords ***
Open Login Page
    Open Browser  ${URL}
    ...  browser=${BROWSER}
```

Use variables from command line

Change data from command line

$robot -v URL:**new** -v BROWSER:**new**

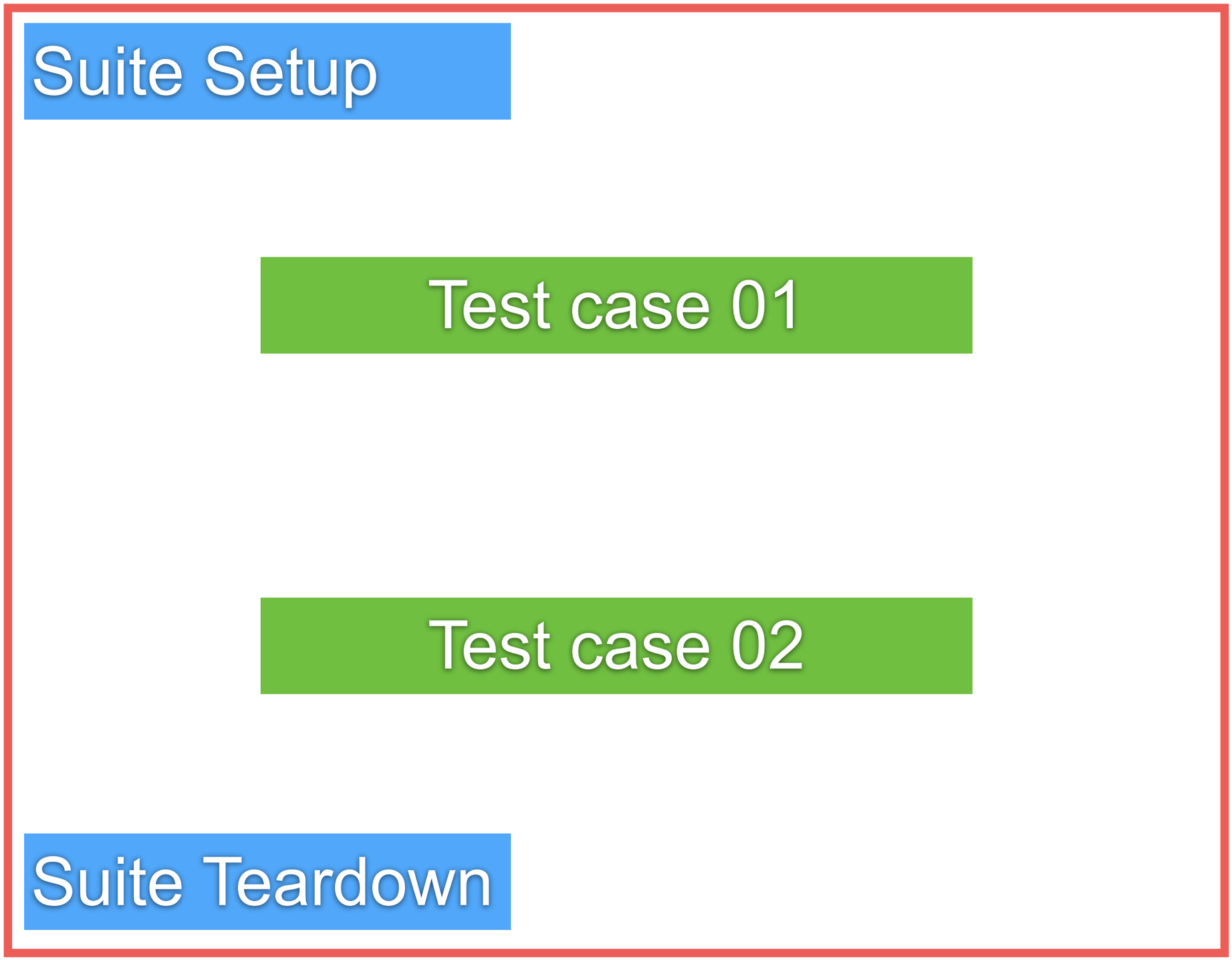
Test Life Cycle

Testing life cycle

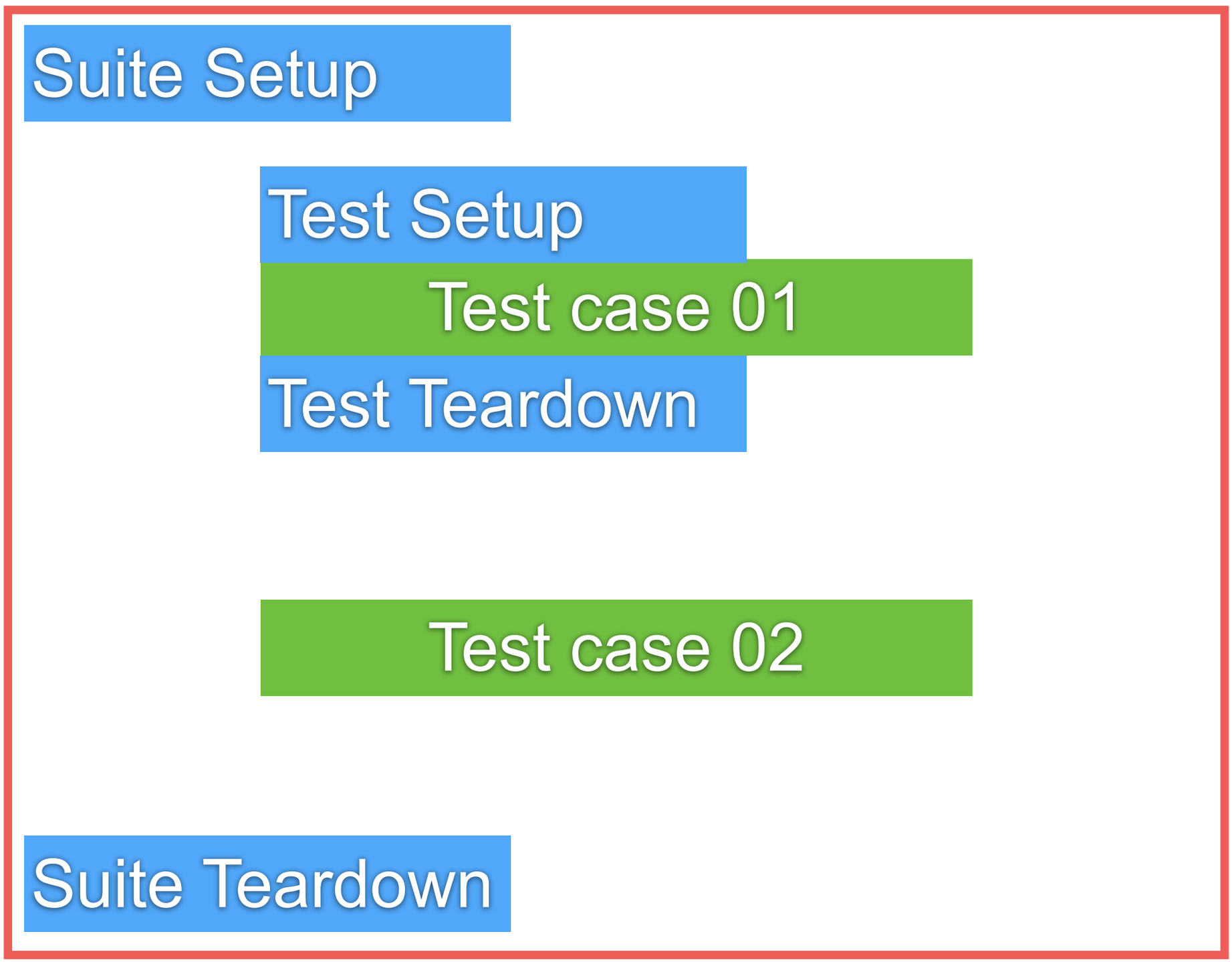
Test case 01

Test case 02

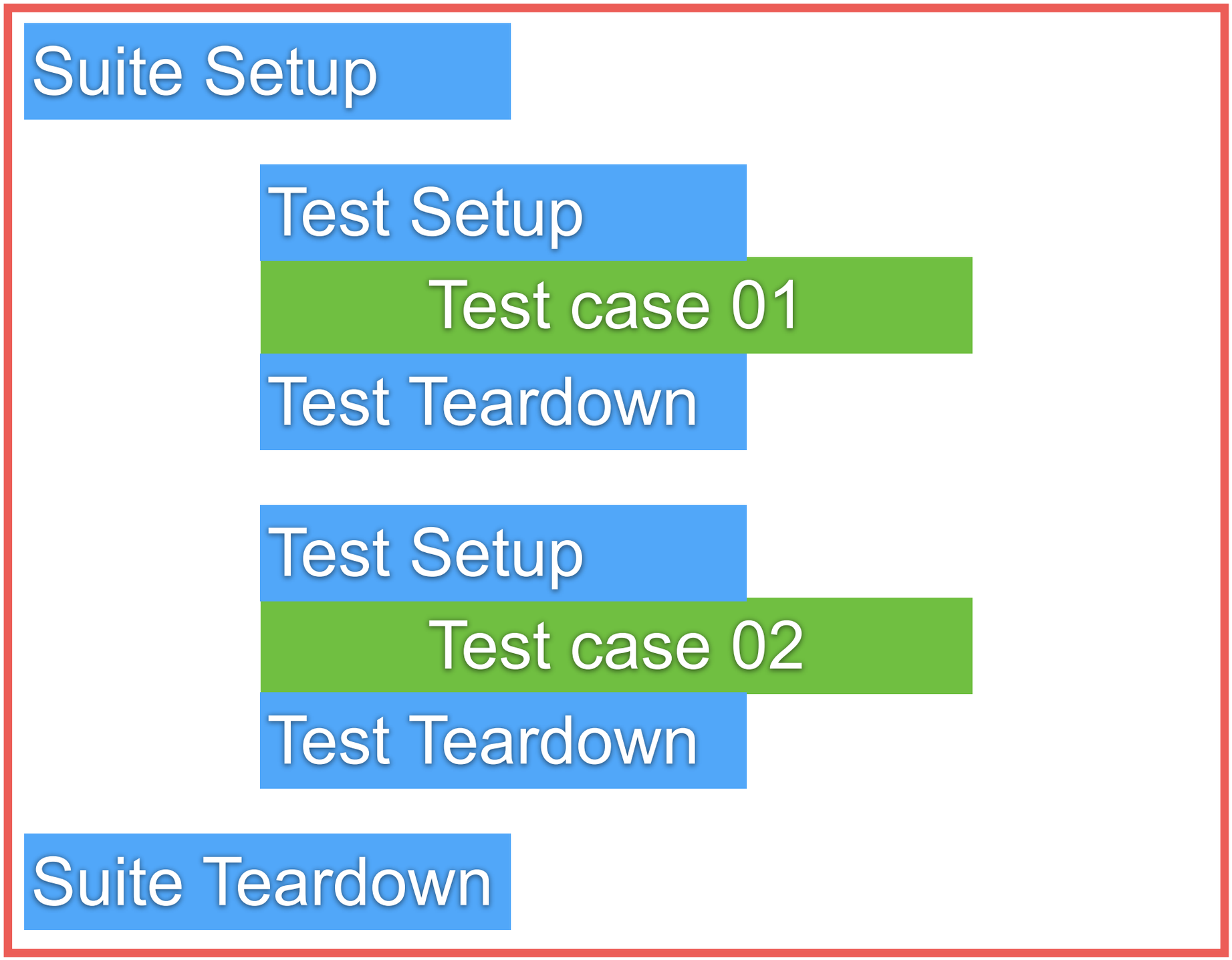
Testing life cycle



Testing life cycle



Testing life cycle



Workshop with Test life cycle

Test Template

Test template

Template per suite
Template per test case

https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#test-templates

Template per suite

