

Robot Framework workshop





Somkiat Puisungnoen

Search

Somkiat | Home

Update Info 1 View Activity Log 10+ ...

Timeline About Friends 3,138 Photos More

When did you work at Opendream? X

... 22 Pending Items

Post Photo/Video Live Video Life Event

What's on your mind?

Public Post

Intro

Software Craftsmanship

Software Practitioner at สยามชานนาภิเษก พ.ศ. 2556

Agile Practitioner and Technical at SPRINT3r

Somkiat Puisungnoen 15 mins · Bangkok · ...

Java and Bigdata



Facebook somkiat.cc

Page Messages Notifications 3 Insights Publishing Tools Settings Help ▾

somkiat.cc
@somkiat.cc

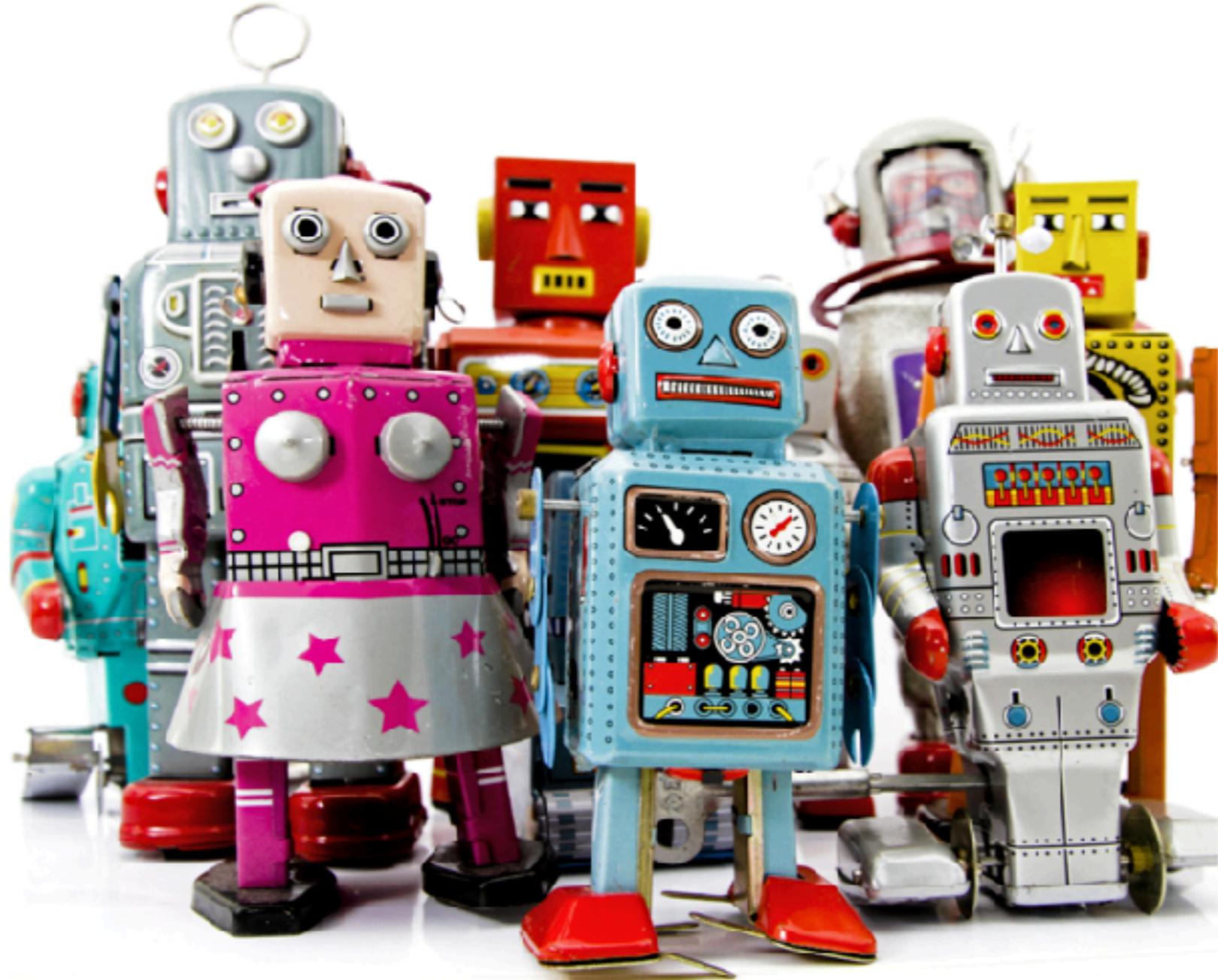
Home Posts Videos Photos

Liked Following Share ... + Add a Button



**[https://github.com/up1/
course-robotframework](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

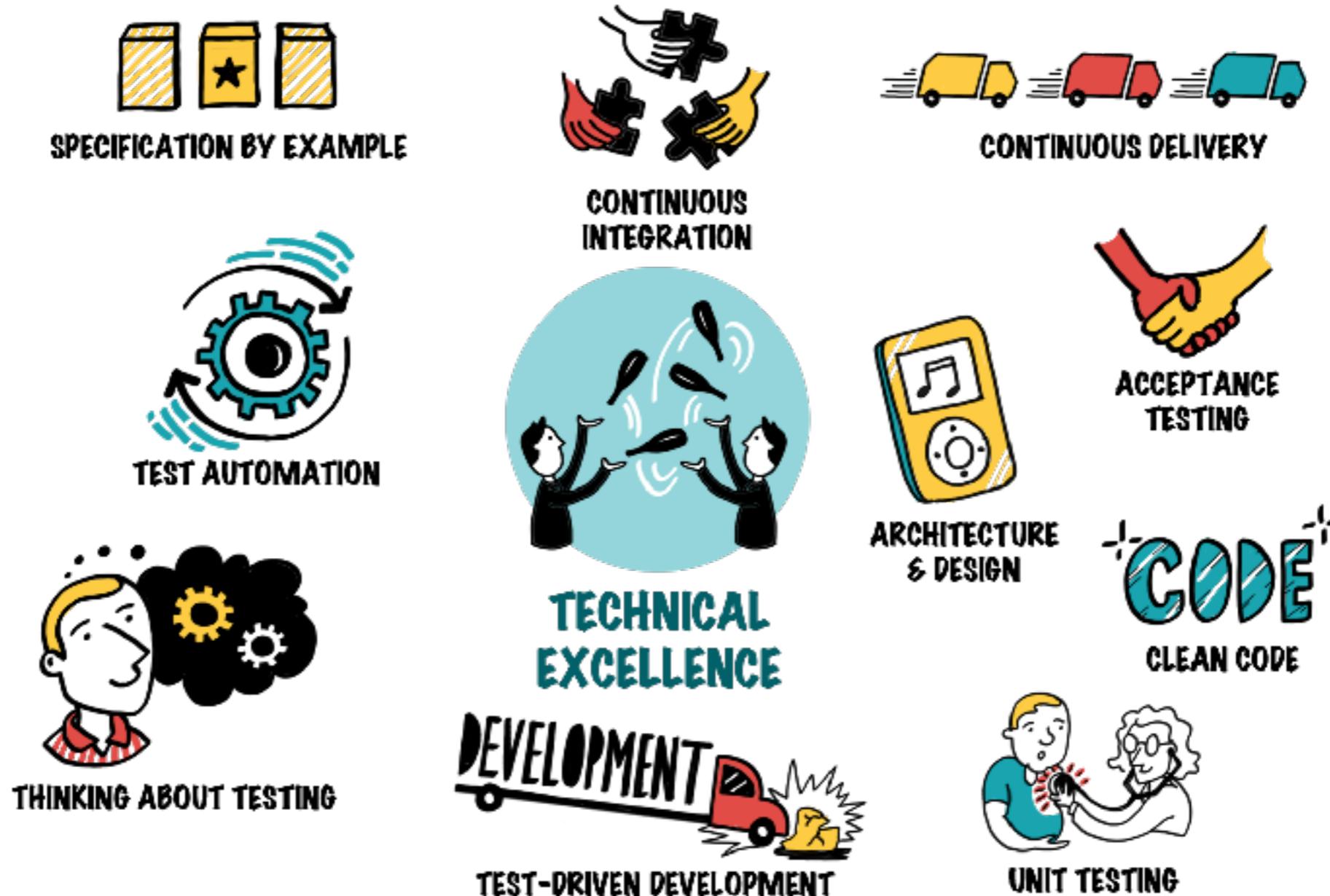


Agenda

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



Technical Excellence



<http://less.works>

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



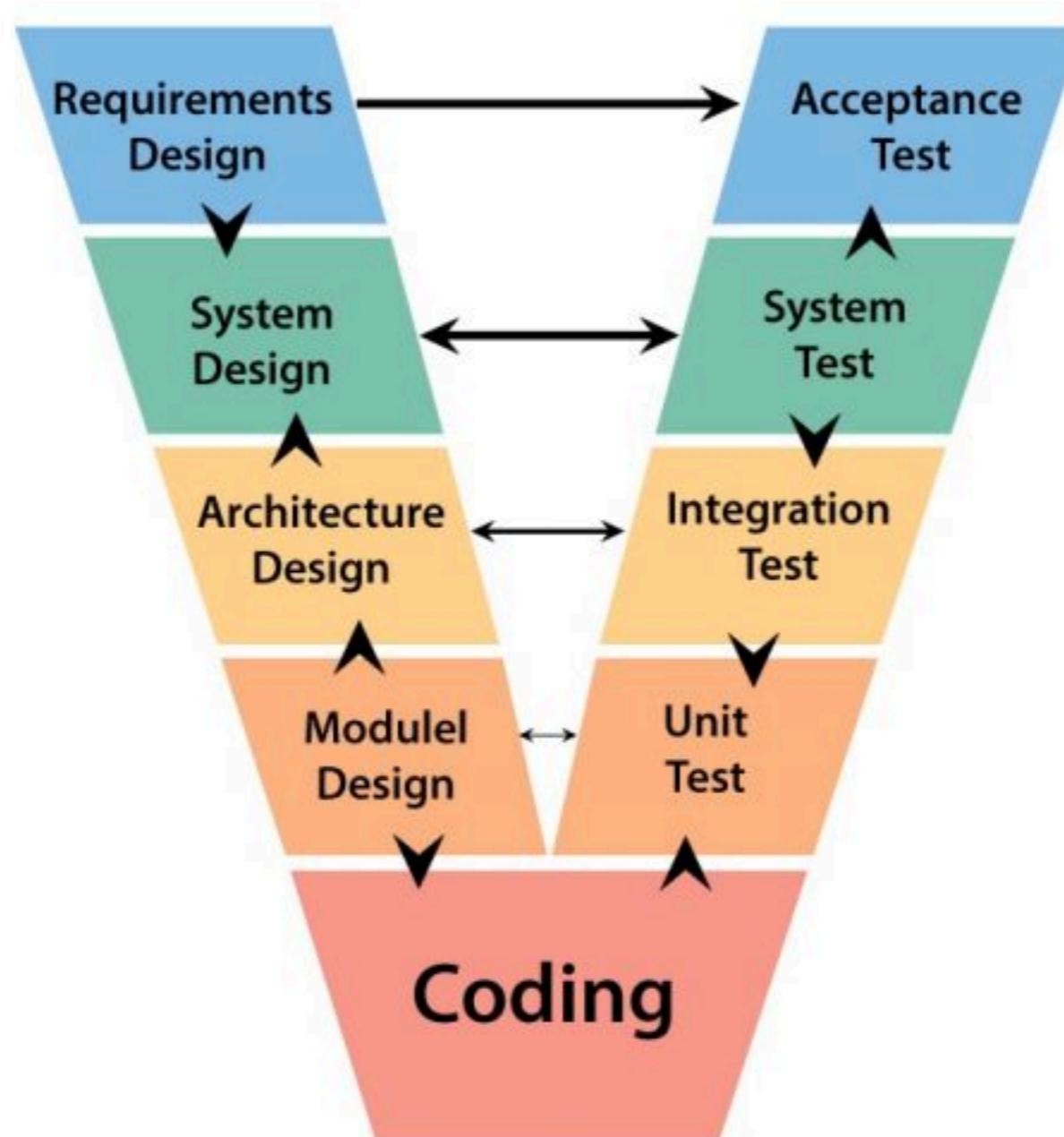
Robot Framework

© 2017 - 2024 Siam Chamnankit Company Limited. All rights reserved.