```
*** Settings ***
Test Template     Flow of login success

*** Keywords ***
Flow of login success
    [Arguments]  ${input}
    Should Be Equal    first    ${input}
```

Template per suite

```
*** Settings ***
Test Template     Flow of login success

*** Keywords ***
Flow of login success
    [Arguments]  ${input}
    Should Be Equal    first    ${input}

*** Test Cases ***
Case 01     data 1
Case 02     data 2
Case 03     data 3
```

Template per test case

```
*** Test Cases ***
Success case
    [Template]    Flow of login success
    data 1
    data 2
    data 3

*** Keywords ***
Flow of login success
    [Arguments]  ${input}
    Should Be Equal    first    ${input}
```

Workshop test template

Working with Resource files

https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#resource-files

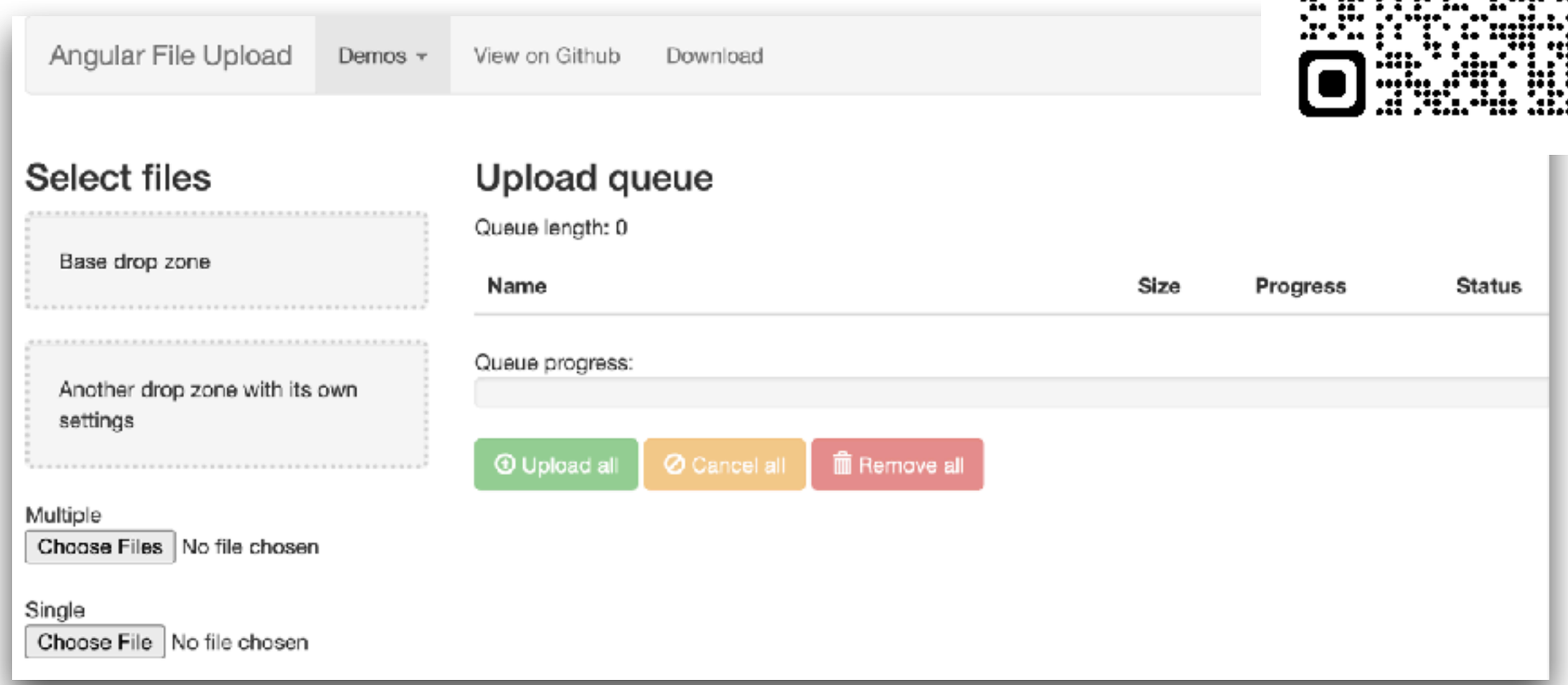
Resource files

Reuse data in settings, variables and keywords
Extension with **.resource** or **.robot**