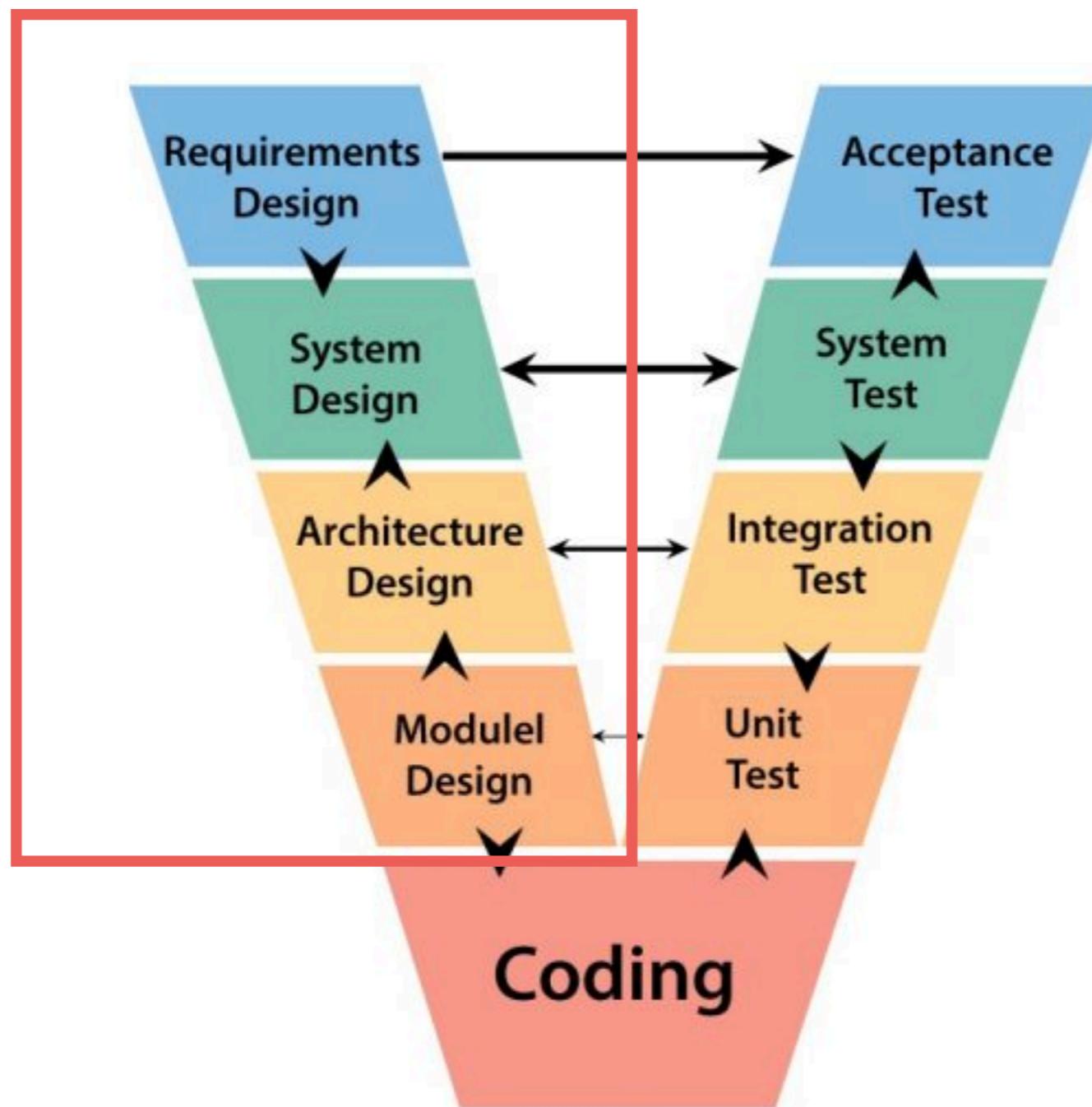
Software Development



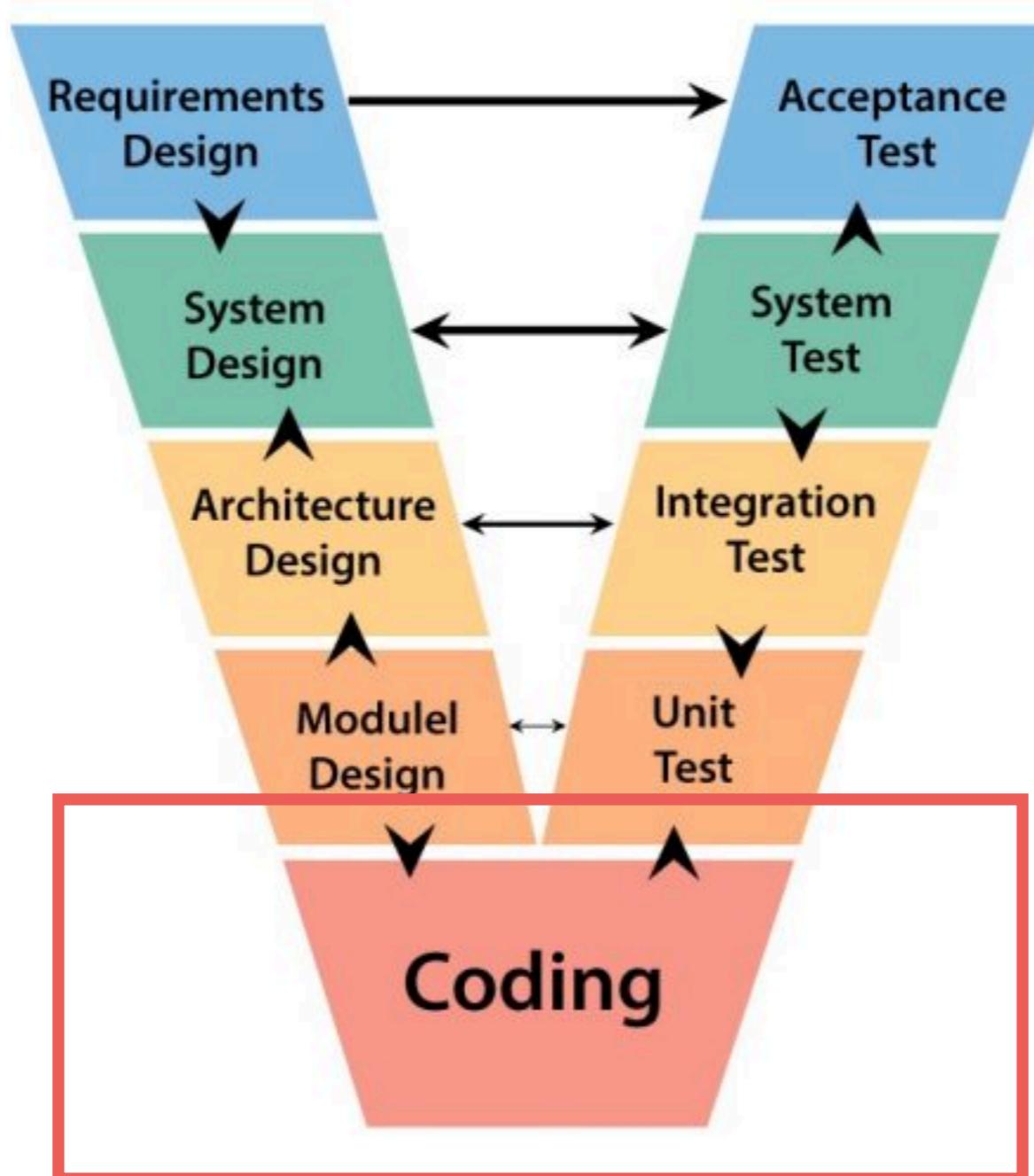
V Model or Waterfall Model



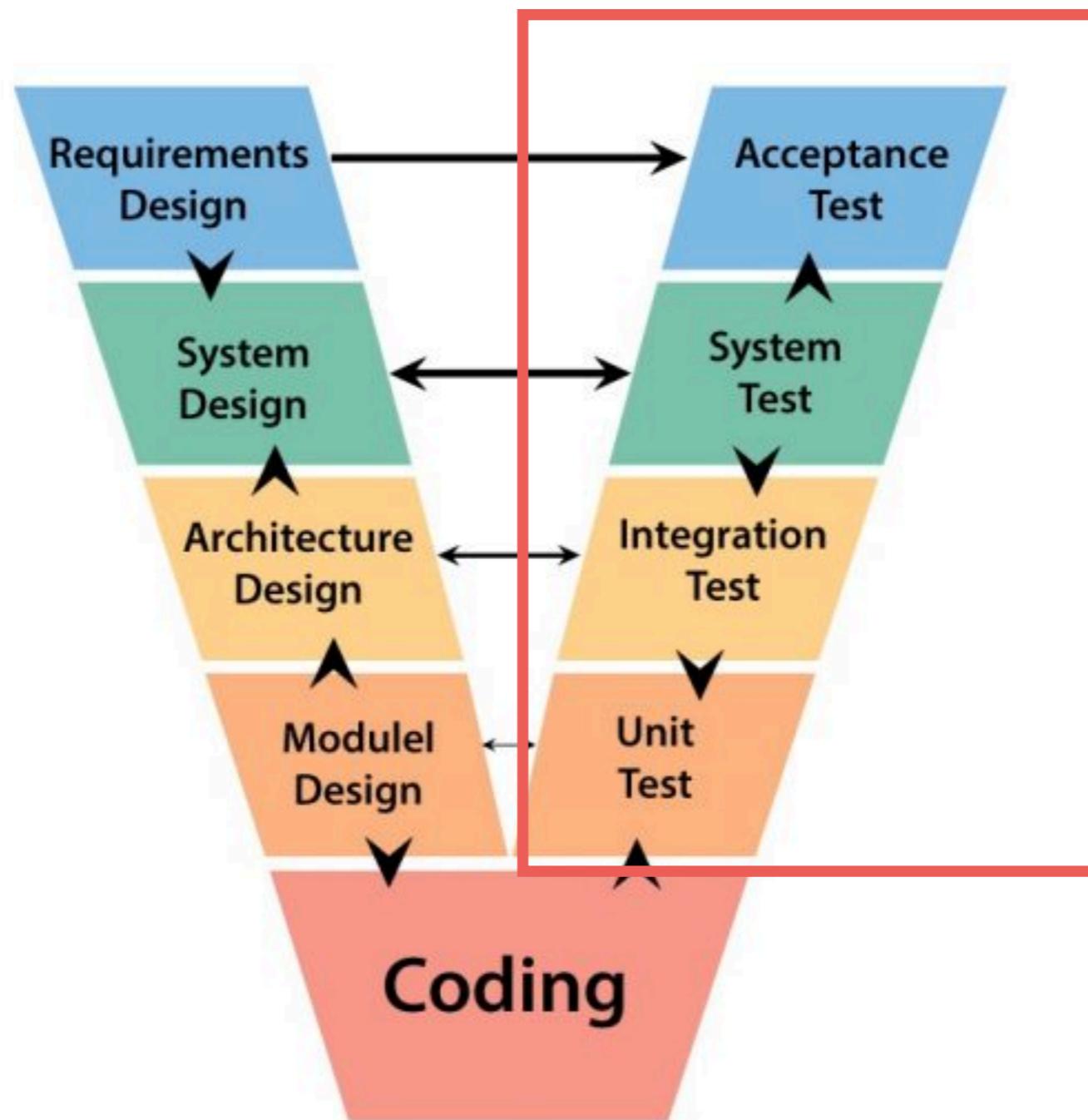
Verification phase



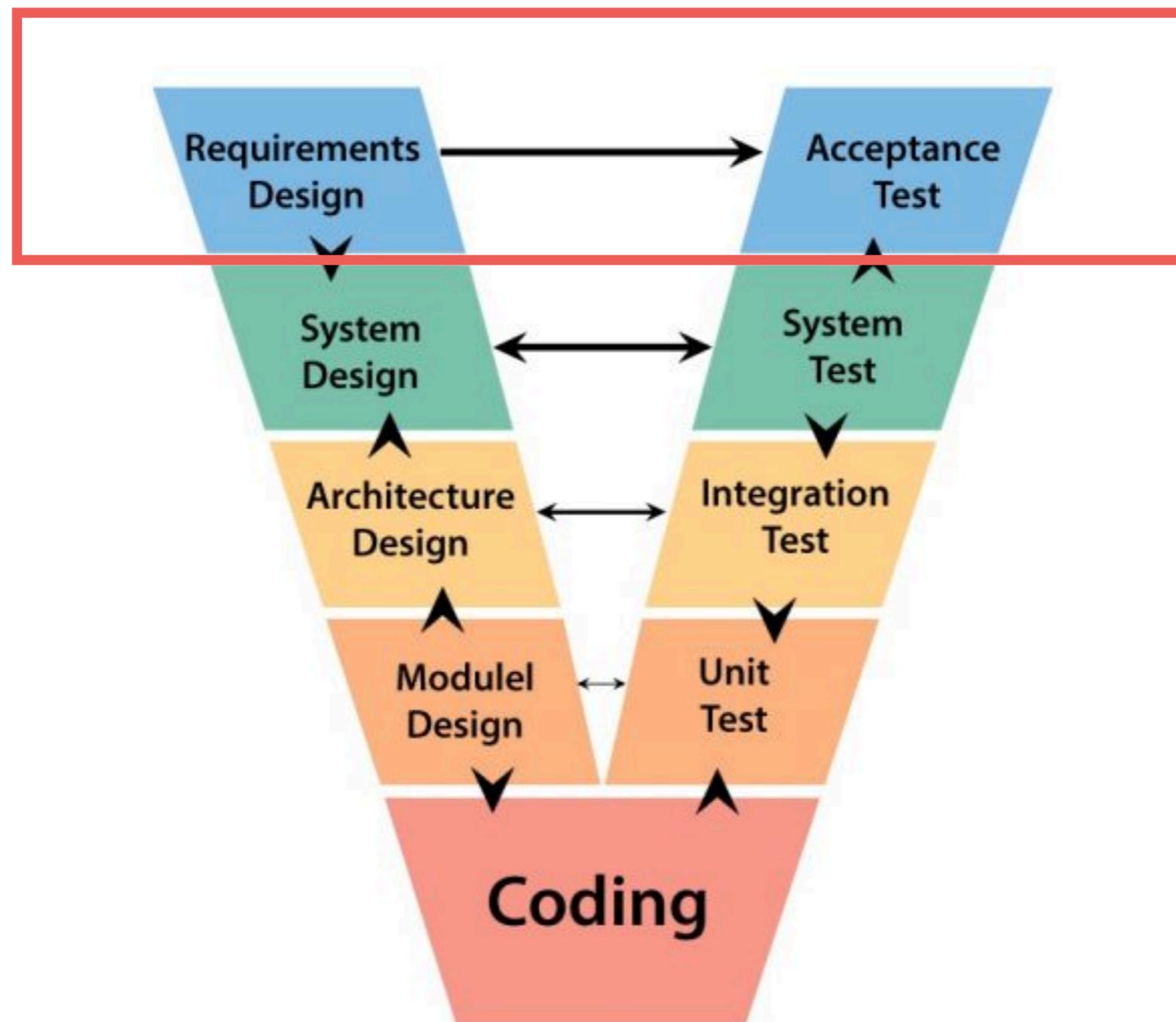
Coding phase



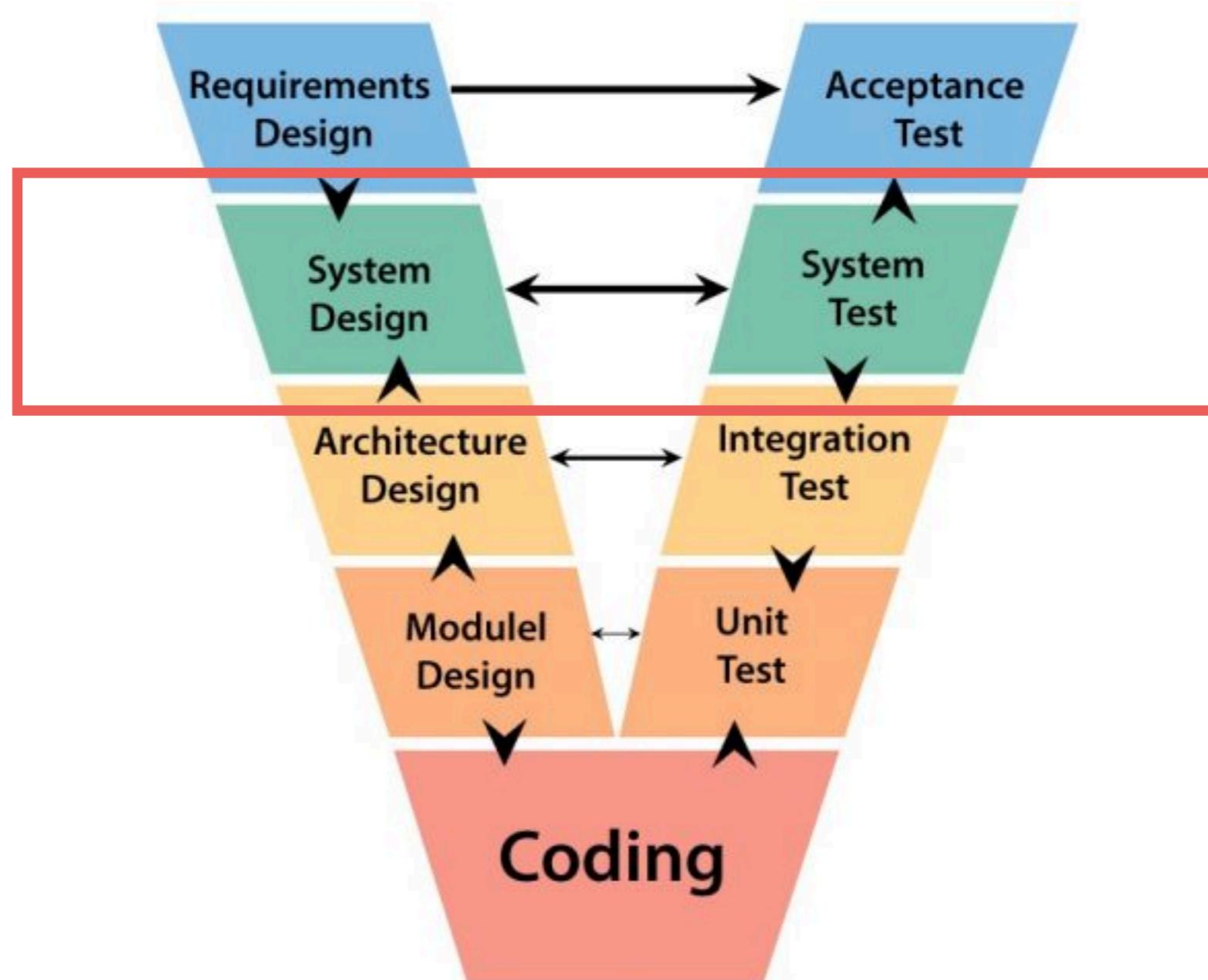
Validation phase



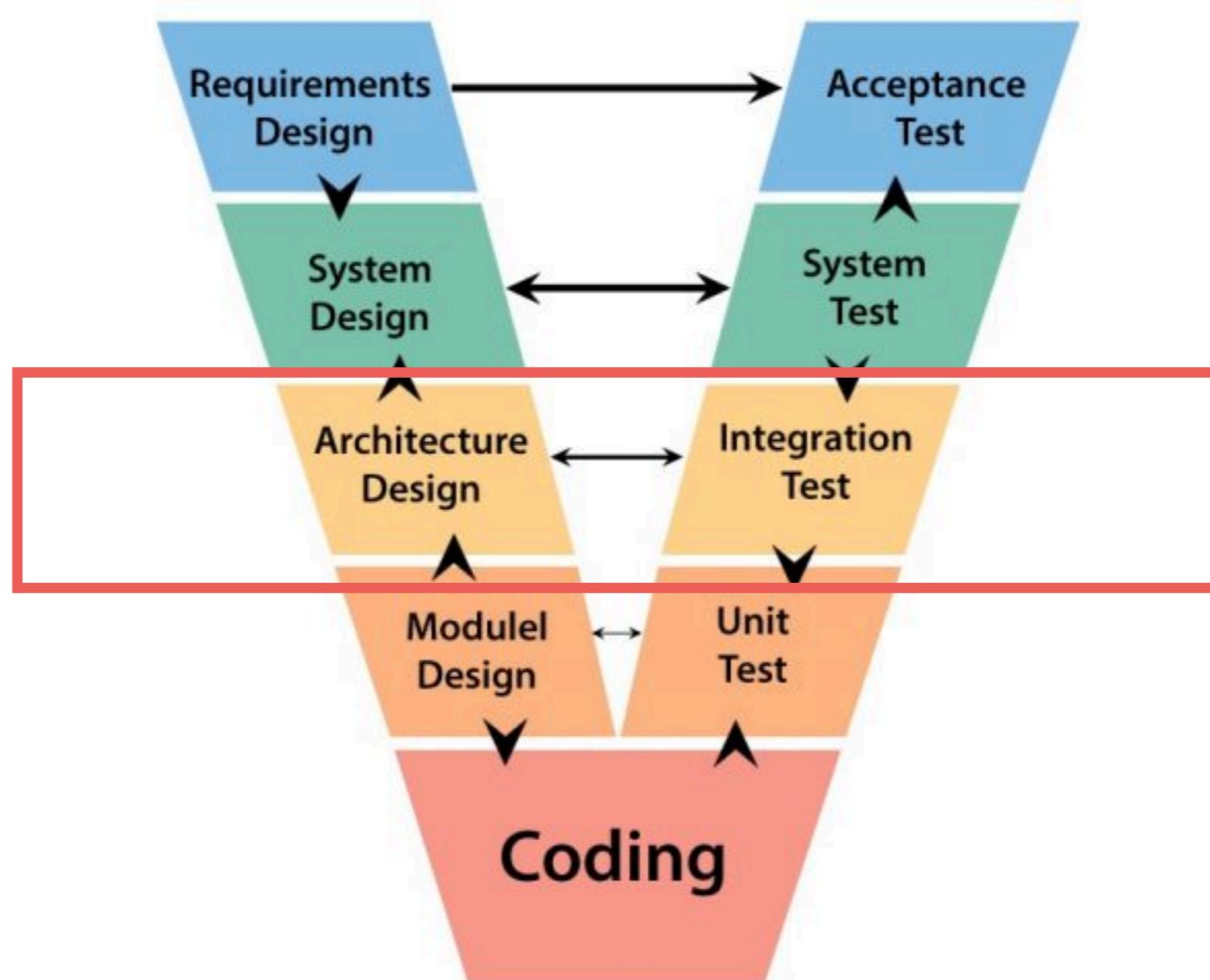
Verification and Validation



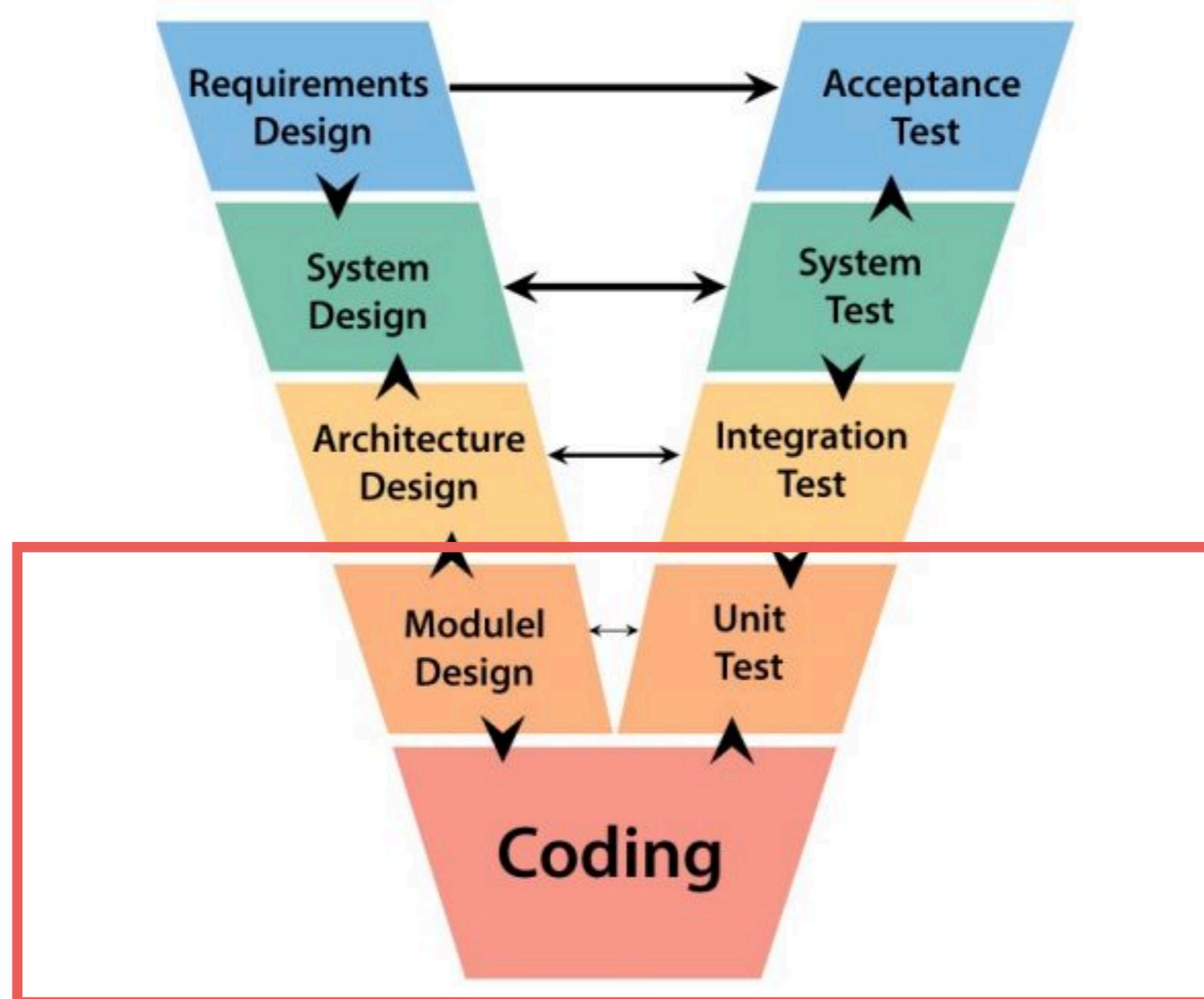