```
*** Settings ***
Resource  my_keyword.resource
Resource  my_keyword.robot
```

https://robotframework.org/robotframework/latest/RobotFrameworkUserGuide.html#test-templates

More Workshop



http://nervgh.github.io/pages/angular-file-upload/examples/simple/

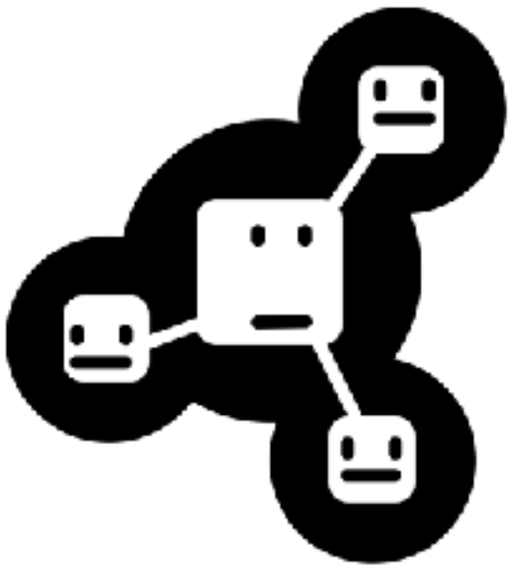
Scaling Testing

Scaling Testing

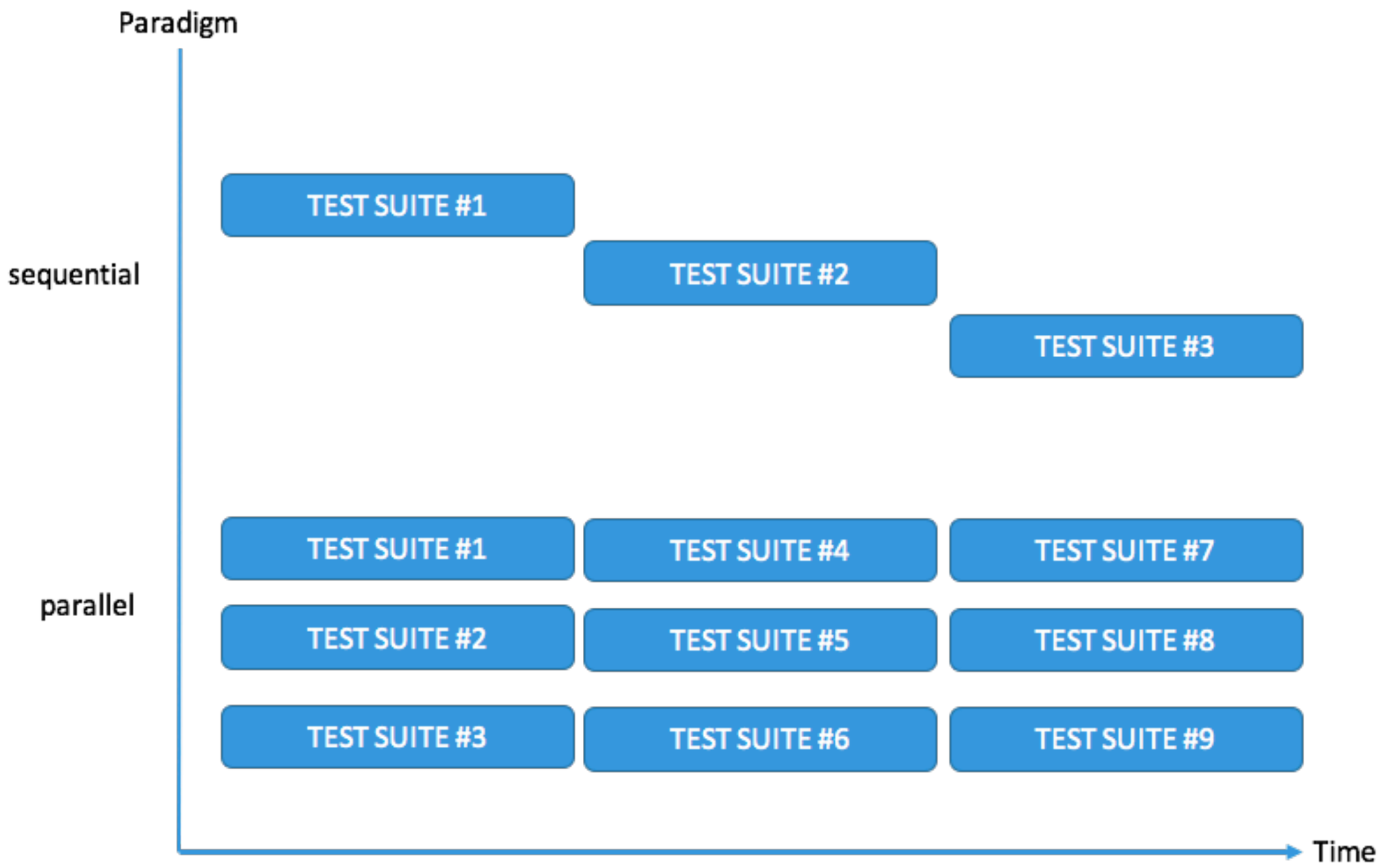
- Pabot
- Selenium Grid

Pabot

Parallel executor for Robot Framework
Split one execution into multiple
https://pabot.org/



Test execution



Using Pabot

$pip install -U robotframework-pabot
$pabot

Parallel test suites

$pabot flow_dress_sorting.robot

Parallel test cases

$pabot **--testlevelsplit**
flow_dress_sorting.robot

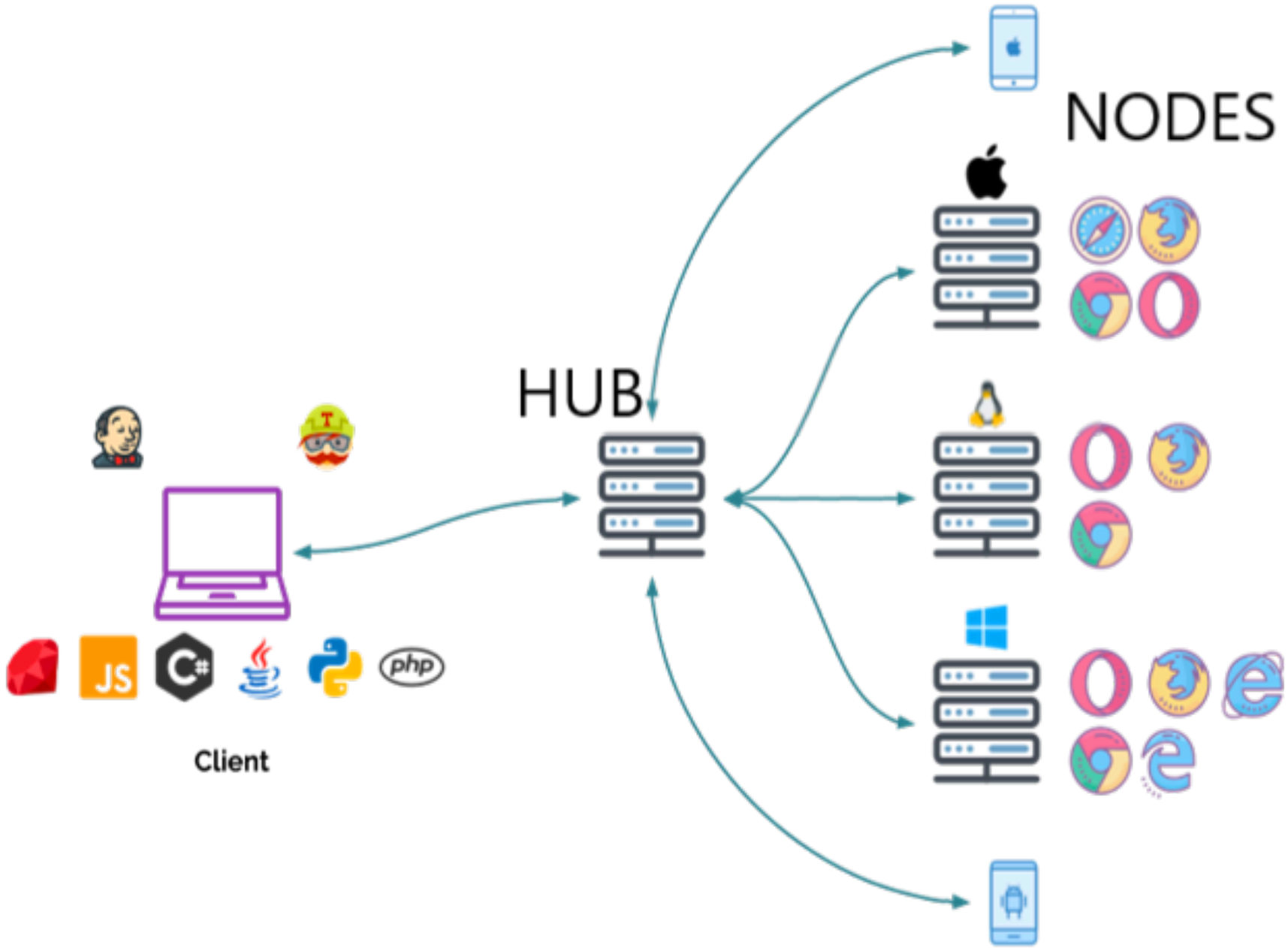
Workshop with Pabot

Selenium grid

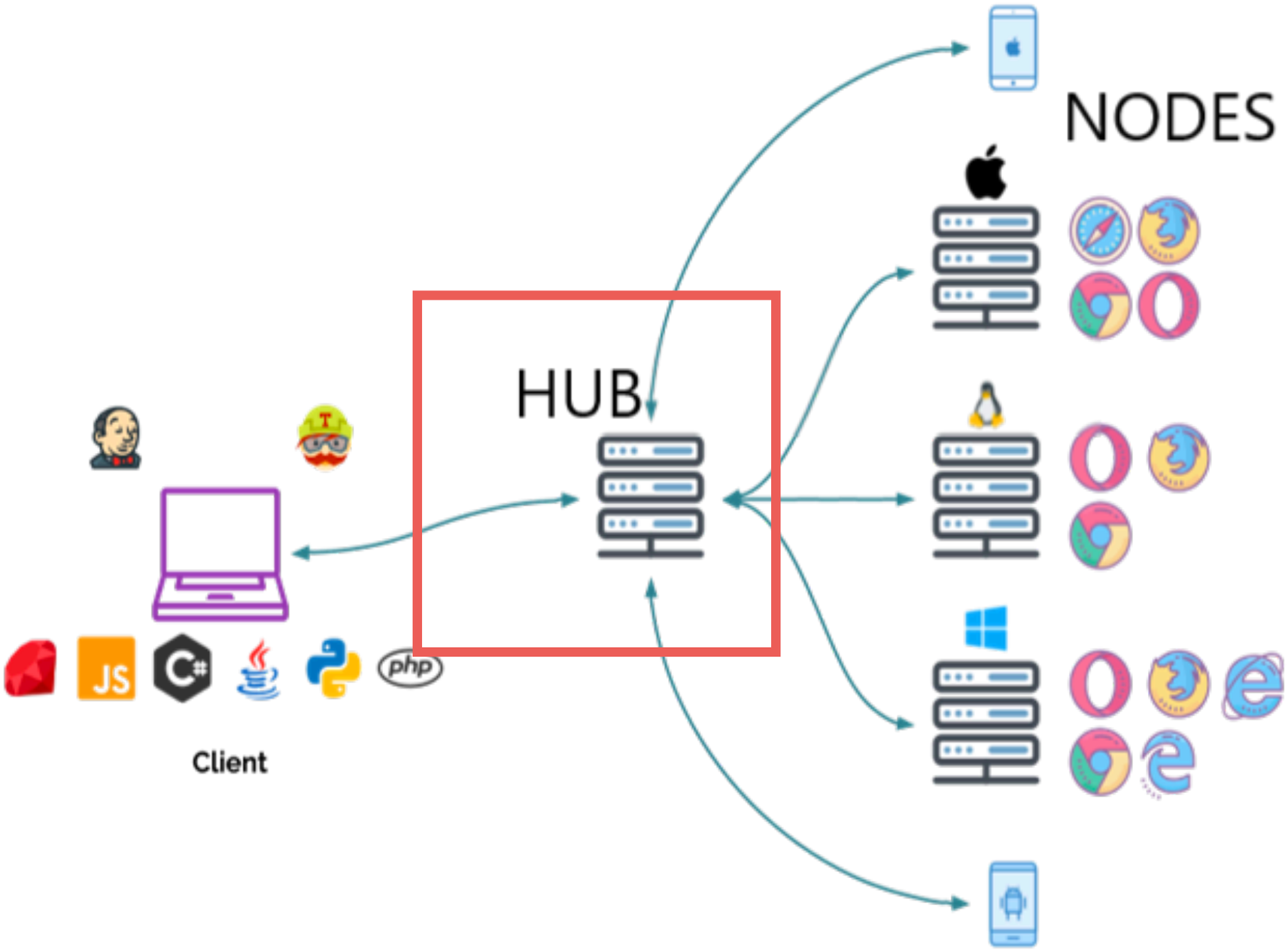
Run test cases on different machines
Parallel testing

https://github.com/SeleniumHQ/selenium/wiki/Grid2

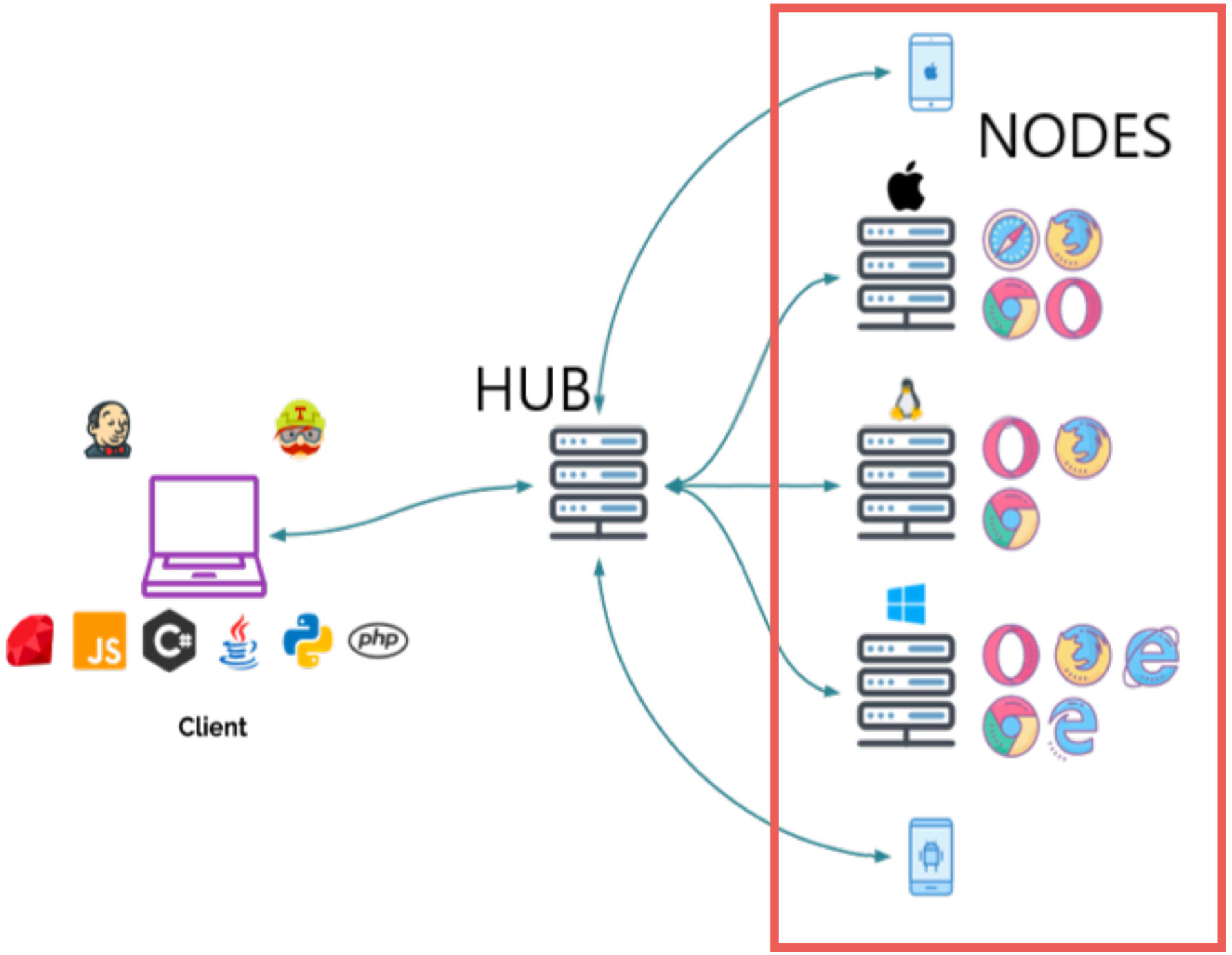
Selenium grid architecture



Selenium grid architecture



Selenium grid architecture



Selenium grid architecture

Hub
Nodes

Download selenium grid

http://selenium-release.storage.googleapis.com/index.html

Download selenium grid 4.0

Not Secure | selenium-release.storage.googleapis.com/index.html?path=4.0/

Index of /4.0/

Name	Last modified	Size	ETag
Parent Directory		-	
selenium-html-runner-4.0.0-alpha-1.jar	2019-04-24 16:17:02	13.52MB	2eca35318710f46d1ba5ed5543a906c9
selenium-html-runner-4.0.0-alpha-2.jar	2019-07-01 21:32:41	13.76MB	346d72e4f425bfec91c7073a46c96208
selenium-java-4.0.0-alpha-1.zip	2019-04-24 16:17:01	8.46MB	db9ed262a07c1cd2bb6098263c7f1e7b
selenium-java-4.0.0-alpha-2.zip	2019-07-01 21:32:33	8.74MB	2d31929580c3d829197eea97ade5f4f0