Verification and Validation



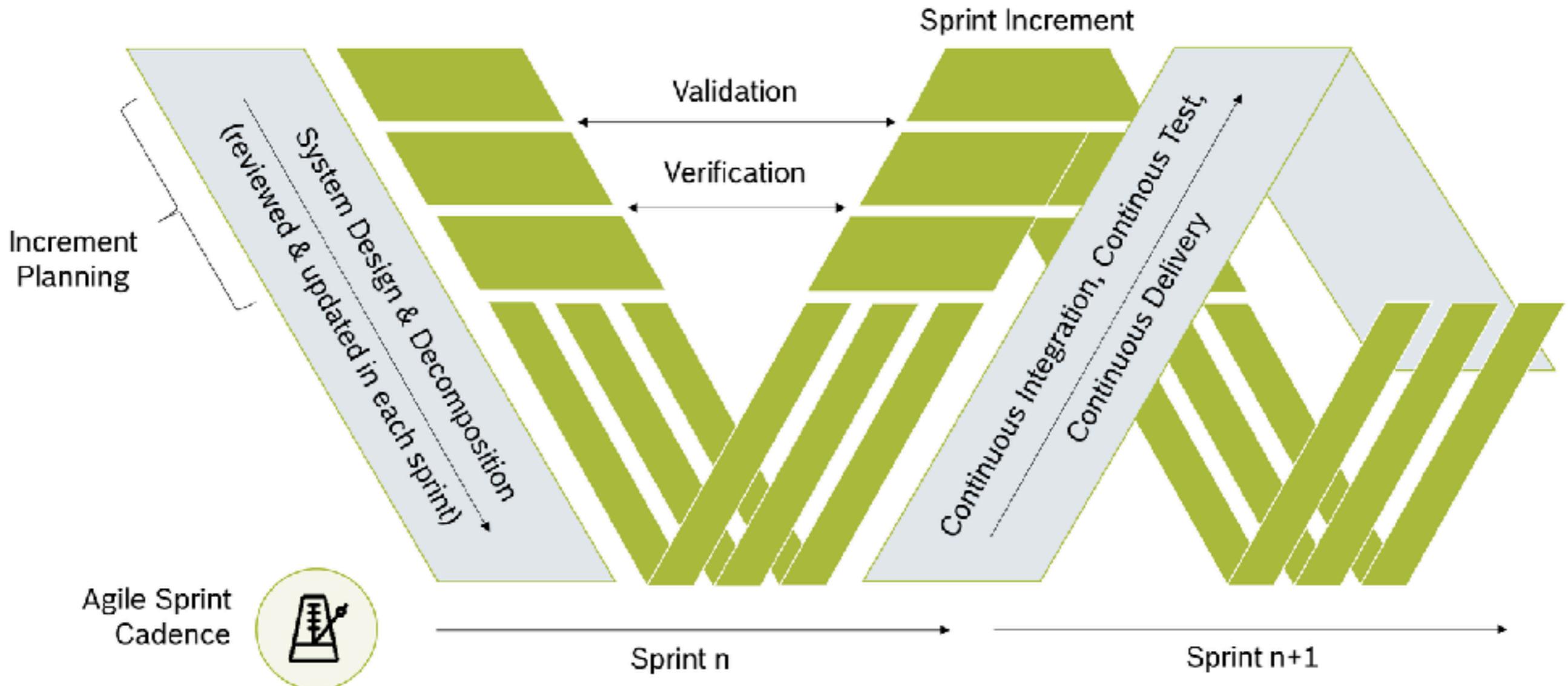
Verification and Validation



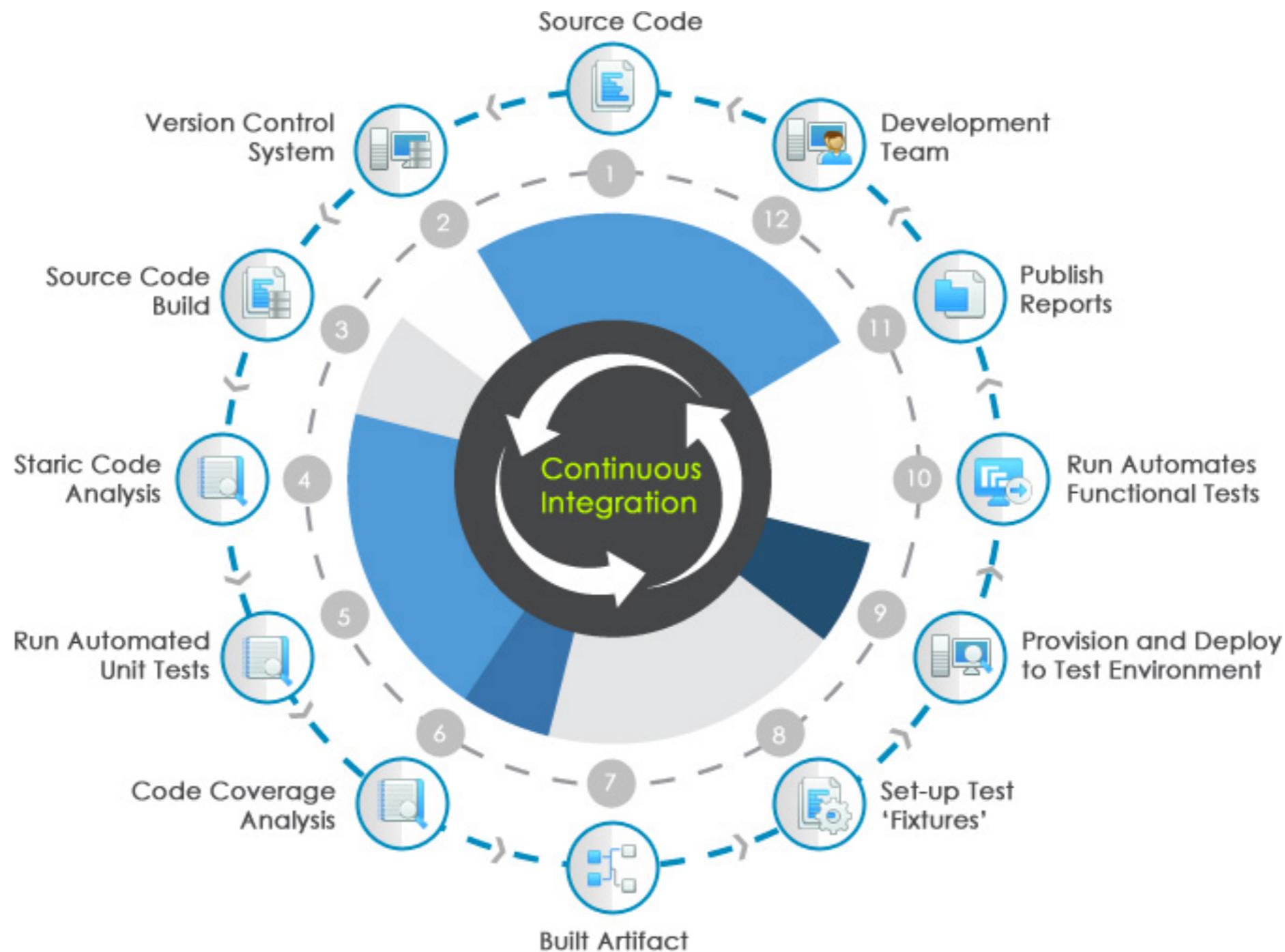
Verification and Validation



Iterative and Incremental



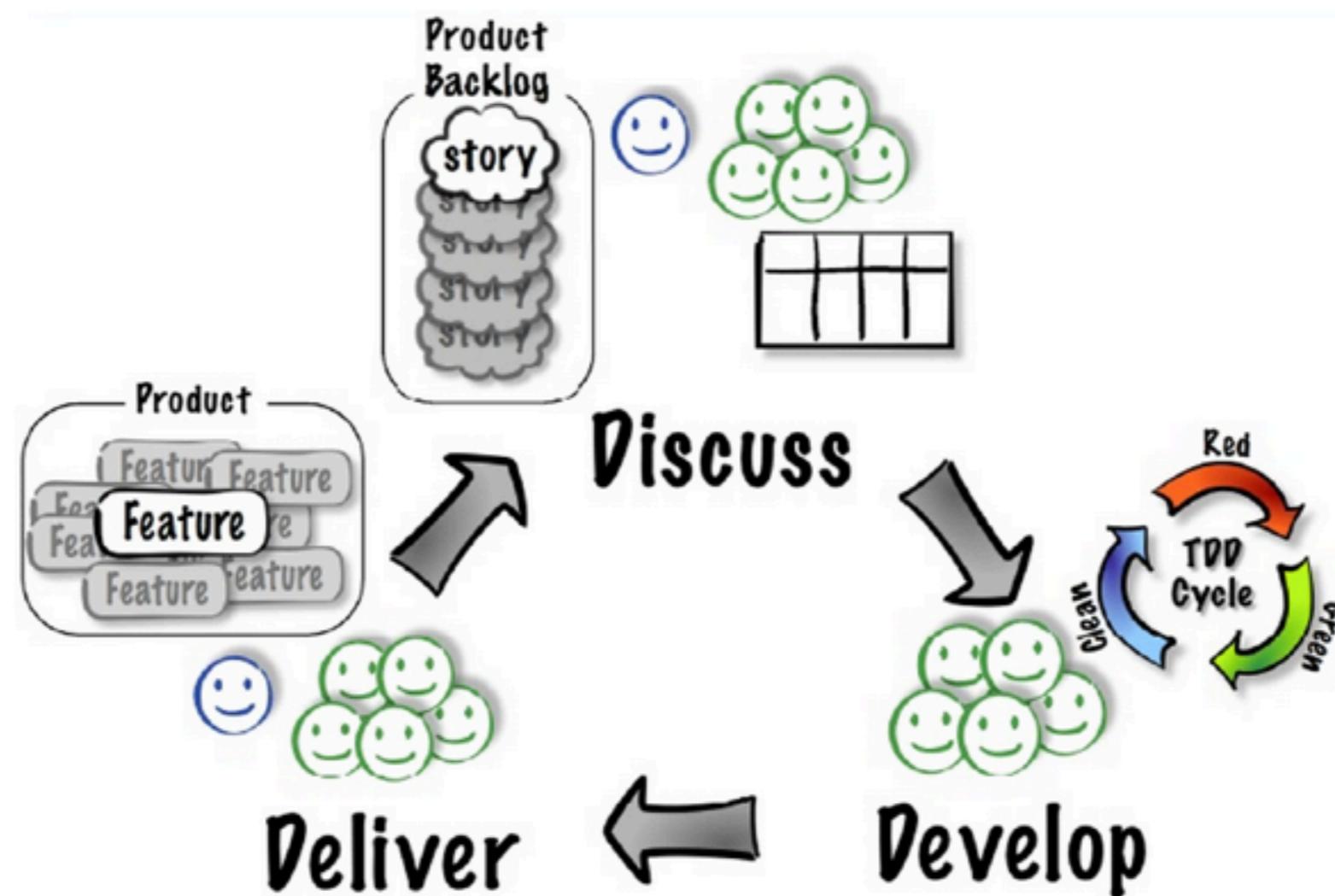
Continuous Integration process



Acceptance Test-Driven Development (ATDD)

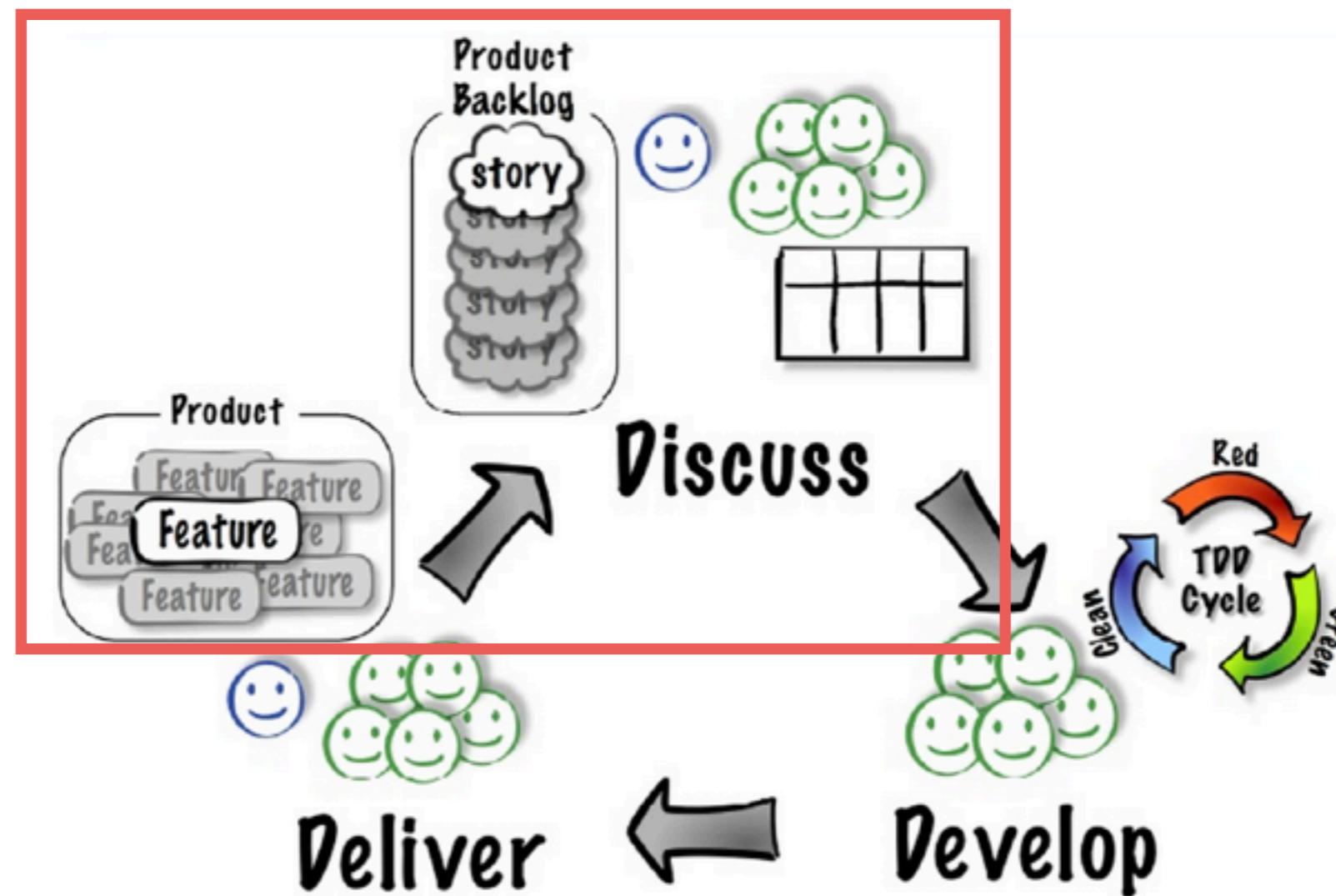


Acceptance Test-Driven Development (ATDD)



Acceptance Test-Driven Development (ATDD)