selenium-server-4.0.0-alpha-1.jar	2019-04-24 16:16:58	10.62MB	c32b1dd1c12cdb42b48f345d65d657fb
selenium-server-4.0.0-alpha-1.zip	2019-04-24 16:16:59	10.20MB	7f0bc4bb4fc2a5a7f0a262f62bf782d3
selenium-server-4.0.0-alpha-2.jar	2019-07-01 21:32:04	10.79MB	d0676e6b3ee508b48416aba603662573
selenium-server-4.0.0-alpha-2.zip	2019-07-01 21:32:11	10.47MB	fb19d62db44a7b163f1fbc2fff9dff0a
selenium-server-standalone-4.0.0-alpha-1.jar	2019-04-24 16:17:00	11.98MB	ac553ec987d16d2af8c8e3ef9061772c
selenium-server-standalone-4.0.0-alpha-1.zip	2019-04-24 16:17:00	11.77MB	1974b11f970bad6e15c84e3840ec3897
selenium-server-standalone-4.0.0-alpha-2.jar	2019-07-01 21:32:19	12.33MB	d000d97d24389fde5bfb94f450ede780
selenium-server-standalone-4.0.0-alpha-2.zip	2019-07-01 21:32:27	12.26MB	2466773c71eeddea02004371a5e32324

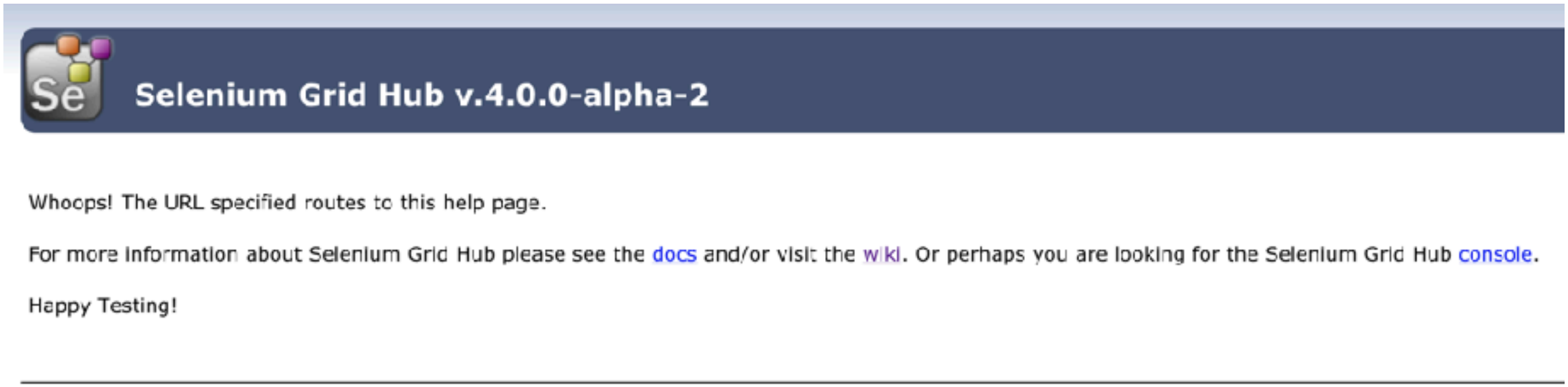
http://selenium-release.storage.googleapis.com/index.html