Discussion process



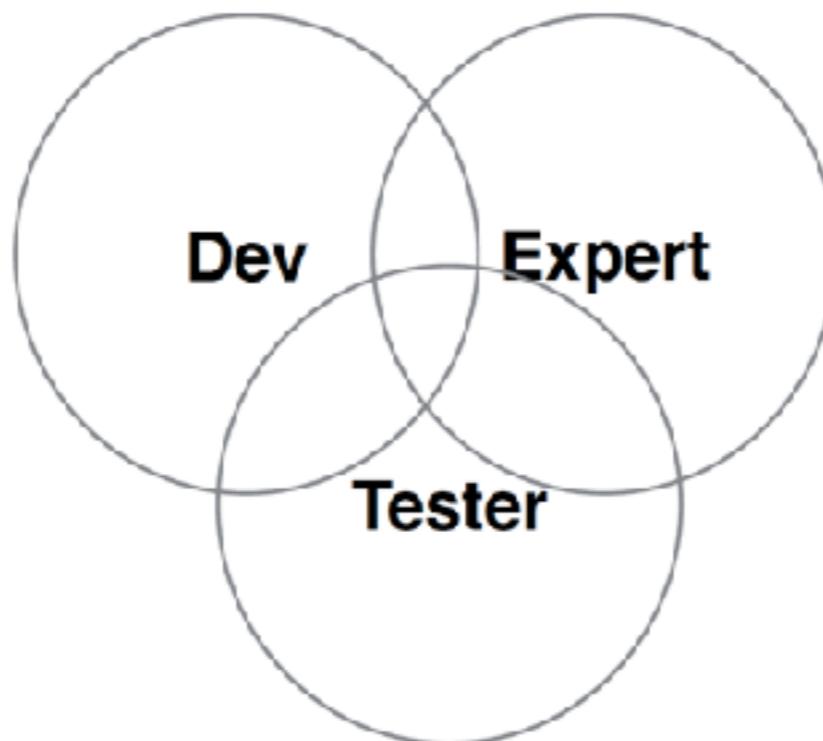
Discussion process

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

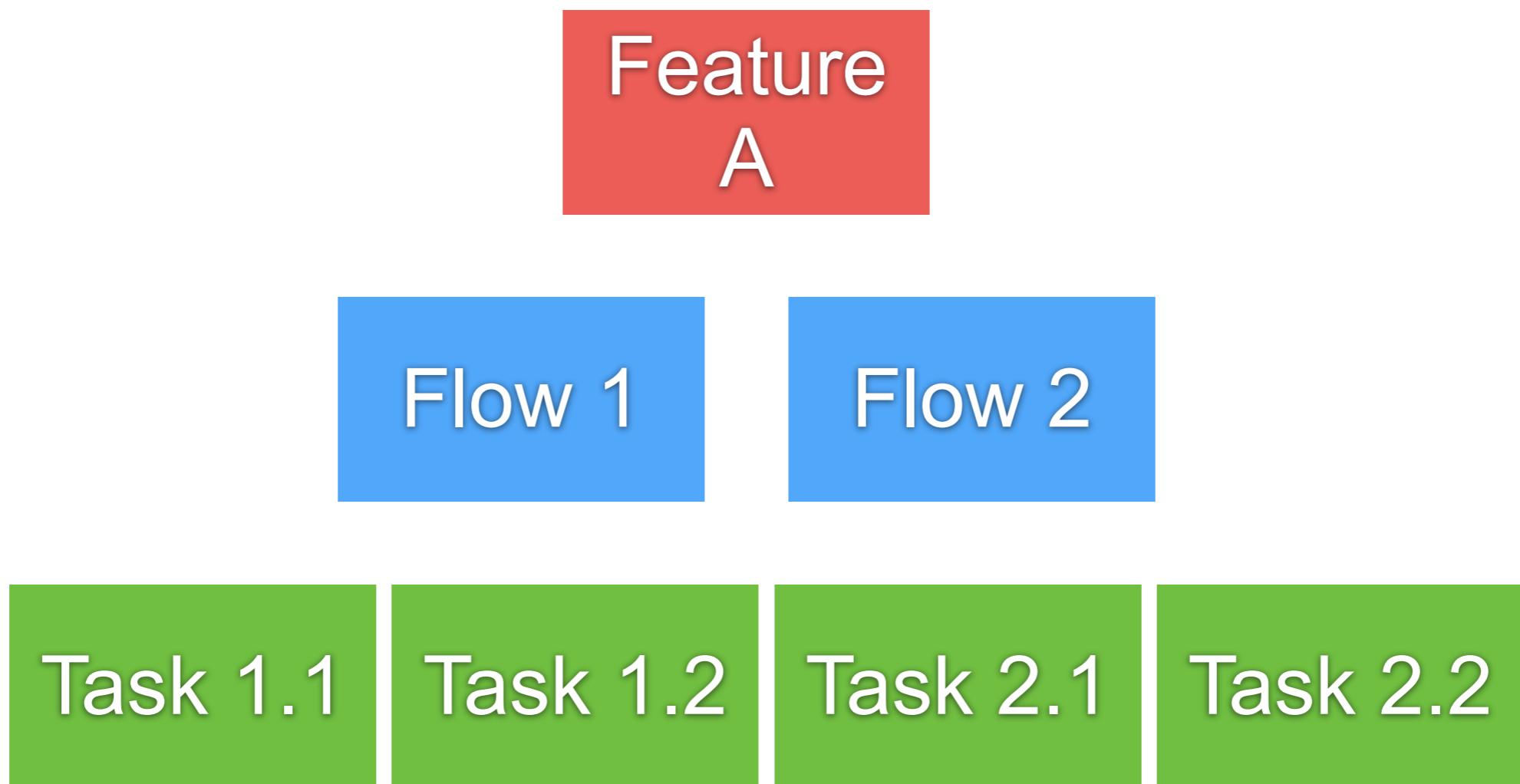


Decompose or Slicing feature

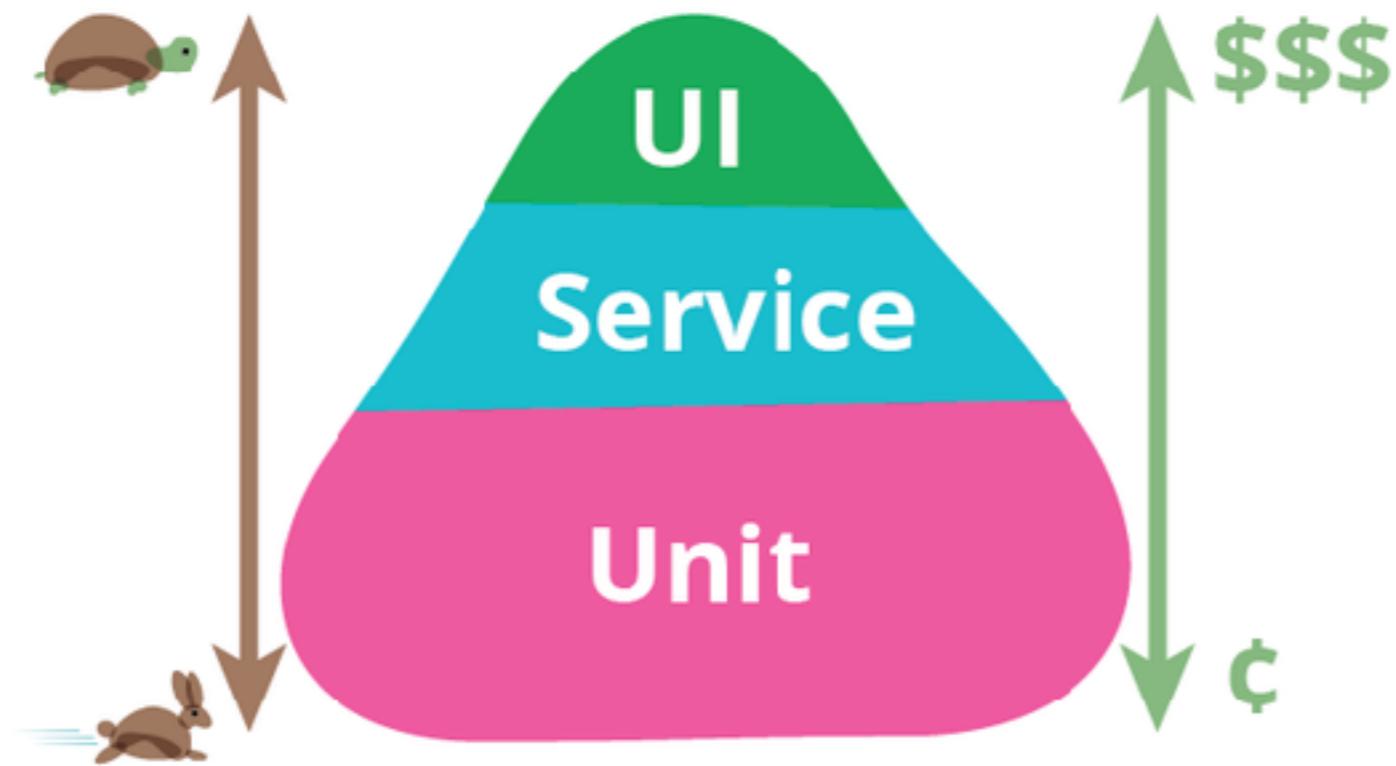
Feature	Flow/User Story	Task	Task
Feature 1	Flow 1.1	Task 1.1.1	Task 1.1.4
Feature 2	Flow 1.2	Task 1.1.2	Task 1.1.5
Feature 3	Flow 1.3	Task 1.1.3	Task 1.1.6



Decompose or Slicing feature



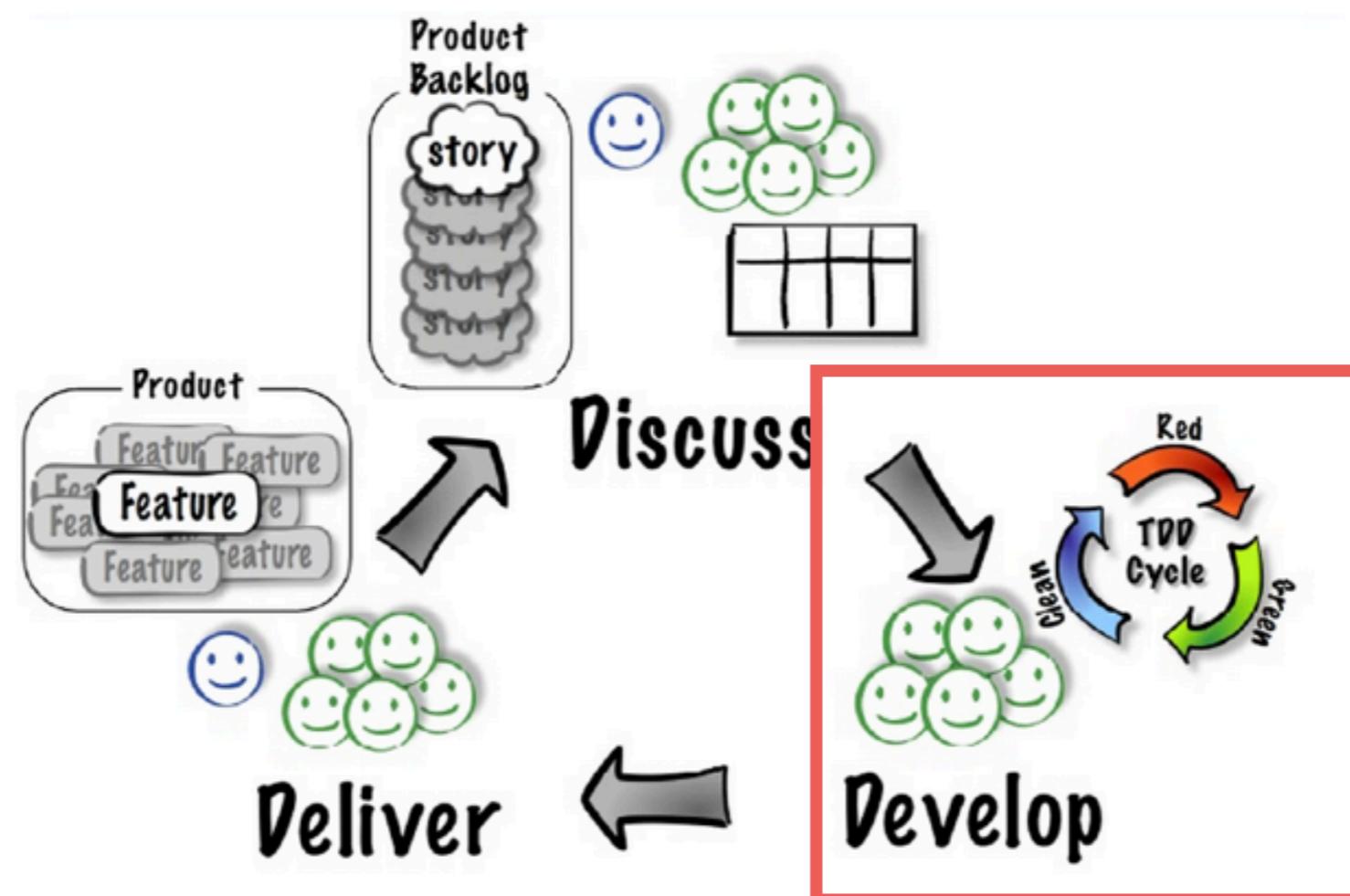
Pyramid Testing



Development process

=

Coding + Testing



Development process

Implementation with example/data

All examples are passed

Re-Test ?

Regression Testing ?



Re-test vs Regression test

Re-test

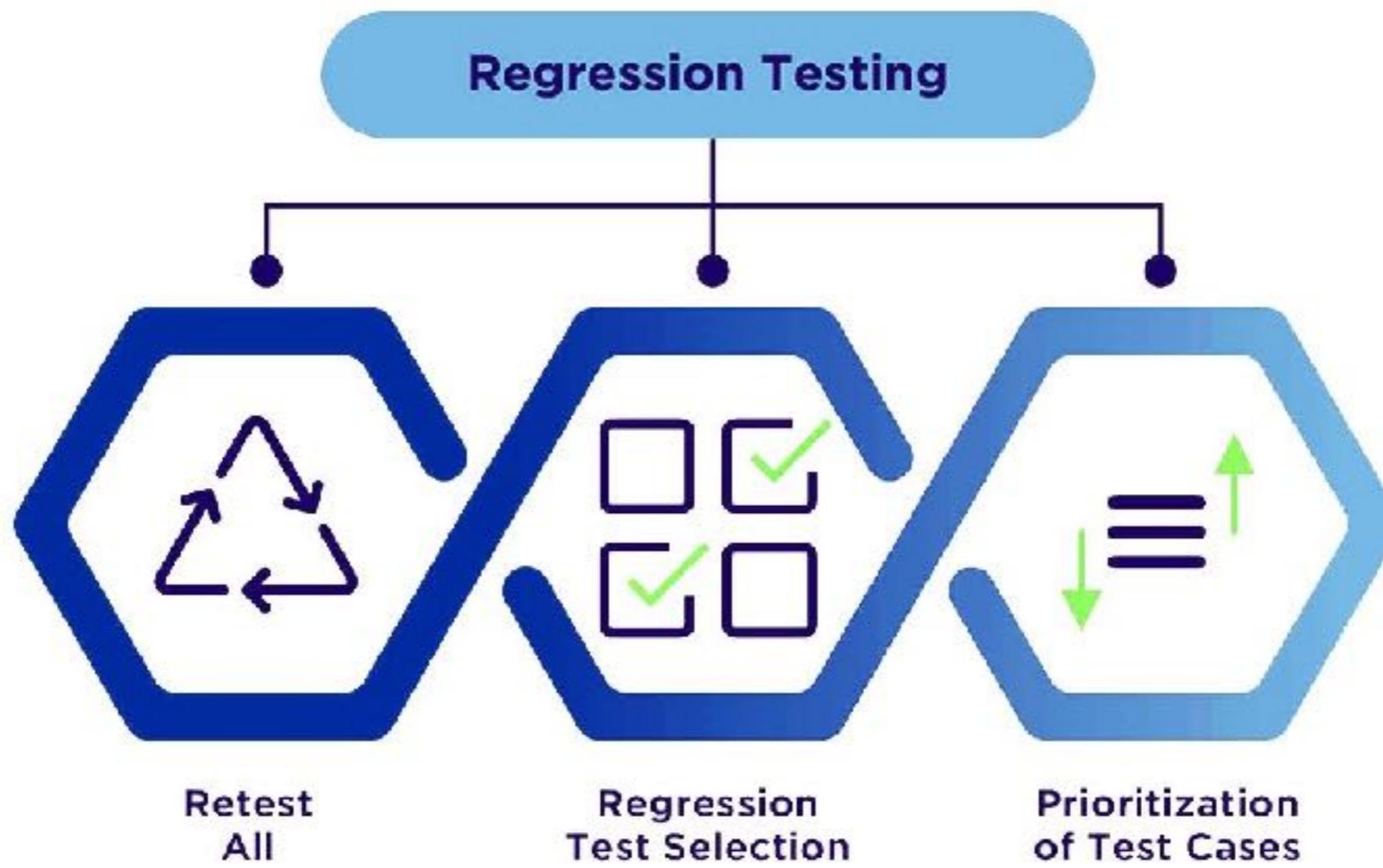
Testing code which was not working,
But you believe to have been fixed

Regression test

Testing code which was not working,
But now, due to updates, might

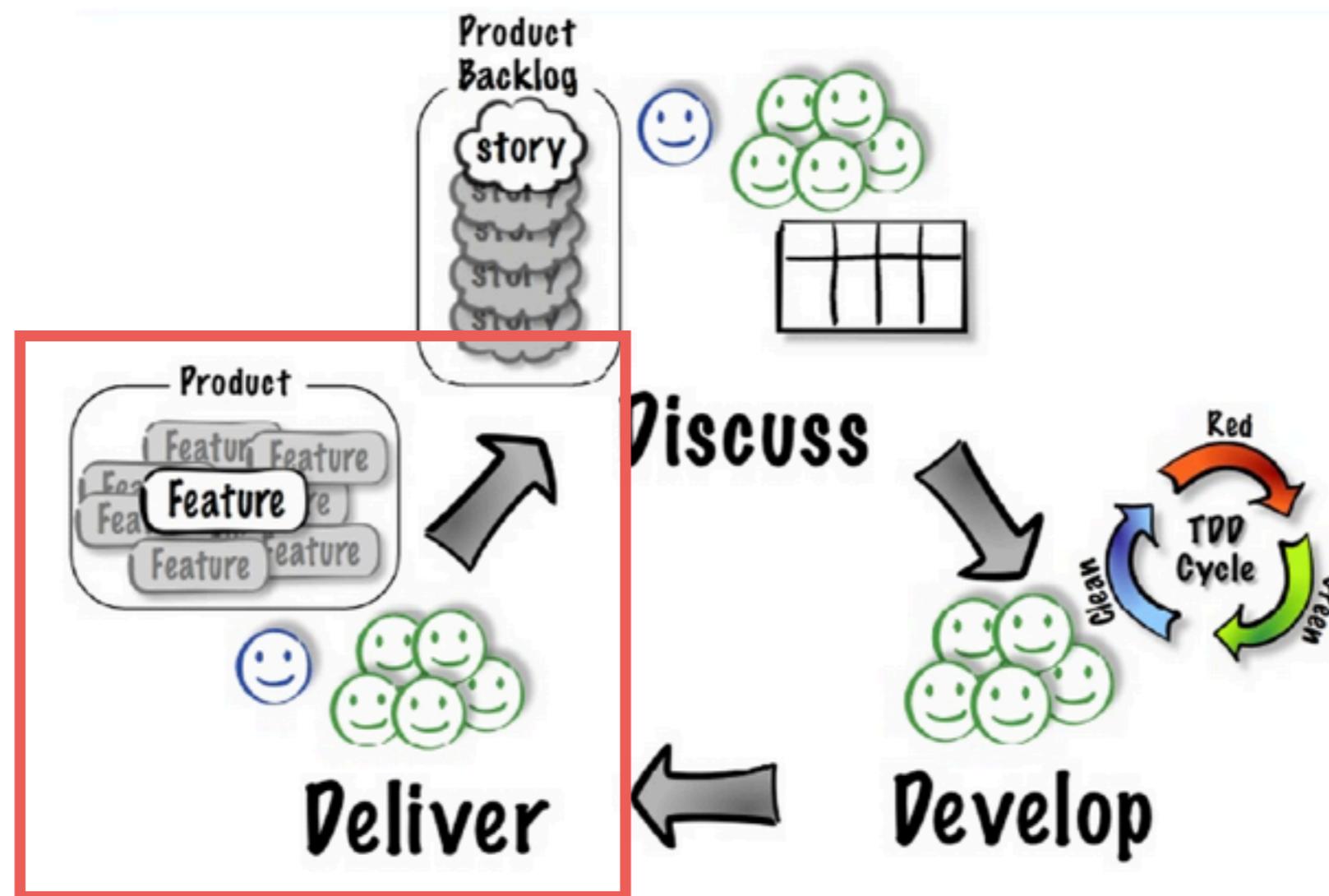


Regression Testing



Acceptance Test-Driven Development (ATDD)

Delivery process



Delivery process

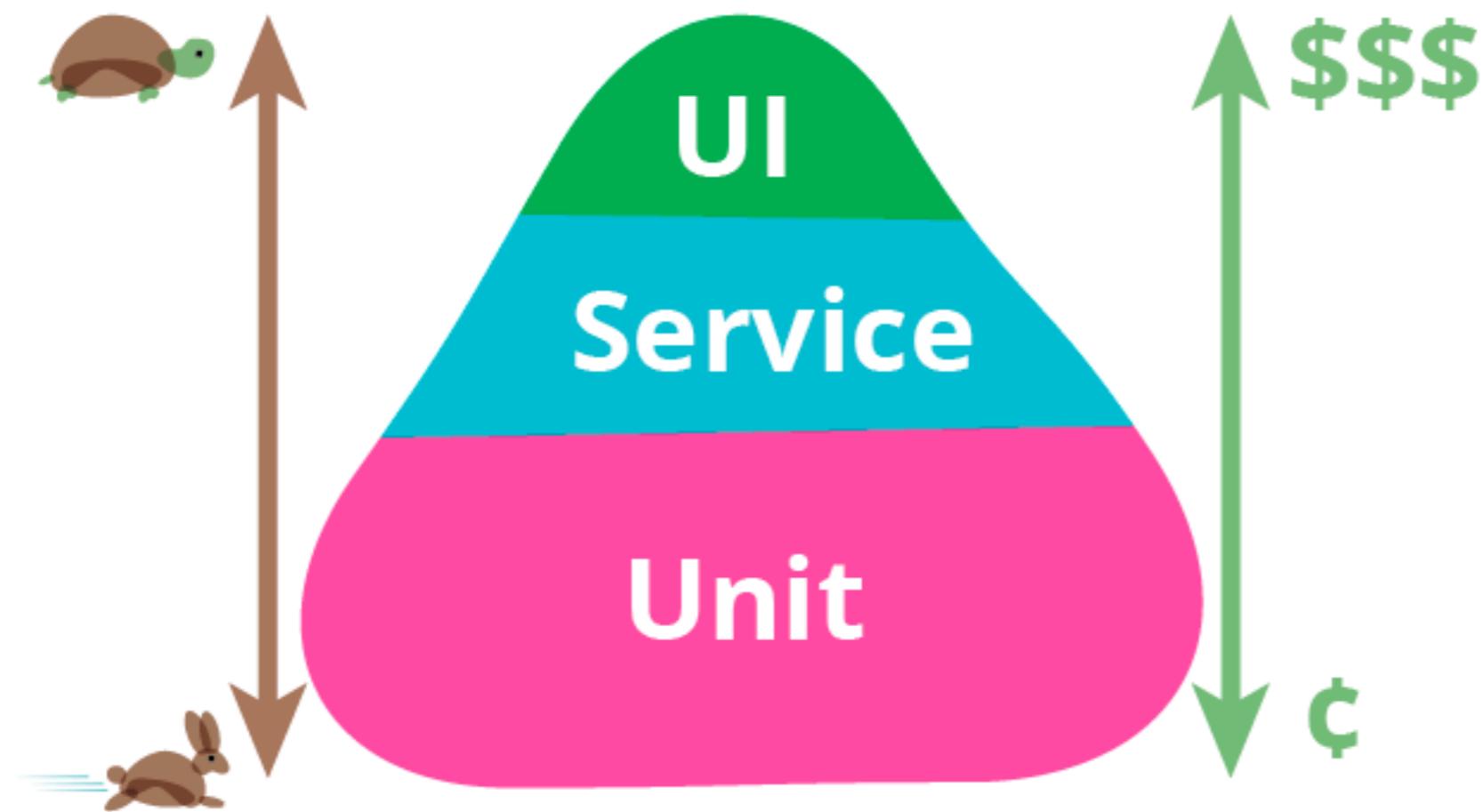
Features are demonstrated to all stakeholders
All examples are passed (new + existed)