Start Hub

$java -jar selenium-server-standalone-<version>.jar **-role hub**

Start Hub

Open url=http://localhost:4444

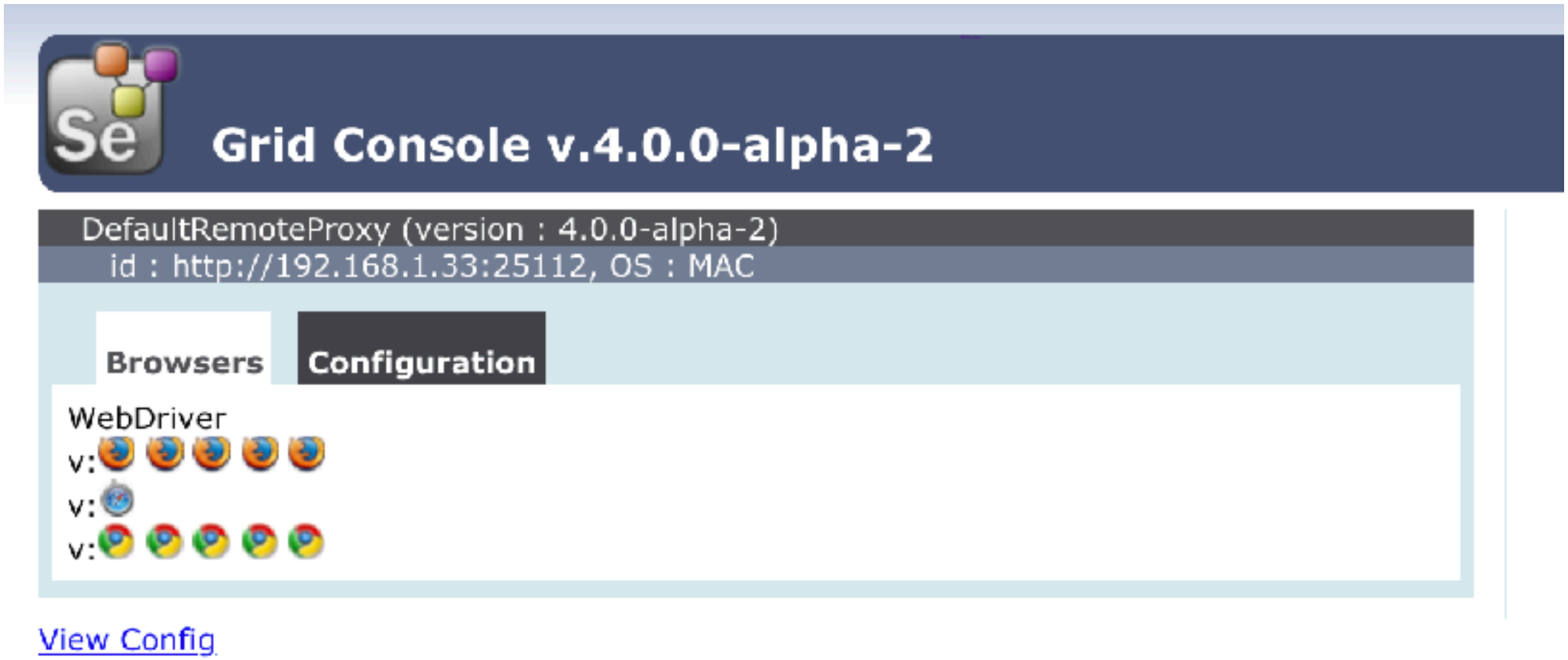


Start Node

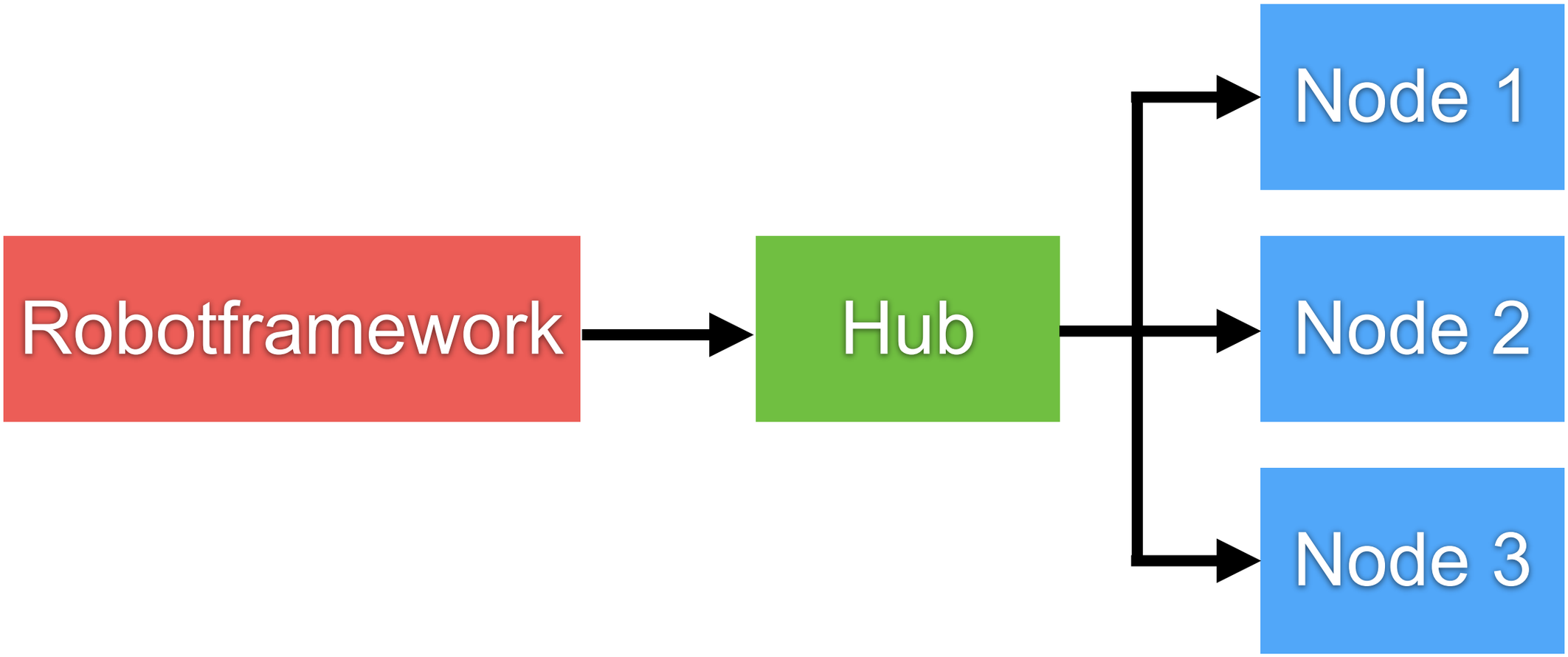
$java -jar selenium-server-standalone-<version>.jar **-role node -hub http://localhost:4444/grid/register**

Start Node

Open url=http://localhost:4444/grid/console



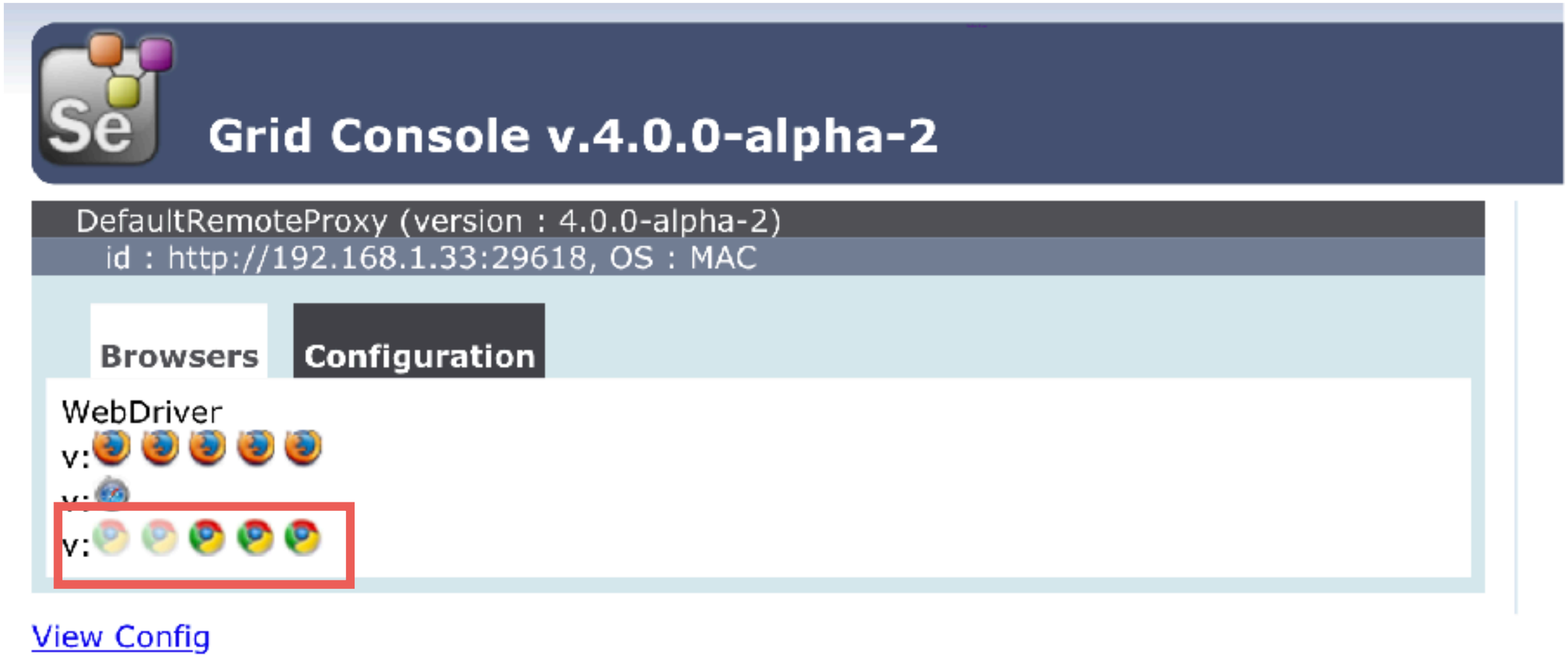
Testing with Robotframework



Testing with Robotframework

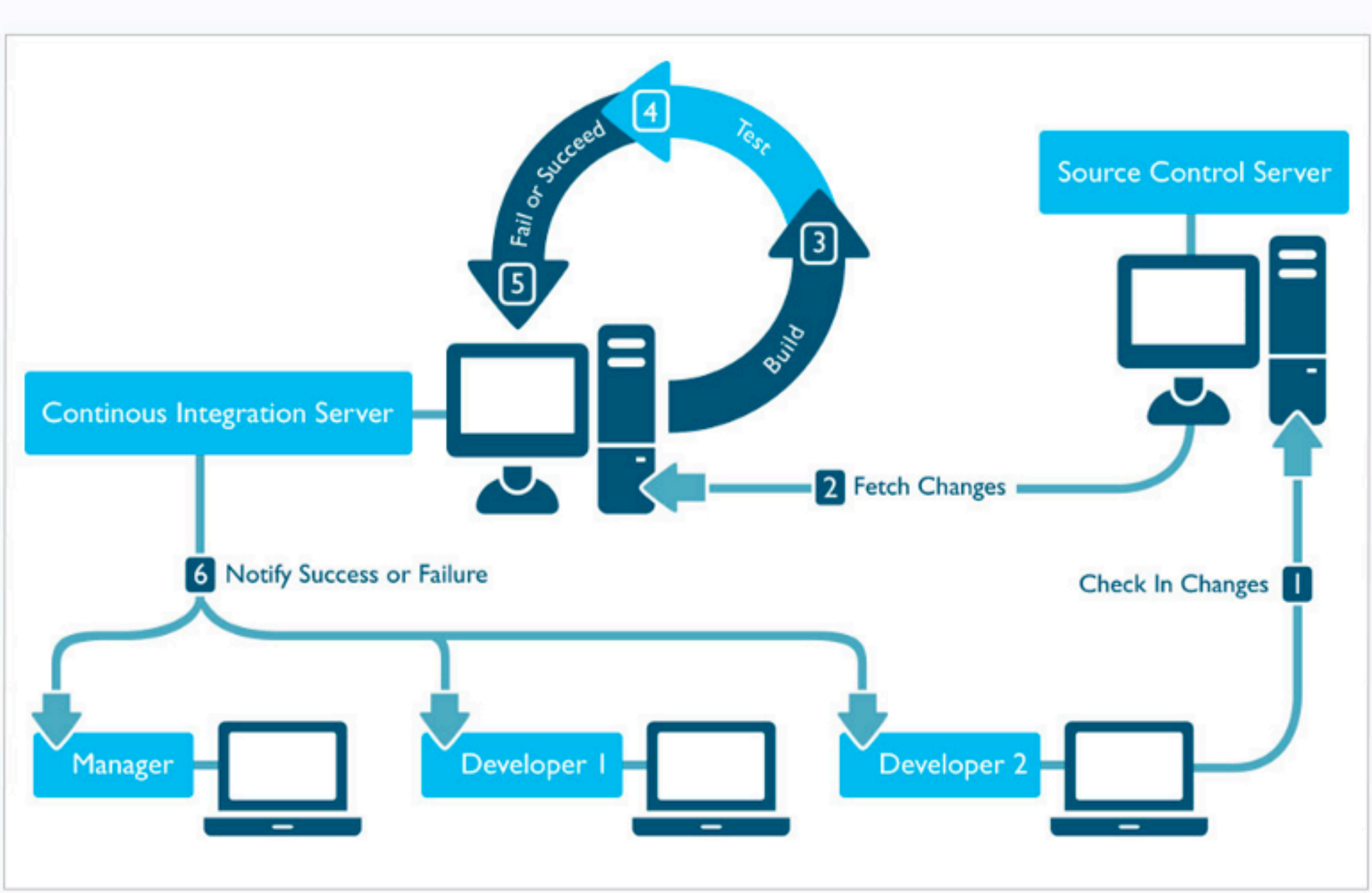
```
*** Keywords ***
Open with selenium grid
    Open Browser    ${URL}
    ...   browser=${BROWSER}
    ...   remote_url=http://localhost:4444/wd/hub
    ...   desired_capabilities=browserName:chrome
```

State of nodes in selenium grid



Continuous Integration

Continuous Integration process



Close browser in each test case

```
*** Settings ***
Resource  ./pages/welcome.robot
Resource  ./pages/catalog.robot
Test Teardown  Close Browser
```

Workshop with Selenium grid

API testing

API testing

Robot framework
Postman

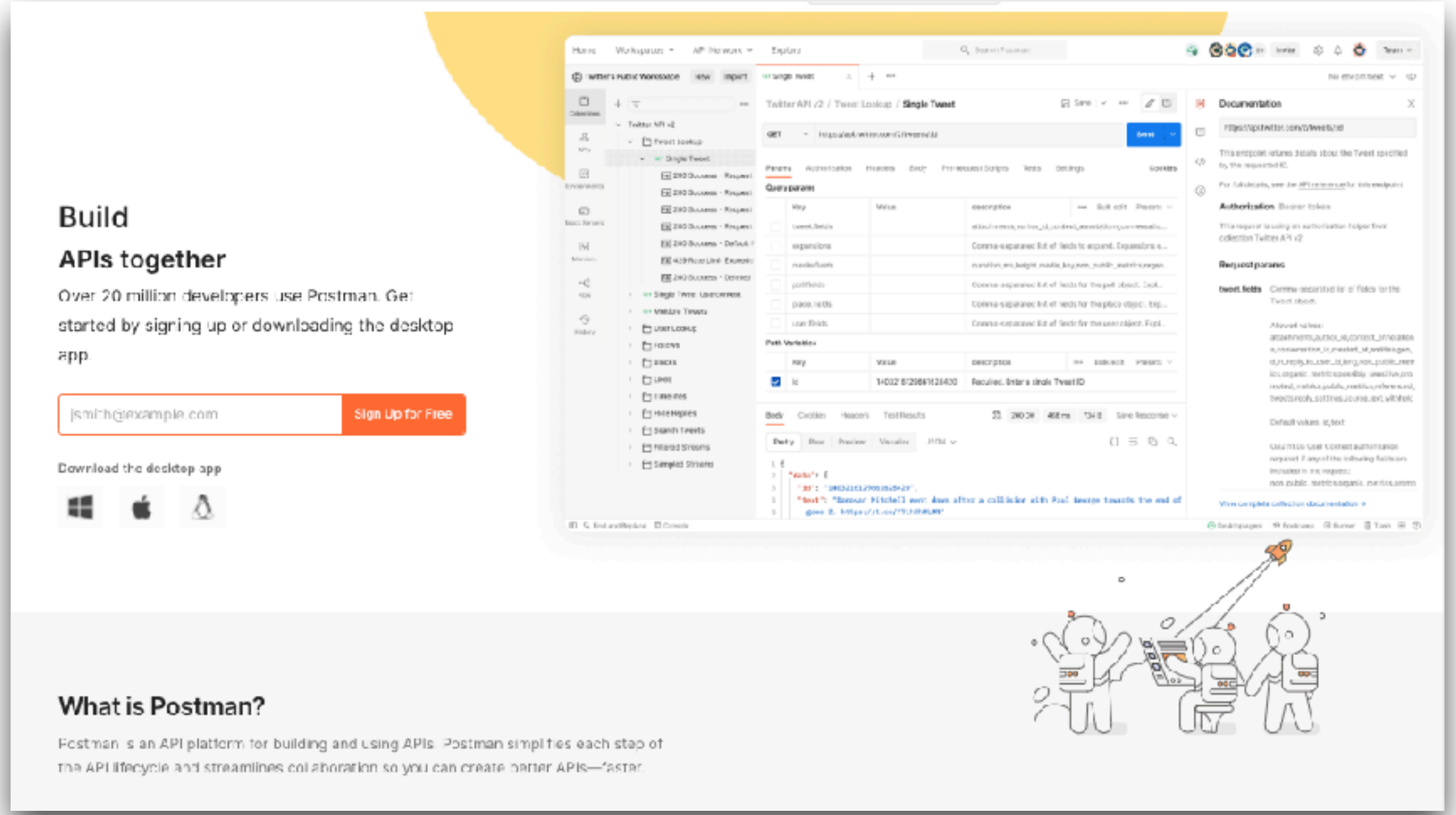
API testing with Robot

Using RequestsLibrary

$pip install -U requests
$pip install -U robotframework-requests

https://github.com/bulkan/robotframework-requests

API testing with Postman



https://www.postman.com/

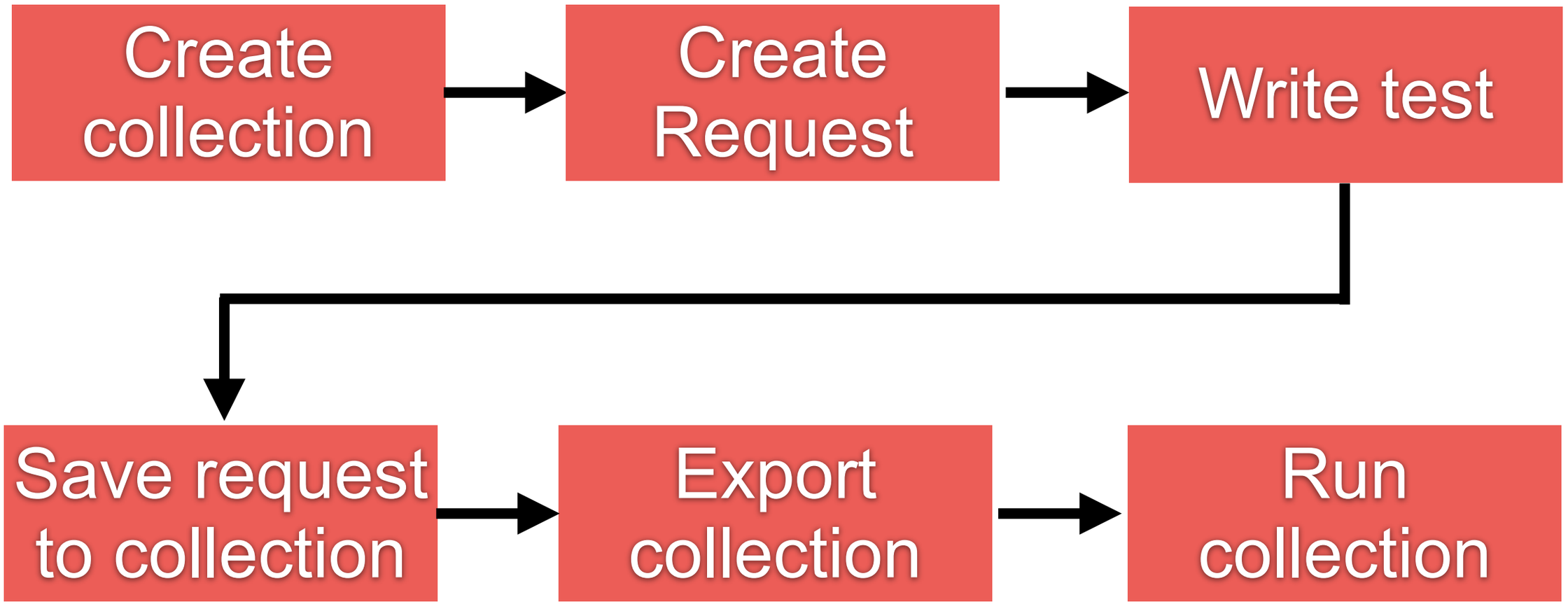
Postman in command line

$npm install -g newman
$newman run <collection file>
$newman run <collection file> **-r cli,junit**

$newman run <collection file> **-r cli,junit**
--reporter-junit-export outputfile.xml

https://www.npmjs.com/package/newman

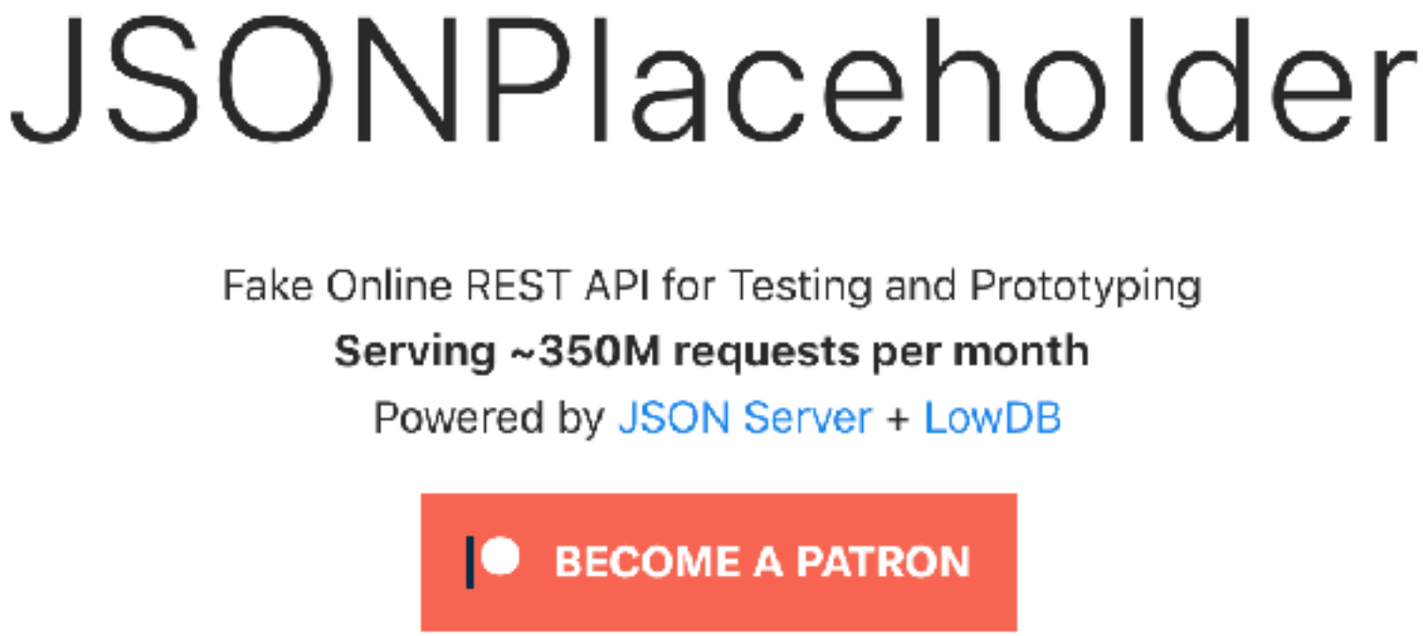
Testing Flow



Workshop API testing

Workshop API testing

https://jsonplaceholder.typicode.com/



Workshop API testing

https://jsonplaceholder.typicode.com/users