Feedback as input to the next discussion



Test Strategies

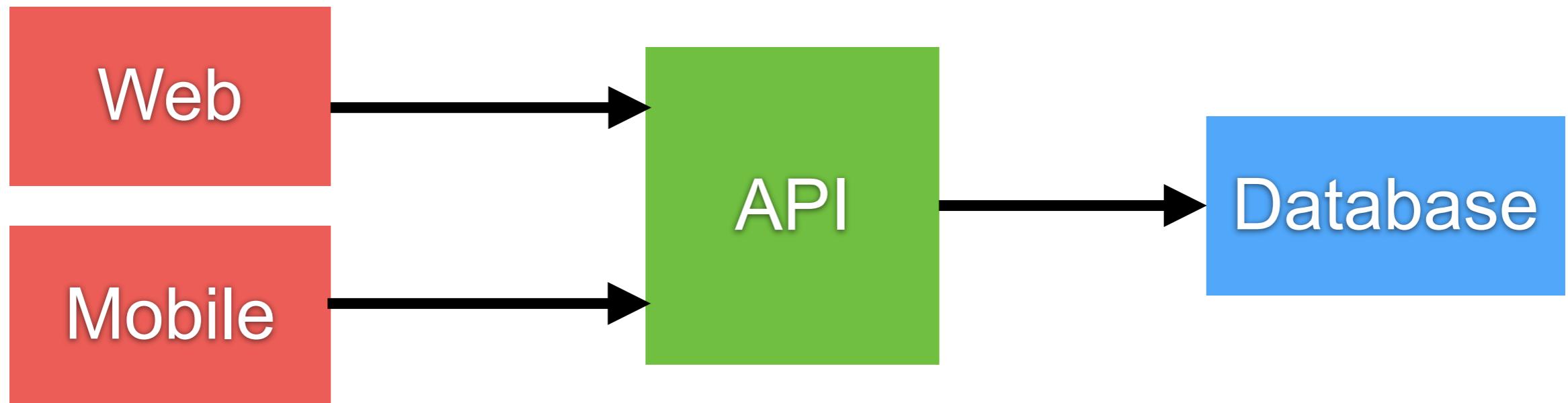




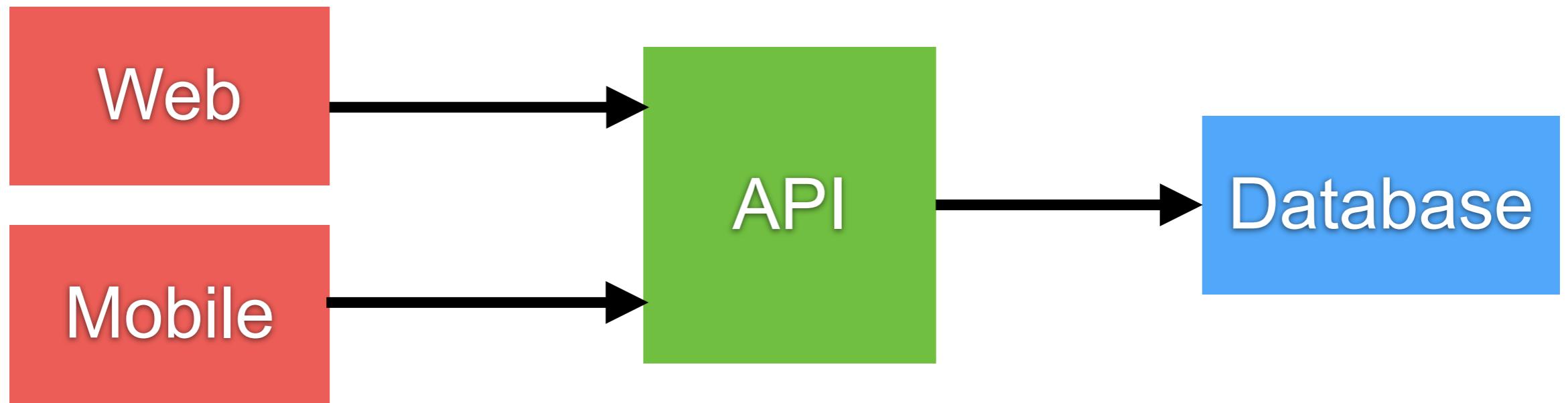
**Why -> What -> How
to test ?**



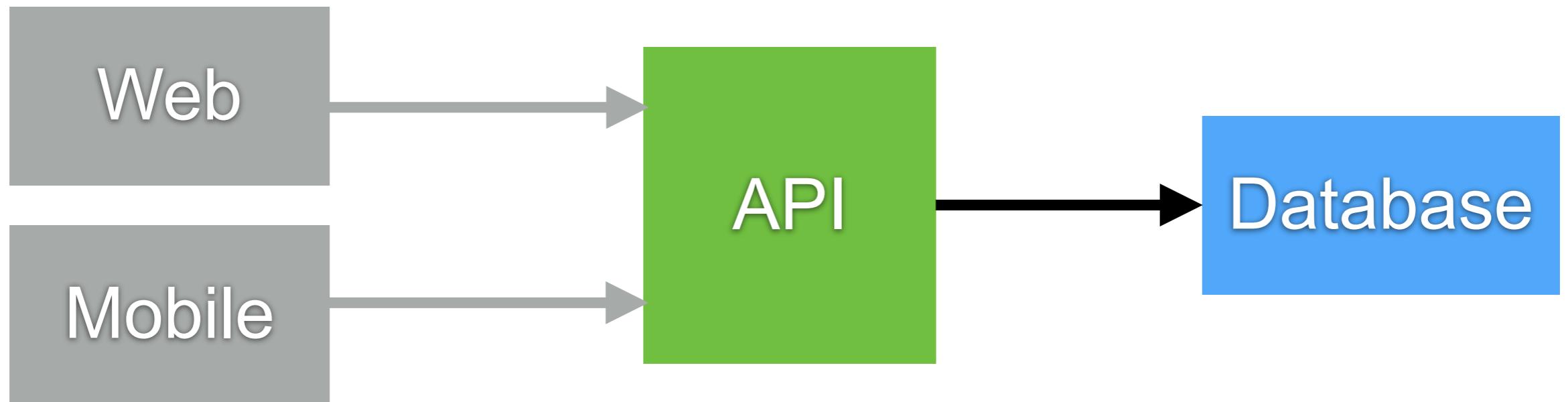
Architecture



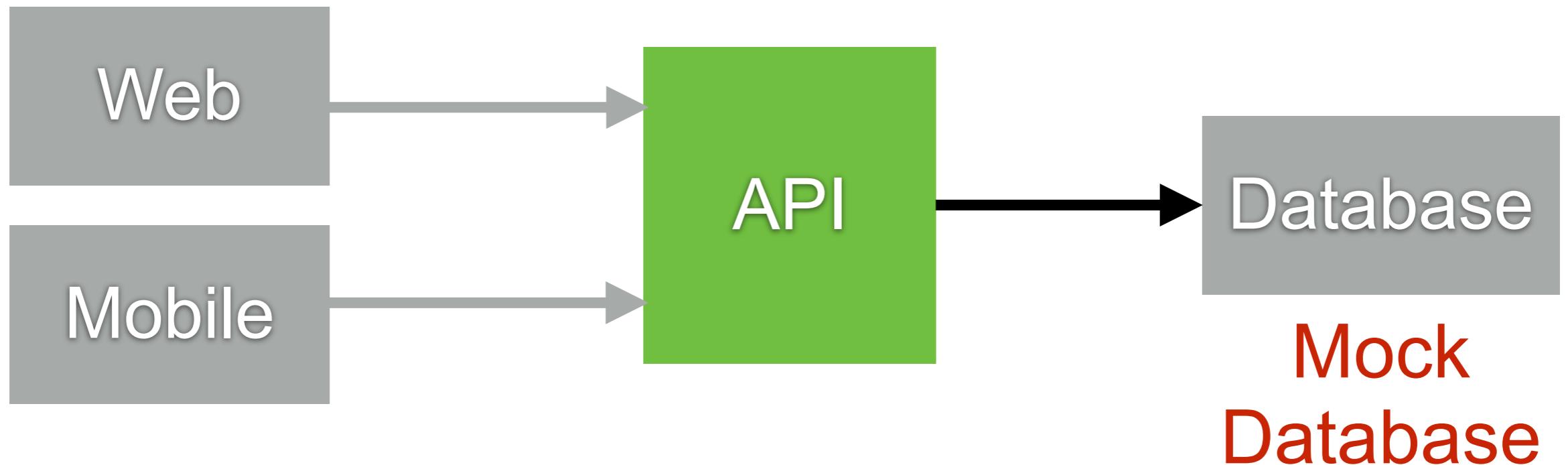
Test ?



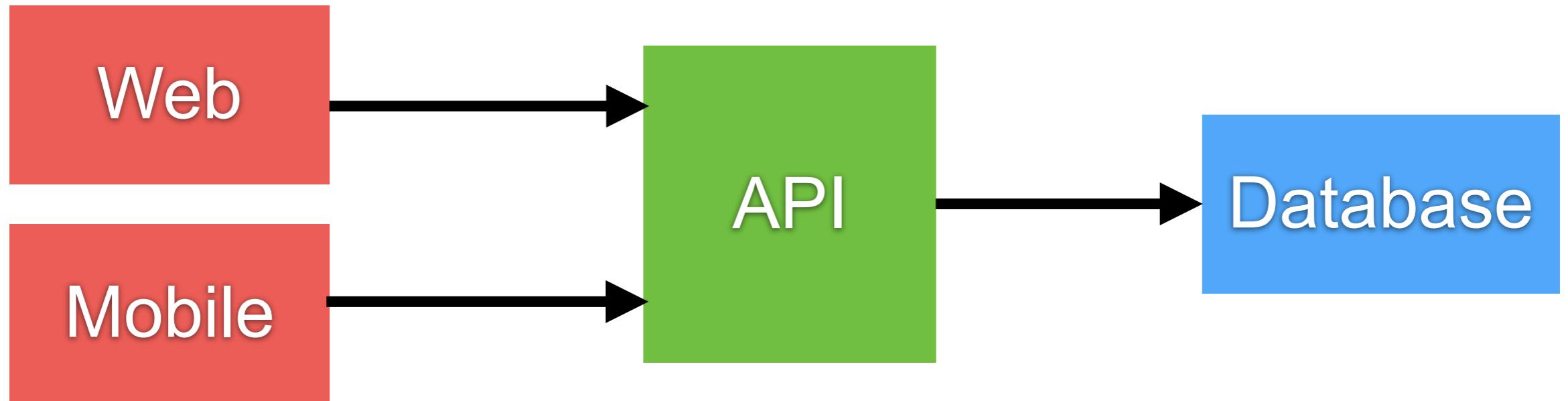
API Testing ?



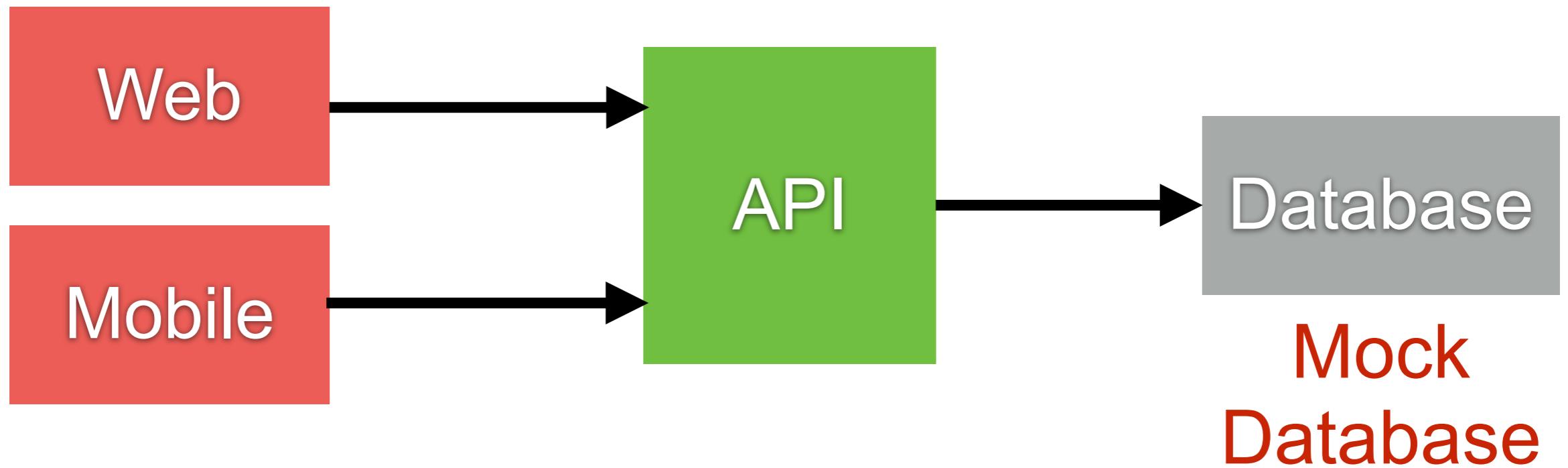
API Testing ?



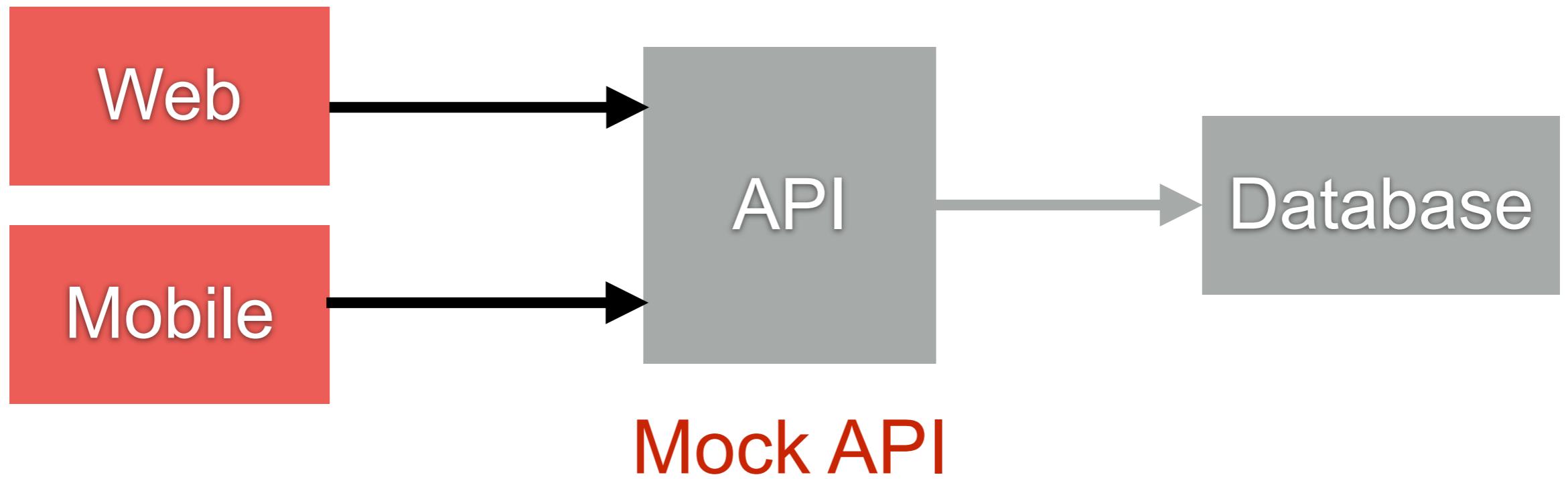
UI Test ?



UI Test ?



UI Test ?



Design Workshop



Example

Login process



Login page

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



The screenshot shows a login form with a light gray background and a thin gray border. At the top center, the text "Login Page" is displayed in a large, bold, black font. Below it, a message in a smaller black font reads: "Please input your user name and password and click the login button." To the left of the first input field, the label "User Name:" is followed by a horizontal input field. To the left of the second input field, the label "Password:" is followed by another horizontal input field. Below these fields is a single-line rectangular button with the word "LOGIN" centered in capital letters.

**username=demo
password=mode**



Result page

Welcome and Error page

Welcome Page

Login succeeded. Now you can [logout](#).

Error Page

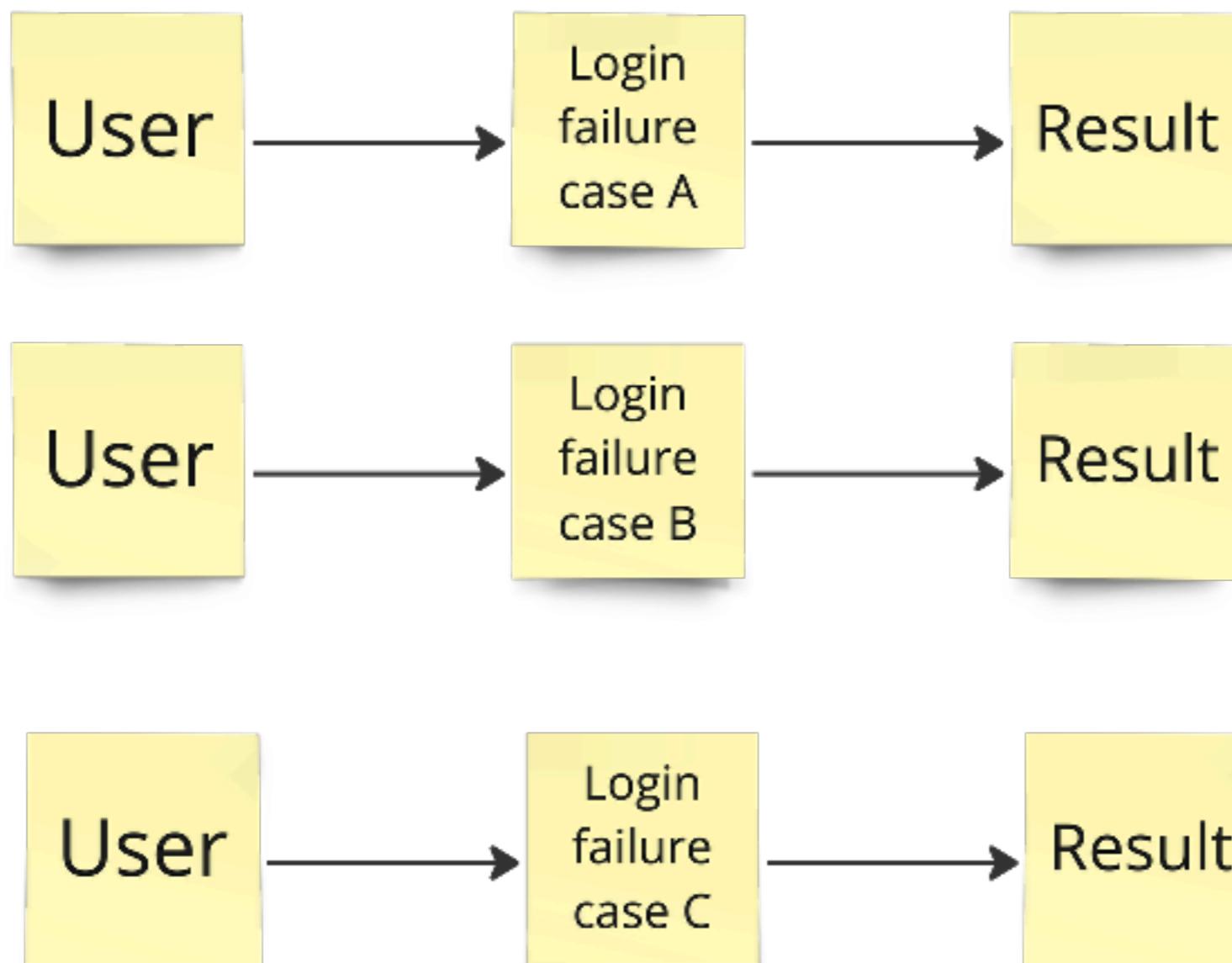
Login failed. Invalid user name and/or password.



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



Testing Tools



Testing Tools



Selenium



ROBOT
FRAME
WORK



appium



Playwright



POSTMAN



Katalon

JUnit 5

xUnit.net

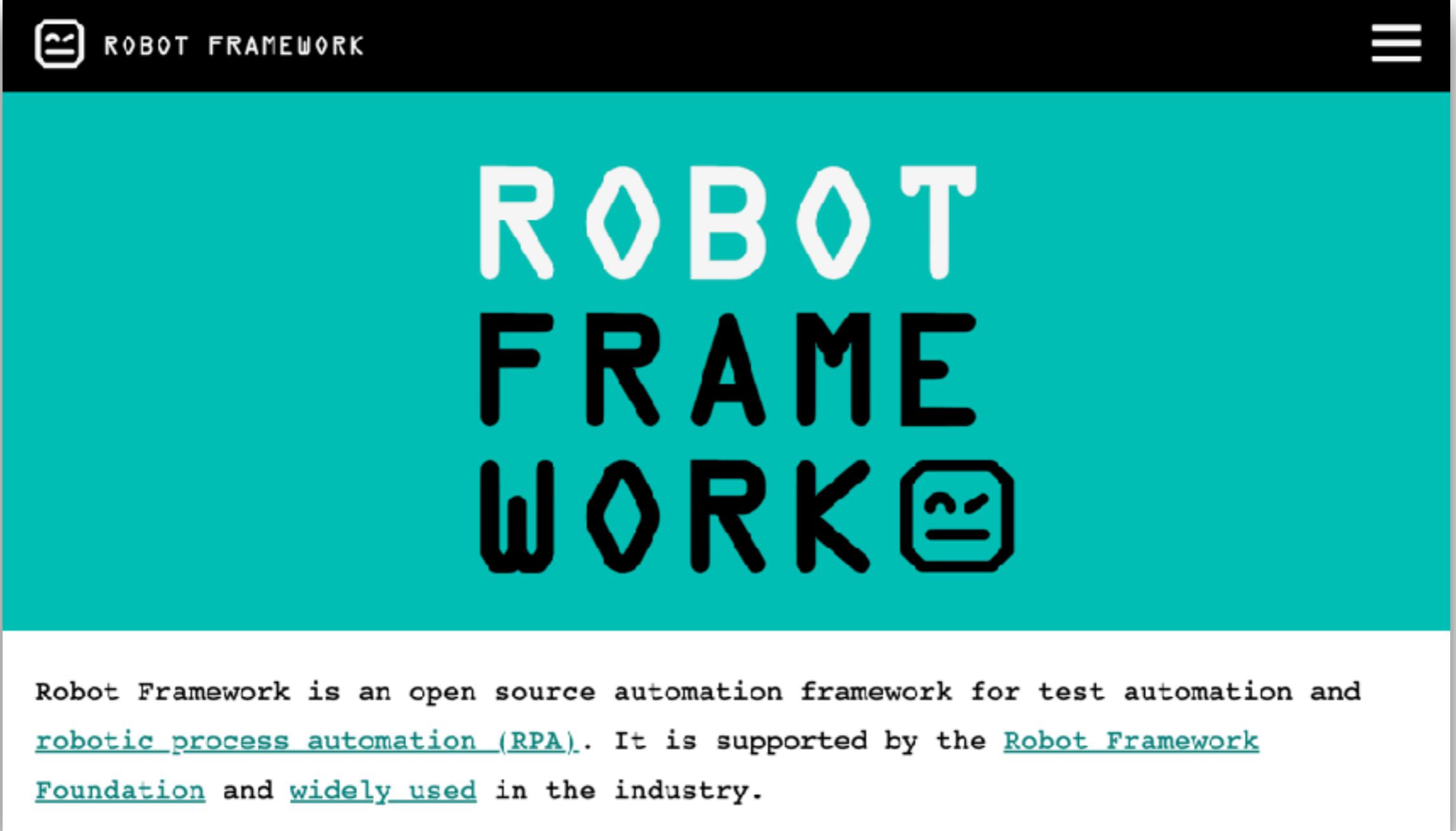




Introduction with Robot Framework



Robot Framework



The screenshot shows the official website for Robot Framework. At the top left is the "ROBOT FRAMEWORK" logo, which consists of a stylized robot head icon followed by the text. At the top right is a three-line menu icon. The main title "ROBOT FRAMEWORK" is centered on a teal background in large, bold, white and black letters. Below the title, a white text box contains a paragraph about the framework's purpose and support.

Robot Framework is an open source automation framework for test automation and [robotic process automation \(RPA\)](#). It is supported by the [Robot Framework Foundation](#) and [widely used](#) in the industry.

<https://robotframework.org/>



Guides for beginner

docs.robotframework.org

Robot Framework Guides

About

Getting Started

Libraries

Examples

Docker And CI Systems

Extending Robot Framework

Re-Execute failed tests

Running tests in parallel

Parsing Test Results

Reporting Test Results

Testcase Styles

Variables

Guides User Guide Standard Library API Documentation Slack GitHub Search

Robot Framework Guides

Welcome to ROBOT FRAMEWORK GUIDES

We hope these guides will help you get started with Robot Framework faster and easier. If you have any questions, please reach out to our awesome community on Slack.

Getting Started

Set up your machine to use Robot Framework

 **Test Automation**
How to set up Robot Framework for testing

 **RPA**
How to set up Robot Framework for Robotic Process Automation (RPA)

 **IDE**
Install and set up your IDE for coding and debugging

<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](#)
- [1.1.3 Screenshots](#)

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



Robot Framework

© 2017 - 2024 Siam Chamnankit Company Limited. All rights reserved.

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.	TOOL

<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 Requirements



Check python

\$python -V

\$pip -V



Install robot framework

\$pip install robotframework

\$pip install robotframework-seleniumlibrary



Check robot framework

\$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

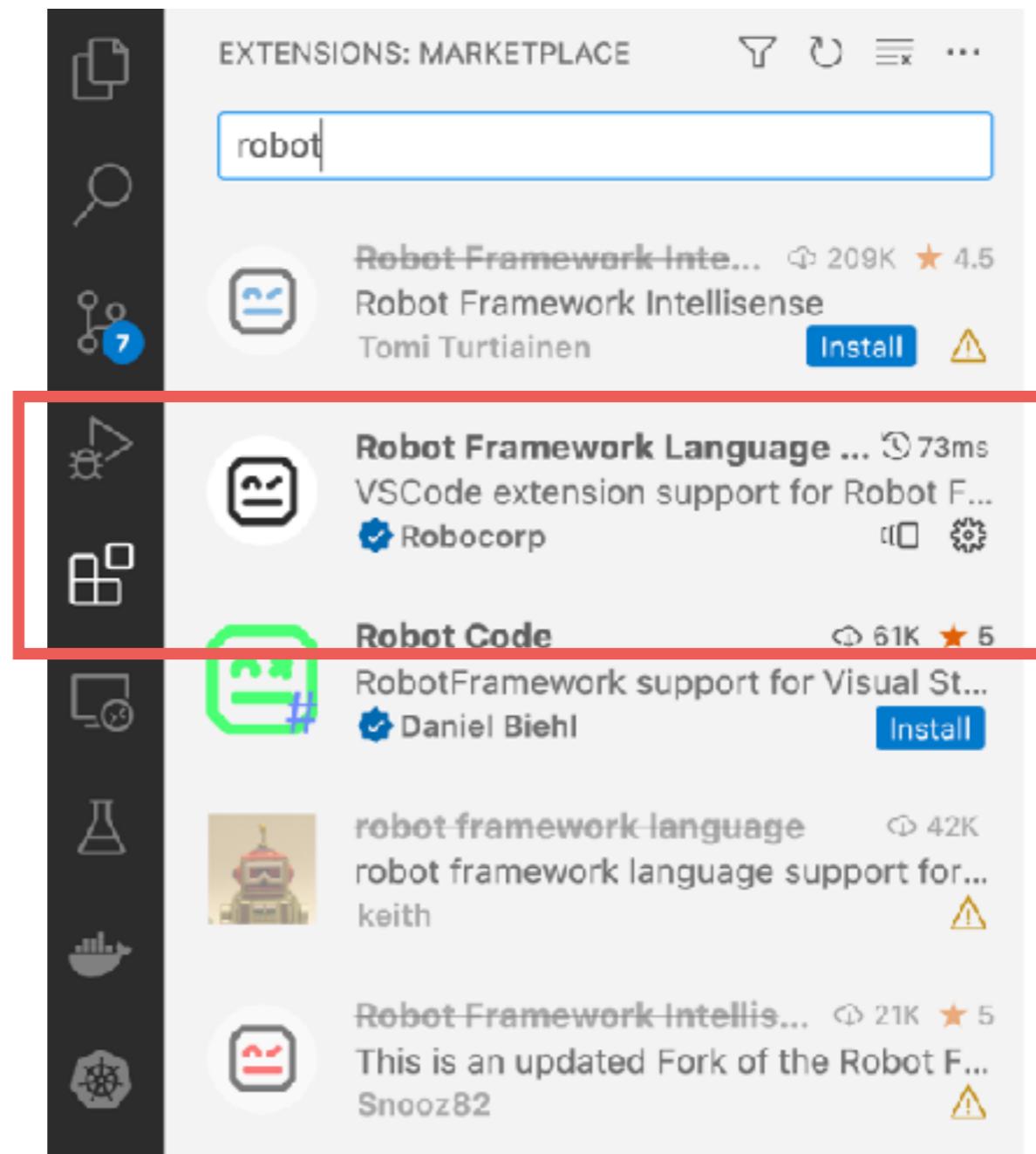
\$robot

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

Try --help for usage information.



VS Code Extension



Let's start



Write First Test Case



Good Test ?

F.I.R.S.T - U

Fast
Isolated
Repeatable
Self-verify
Timely, Thought
Understanding



Test Structure

hello.robot

*** Settings ***

*** Variables ***

*** Test Cases ***

*** Keywords ***

https://docs.robotframework.org/docs/style_guide



Hello Robot Framework

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

*** Variables ***
${URL}           https://www.google.com
${BROWSER}        chrome

*** Test Cases ***
Open Browser
  Open Browser    ${URL}      browser=${BROWSER}
  Maximize Browser Window
  Close Browser
```



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>



Start with 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>



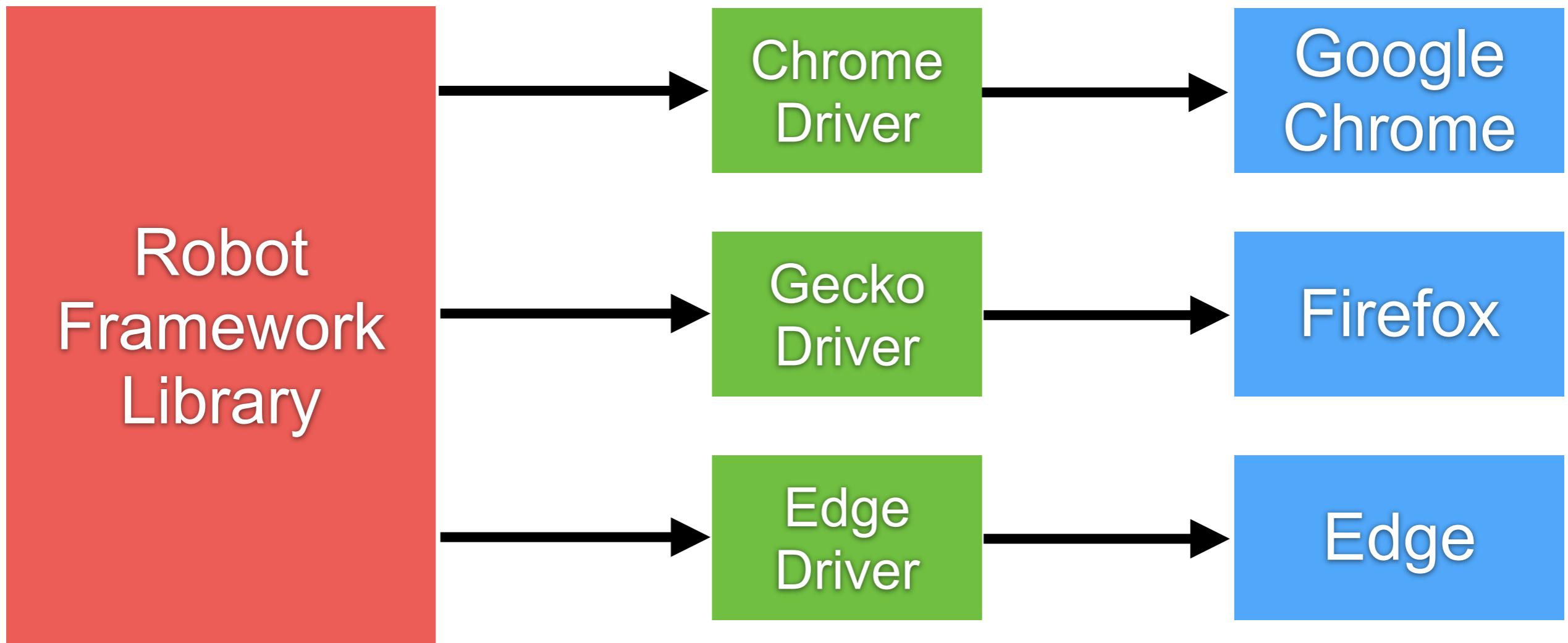
Installation