```
[
  - {
        id: 1,
        name: "Leanne Graham",
        username: "Bret",
        email: "Sincere@april.biz",
    - address: {
        street: "Kulas Light",
        suite: "Apt. 556",
        city: "Gwenborough",
        zipcode: "92998-3874",
      - geo: {
            lat: "-37.3159",
            lng: "81.1496"
        }
    },
```


Command line

Tips and Tricks

Google Chrome

Auto closed browser !!
Disabled notification
Disabled camera and audio (media)

Disabled Auto closed browser

```
*** Settings ***
Library   SeleniumLibrary

*** Variables ***
${url}   http://localhost:3000/demo.html
${browser}   chrome

*** Test Cases ***
Disable Camera and Media devices
    Open Browser   ${url}    ${browser}
    ...    options=add_experimental_option("detach", True)
```

Disabled notification

```
*** Settings ***
Library   SeleniumLibrary

*** Variables ***
${url}   http://localhost:3000/demo.html
${browser}   chrome

*** Test Cases ***
Disable Noti
    Open Browser   ${url}    ${browser}
    ...   options=add_argument("--disable-notifications")
```

Disabled camera and audio (media)

```
*** Settings ***
Library   SeleniumLibrary

*** Variables ***
${url}   http://localhost:3000/demo.html
${browser}   chrome

*** Test Cases ***
Disable Camera and Media devices
    Open Browser   ${url}    ${browser}
    ...   options=add_argument("--use-fake-ui-for-media-stream")
```

Working with CSV file

Working with CSV file

Using csv library from Python 3
Read and Write

https://docs.python.org/3/library/csv.html

Working with CSV file

Design first with (Robot keyword)

```
1    *** Settings ***
2    Library    csv_library.py
3
4    *** Test Cases ***
5    Design process
6        ${result}=    Read    users.csv
7        Write    $result    new.csv
```

Working with CSV file

Coding with Python (read and write)

```python
import csv

def read(filename):
    with open(filename) as csvfile:
        spamreader = csv.reader(csvfile)
        for row in spamreader:
            for c in row:
                print(c, end='-')
            print()
```

Q/A