```
$pip install -U robotframework-seleniumlibrary
```

```
$pip list
```



Selenium Library



<https://github.com/robotframework/SeleniumLibrary>



Selenium Library Keywords

SeleniumLibrary

Search X

Keywords (177)

- Add Cookie
- Add Location Strategy
- Alert Should Be Present
- Alert Should Not Be Present
- Assign Id To Element
- Capture Element Screenshot
- Capture Page Screenshot
- Checkbox Should Be Selected
- Checkbox Should Not Be Selected
- Choose File
- Clear Element Text
- Click Button
- Click Element

Library version: 6.1.0
Library scope: GLOBAL

Introduction

SeleniumLibrary is a web testing library for Robot Framework.

This document explains how to use keywords provided by SeleniumLibrary. For information about SeleniumLibrary, visit the [project pages](#). For more information about Robot Framework, see [http://robotframework.org](#).

SeleniumLibrary uses the Selenium WebDriver modules internally to control a web browser. For information about Selenium in general and SeleniumLibrary README.rst [Browser drivers chapter](#).

- [Locating elements](#)
- [Browser and Window](#)
- [Timeouts, waits, and delays](#)
- [Run-on-failure functionality](#)
- [Boolean arguments](#)
- [EventFiringWebDriver](#)
- [Thread support](#)
- [Plugins](#)
- [Importing](#)
- [Keywords](#)

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



Selenium

The screenshot shows the official Selenium website. At the top left is the Selenium logo (a green 'Se' icon). At the top right is a three-line menu icon. A blue header bar contains the text "Appium Conf 2024 Call for Proposals is now open! Submissions close 28 July. [Learn more & submit](#)". The main content area has a green background. It features the text "Selenium automates browsers. That's it!" in large white font, followed by "What you do with that power is entirely up to you." in white font. Below this, in smaller white font, is the text "Primarily it is for automating web applications for testing purposes, but is certainly not limited to just that. Boring web-based administration tasks can (and should) also be automated as well." A URL "https://www.selenium.dev/" is displayed at the bottom of the slide.

Selenium automates browsers. That's it!

What you do with that power is entirely up to you.

Primarily it is for automating web applications for testing purposes, but is certainly not limited to just that.
Boring web-based administration tasks can (and should) also be automated as well.

<https://www.selenium.dev/>



Selenium projects

Getting Started



Selenium WebDriver

If you want to create robust, browser-based regression automation suites and tests, scale and distribute scripts across many environments, then you want to use Selenium WebDriver, a collection of language specific bindings to drive a browser - the way it is meant to be driven.

[READ MORE ▶](#)



Selenium IDE

If you want to create quick bug reproduction scripts, create scripts to aid in automation-aided exploratory testing, then you want to use Selenium IDE; a Chrome, Firefox and Edge add-on that will do simple record-and-playback of interactions with the browser.

[READ MORE ▶](#)



Selenium Grid

If you want to scale by distributing and running tests on several machines and manage multiple environments from a central point, making it easy to run the tests against a vast combination of browsers/OS, then you want to use Selenium Grid.

[READ MORE ▶](#)

<https://www.selenium.dev/>



Robot Framework

© 2017 - 2024 Siam Chamnankit Company Limited. All rights reserved.

Selenium projects

**Selenium
WebDriver**

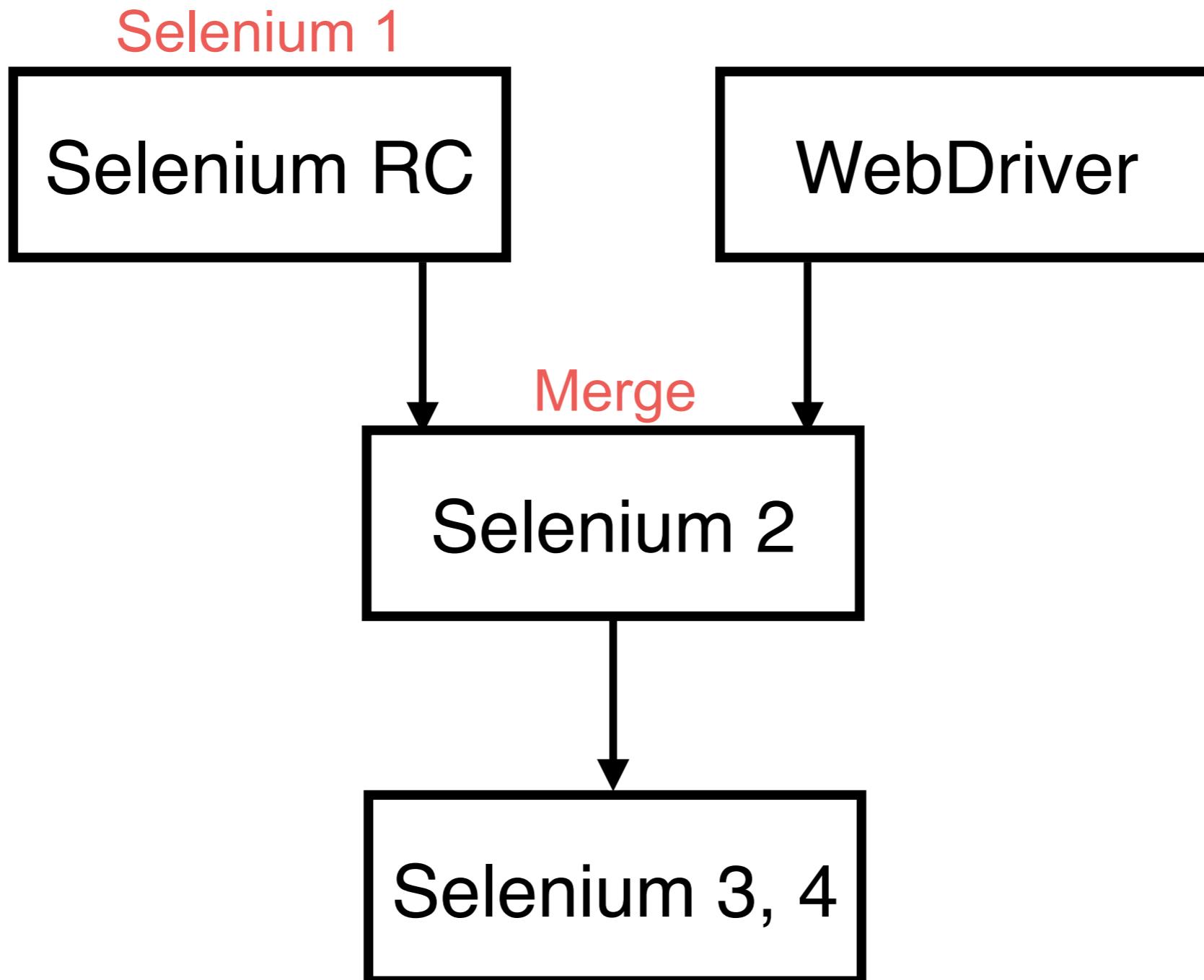
**Selenium
IDE**

**Selenium
Grid**

<https://www.selenium.dev/>



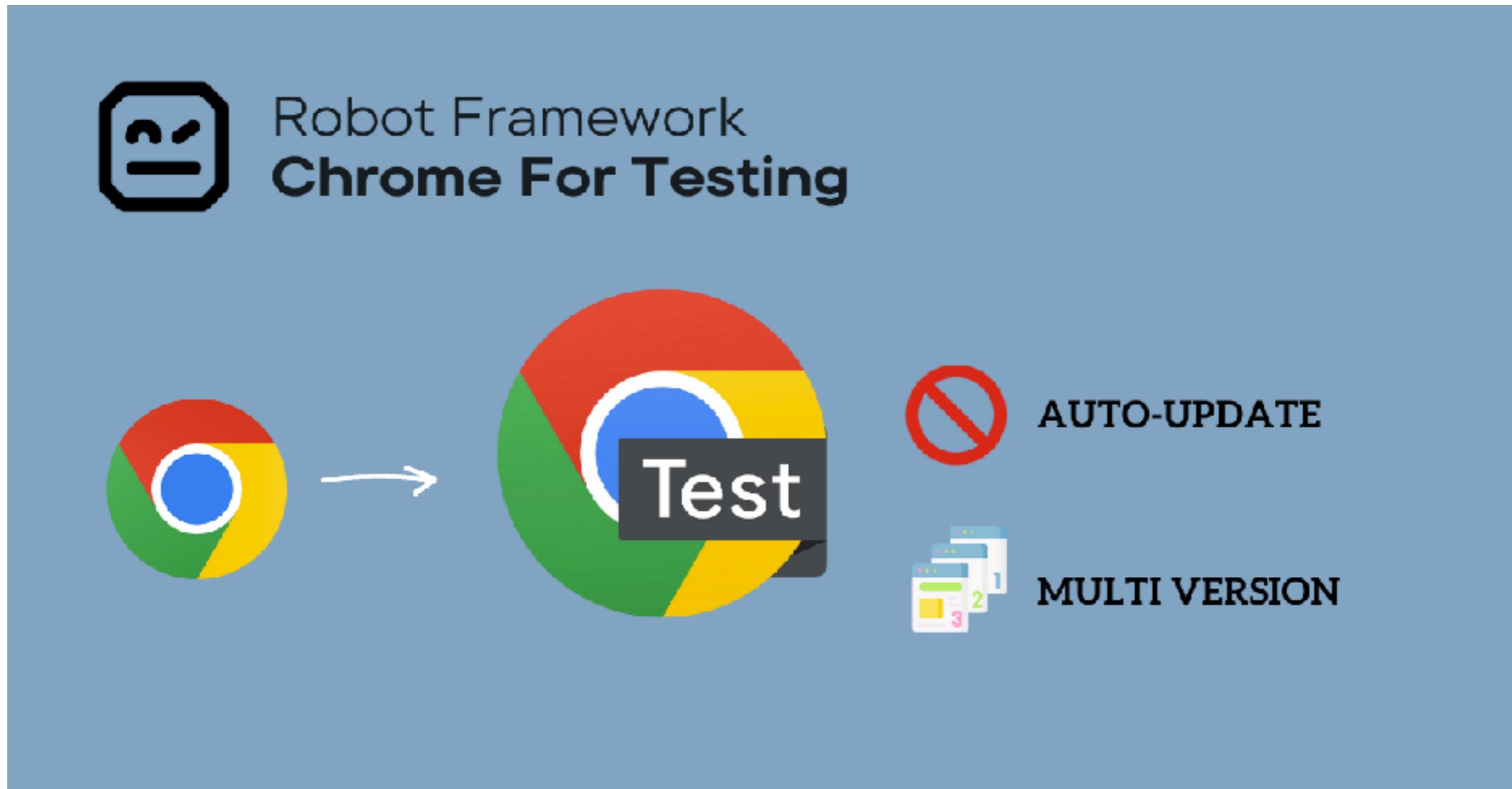
Selenium WebDriver



<https://www.selenium.dev/>



Chrome for Testing



<https://googlechromelabs.github.io/chrome-for-testing/>



Working with Chrome for Testing

*** Settings ***

Library SeleniumLibrary

*** Variables ***

`${URL}` https://www.google.com
 `${BROWSER}` chrome

`${CHROME_FOR_TEST}` \${CURDIR}/chrome-mac-arm64

*** Test Cases ***

Open Browser

`Open Browser ${URL} browser=${BROWSER}`
`... options=binary_location="${CHROME_FOR_TEST}"`

Maximize Browser Window

Close Browser



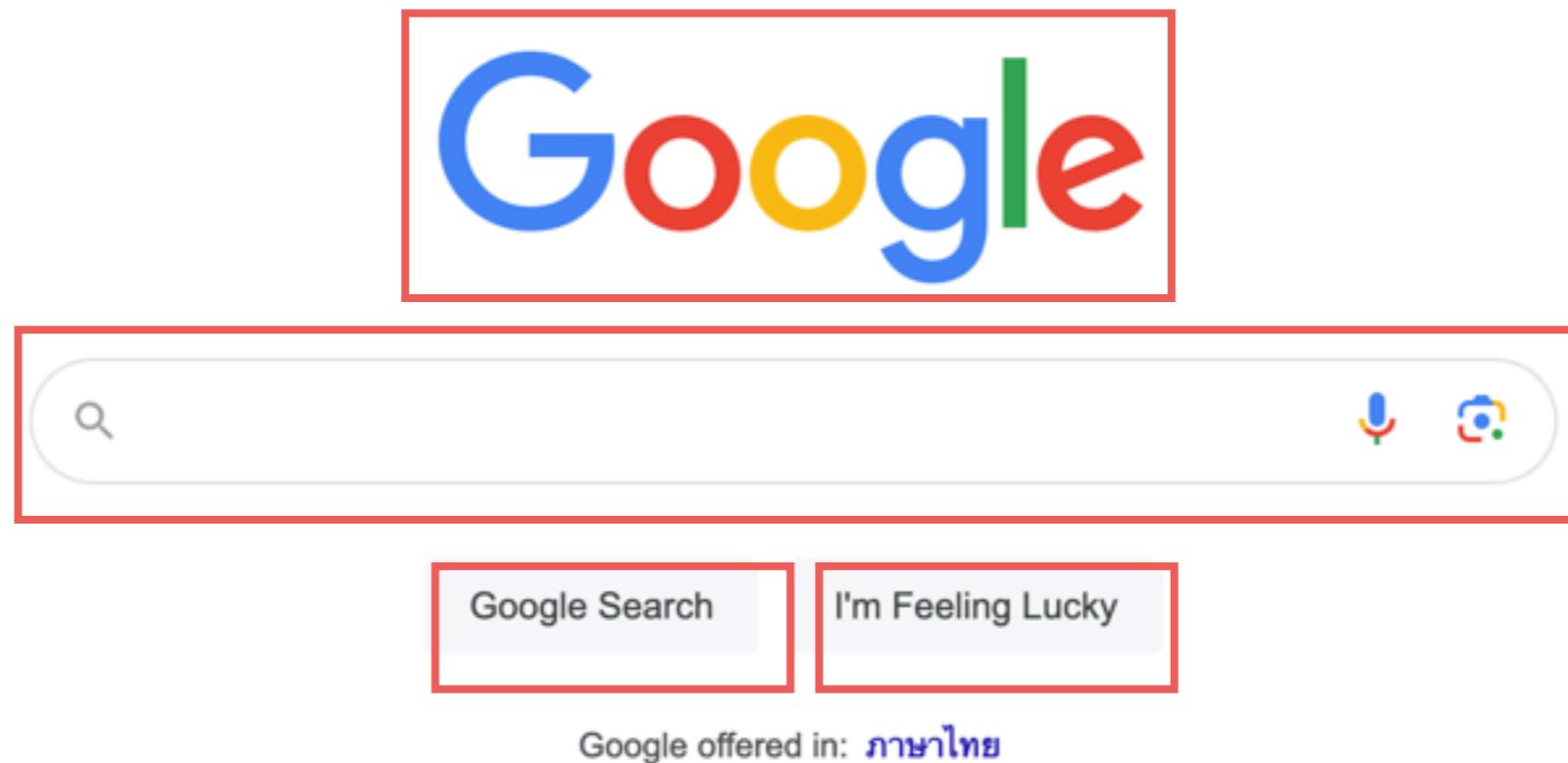
Workshop with First test case

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

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



Bad locator === Flaky test

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

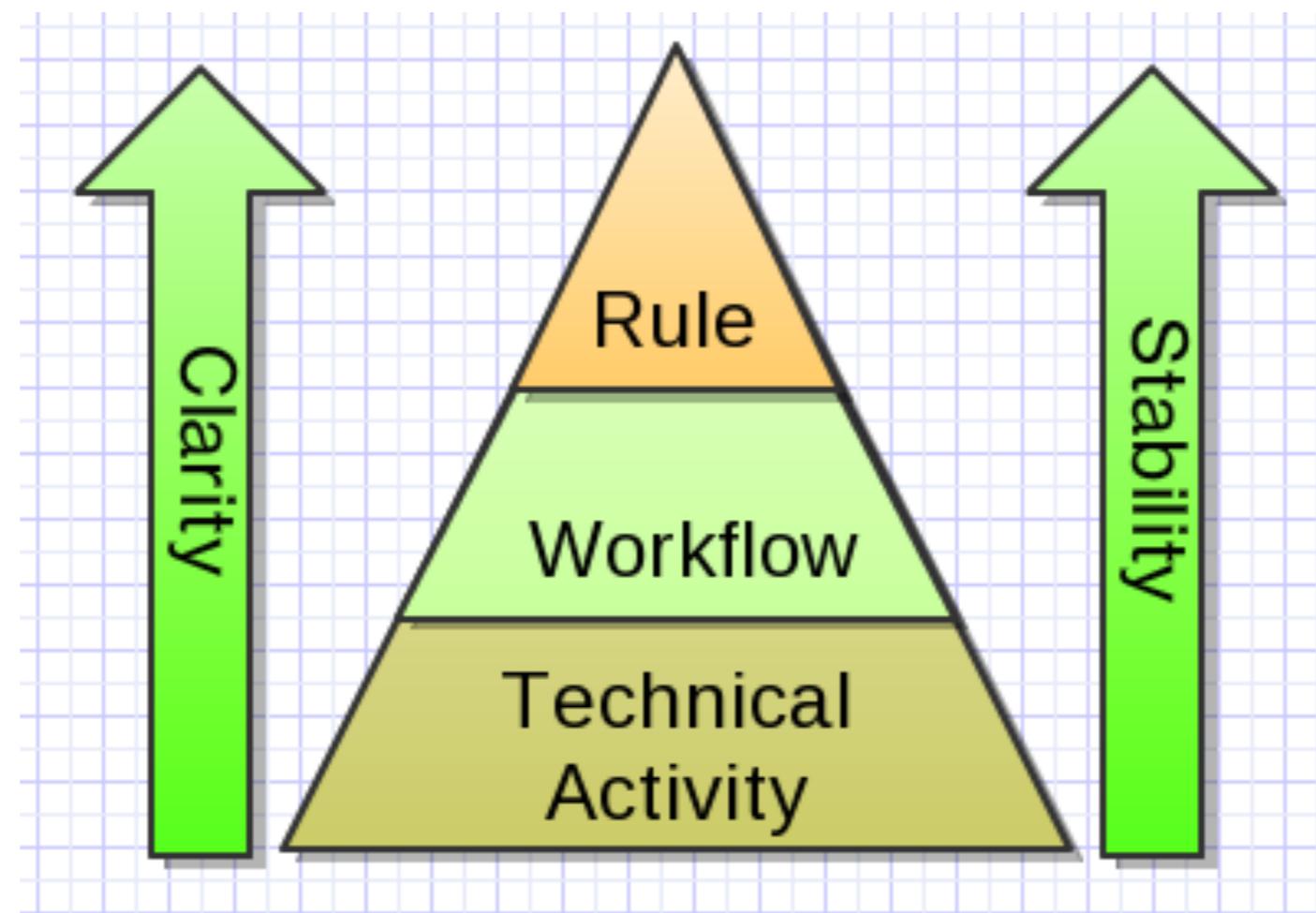


Improve your test case

More understanding



3 levels of UI Automation Test



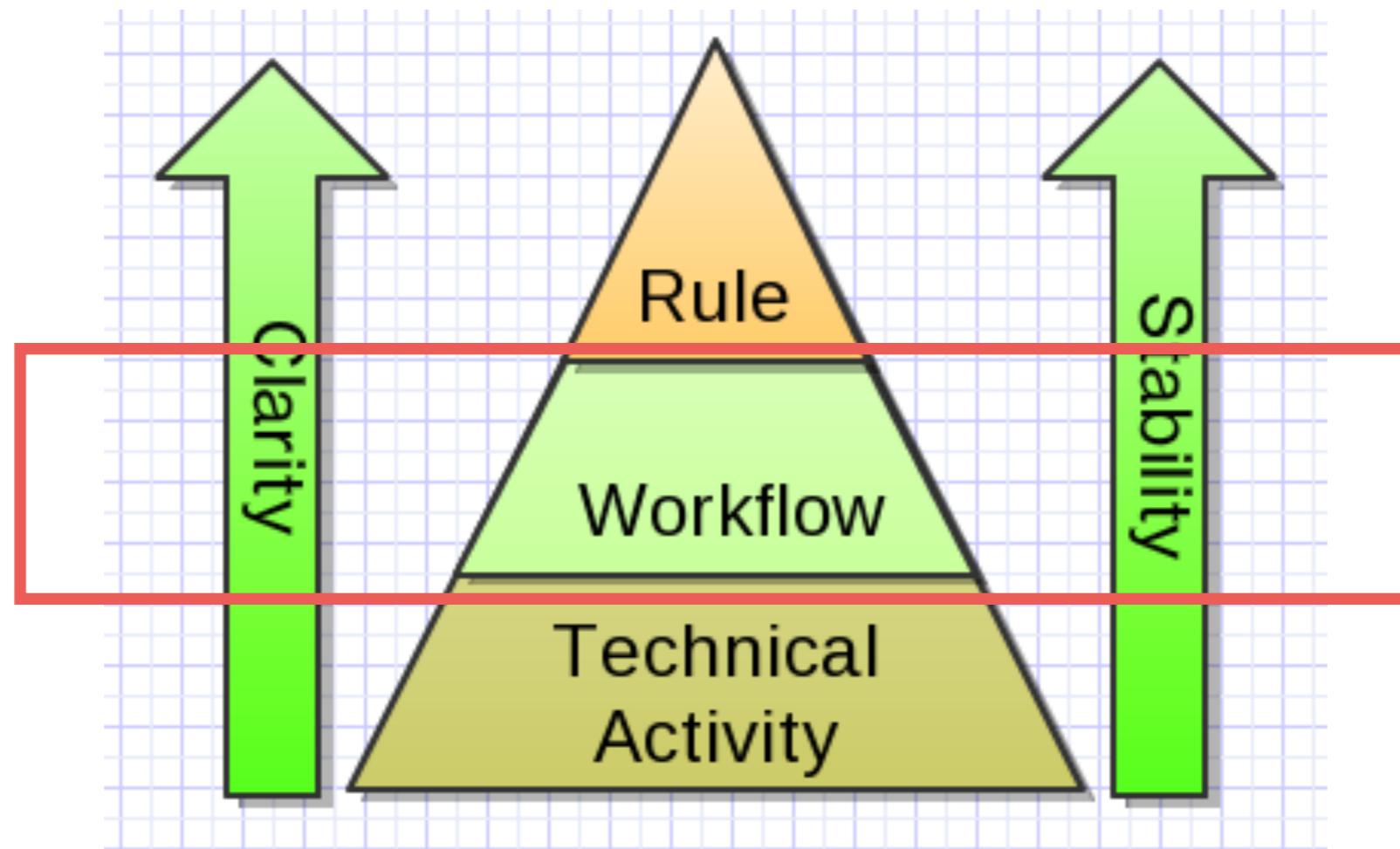
<https://gojko.net/2010/04/13/how-to-implement-ui-testing-without-shooting-yourself-in-the-foot-2/>



Robot Framework

© 2017 - 2024 Siam Chamnankit Company Limited. All rights reserved.

User Interface workflow



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



Robot Framework

© 2017 - 2024 Siam Chamnankit Company Limited. All rights reserved.

Recording Tools



Recording Tools

Selenium IDE
Google Chrome Recorder



Selenium IDE

 Selenium IDE

Docs API Plugins Blog Help

Selenium IDE

Open source record and playback test automation for the web

[CHROME DOWNLOAD](#) [FIREFOX DOWNLOAD](#) [LATEST ZIP](#)

 Star 2,392



Web Ready

Simple, turn-key solution to quickly author reliable end-to-end tests. Works out of the box for any web app.



Easy Debugging

Enjoy easier test debugging with rich IDE features like setting breakpoints and pausing on exceptions.



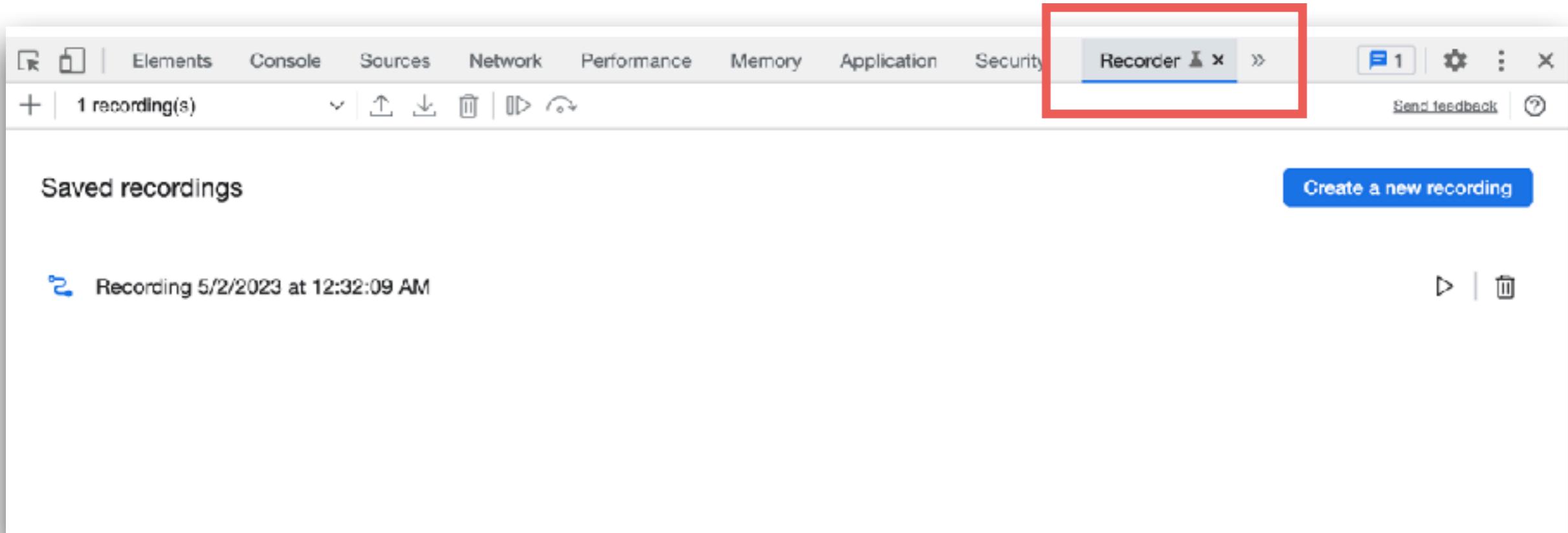
Cross-browser Execution

Run your tests on any browser/OS combination in parallel using the Command-line Runner for 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>



Robot Framework

© 2017 - 2024 Siam Chamnankit Company Limited. All rights reserved.

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

```
$srobot -v URL:new -v BROWSER:new
```



Test Life Cycle



Testing life cycle

Test case 01

Test case 02



Testing life cycle

Suite Setup

Test case 01

Test case 02

Suite Teardown



Testing life cycle

Suite Setup

Test Setup

Test case 01

Test Teardown

Test case 02

Suite Teardown



Testing life cycle

Suite Setup

Test Setup

Test case 01

Test Teardown

Test Setup

Test case 02

Test Teardown

Suite Teardown



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

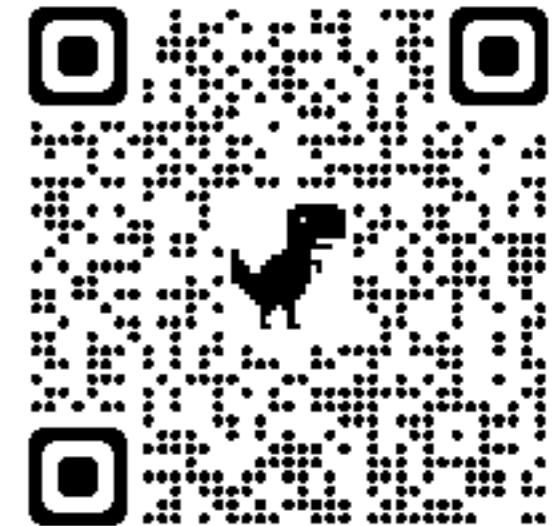
Extension with **.resource** or **.robot**

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

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



More Workshop



Angular File Upload Demos ▾ [View on Github](#) [Download](#)

Select files

Base drop zone

Another drop zone with its own settings

Multiple
 No file chosen

Single
 No file chosen

Upload queue

Queue length: 0

Name	Size	Progress	Status
Queue progress:			

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



Page Object Patterns

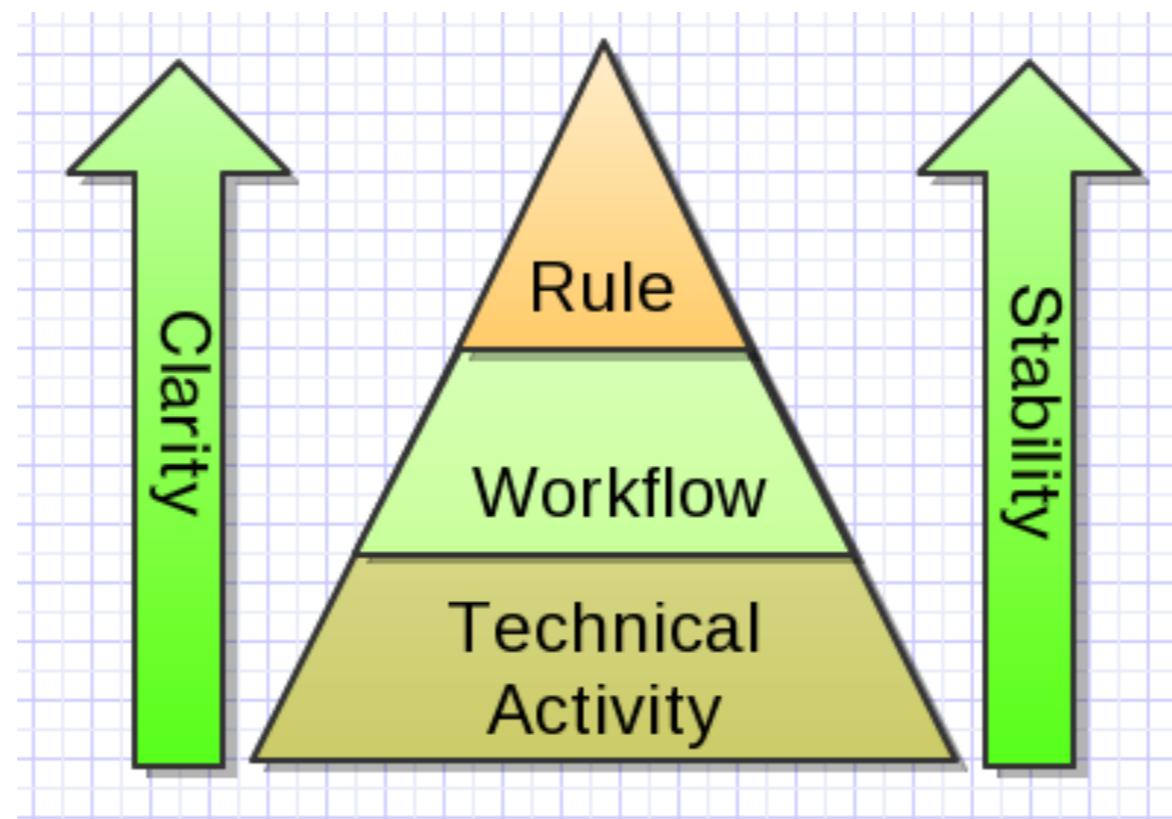


Page Object Patterns

Hide detail/implementation

Reuse keywords

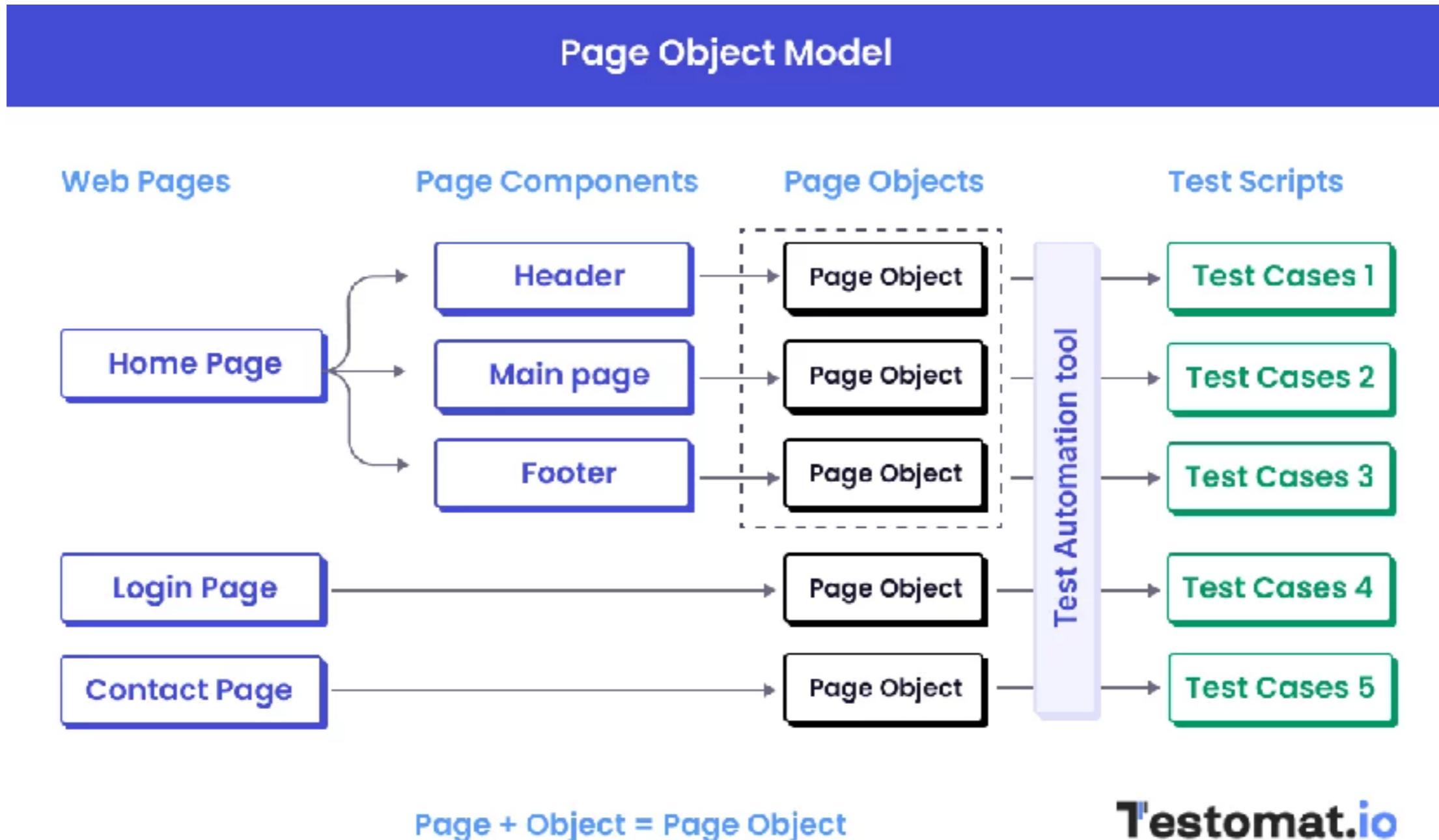
Grouping keywords per page



https://www.selenium.dev/documentation/test_practices/encouraged/page_object_models/



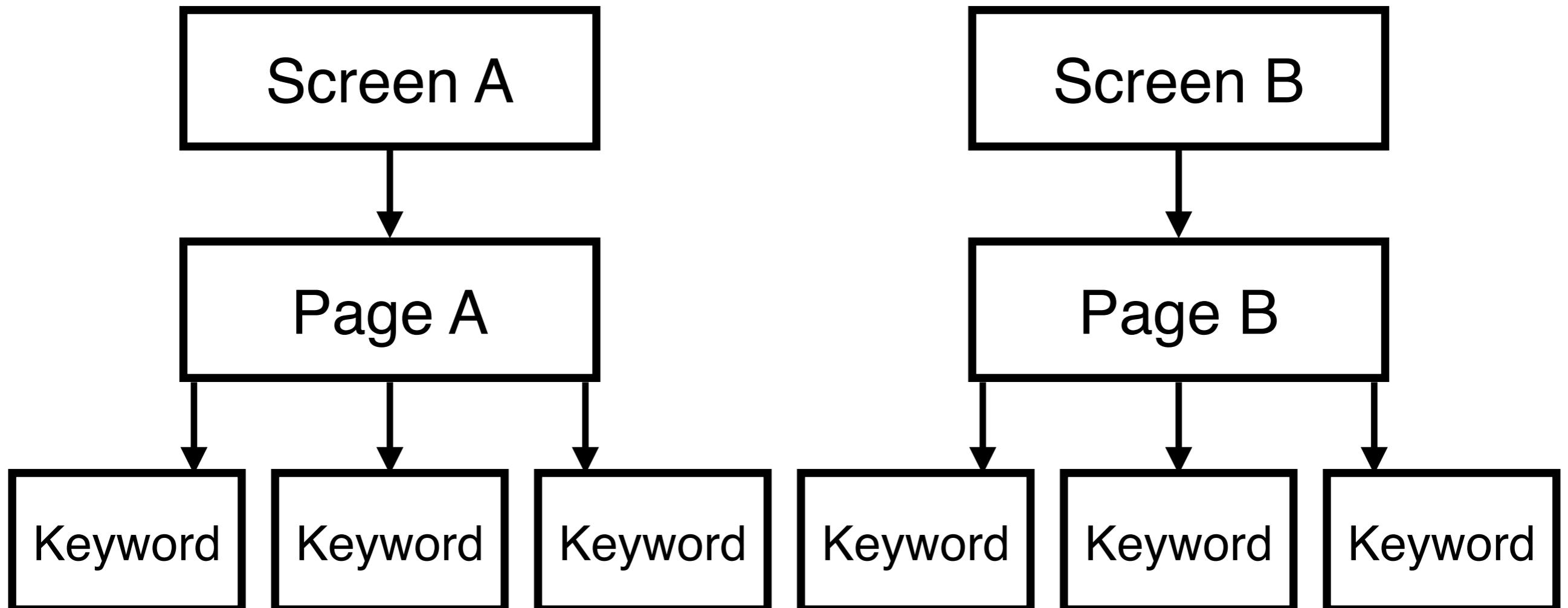
Page Object Patterns



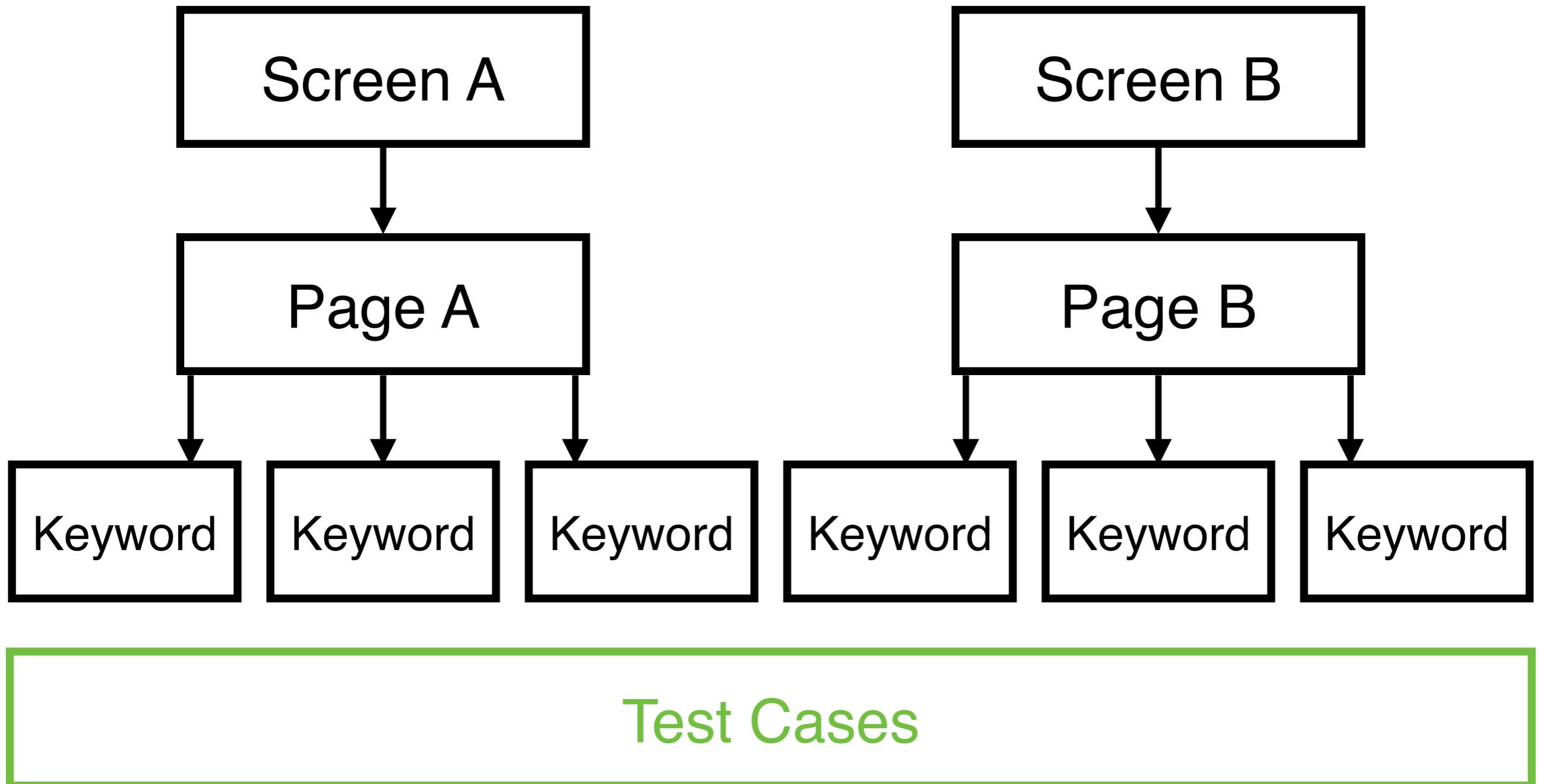
<https://testomat.io/blog/page-object-model-pattern-javascript-with-playwright/>



Page Object Patterns



Page Object Patterns



Manage Dependencies

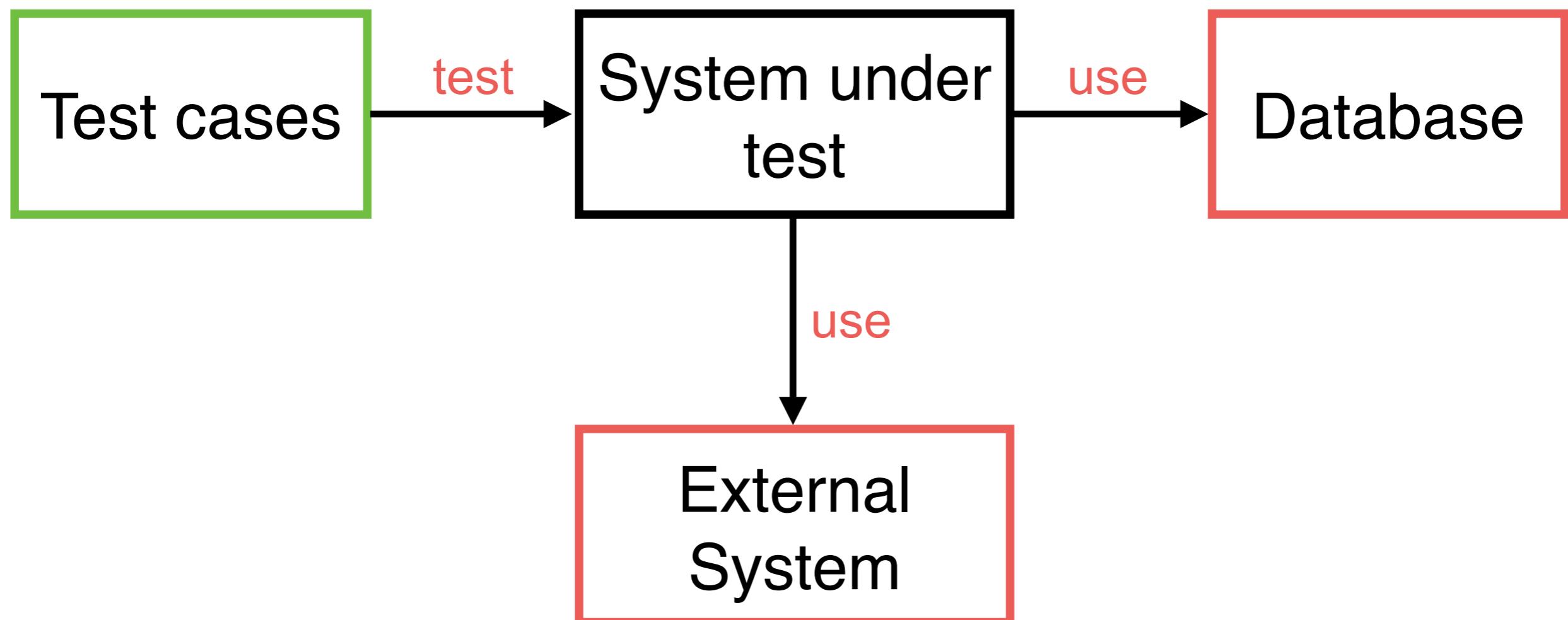


Good Test ?

Fast
Isolated
Repeatable
Self-verify
Timely, Thought
Understanding



Dependencies on system



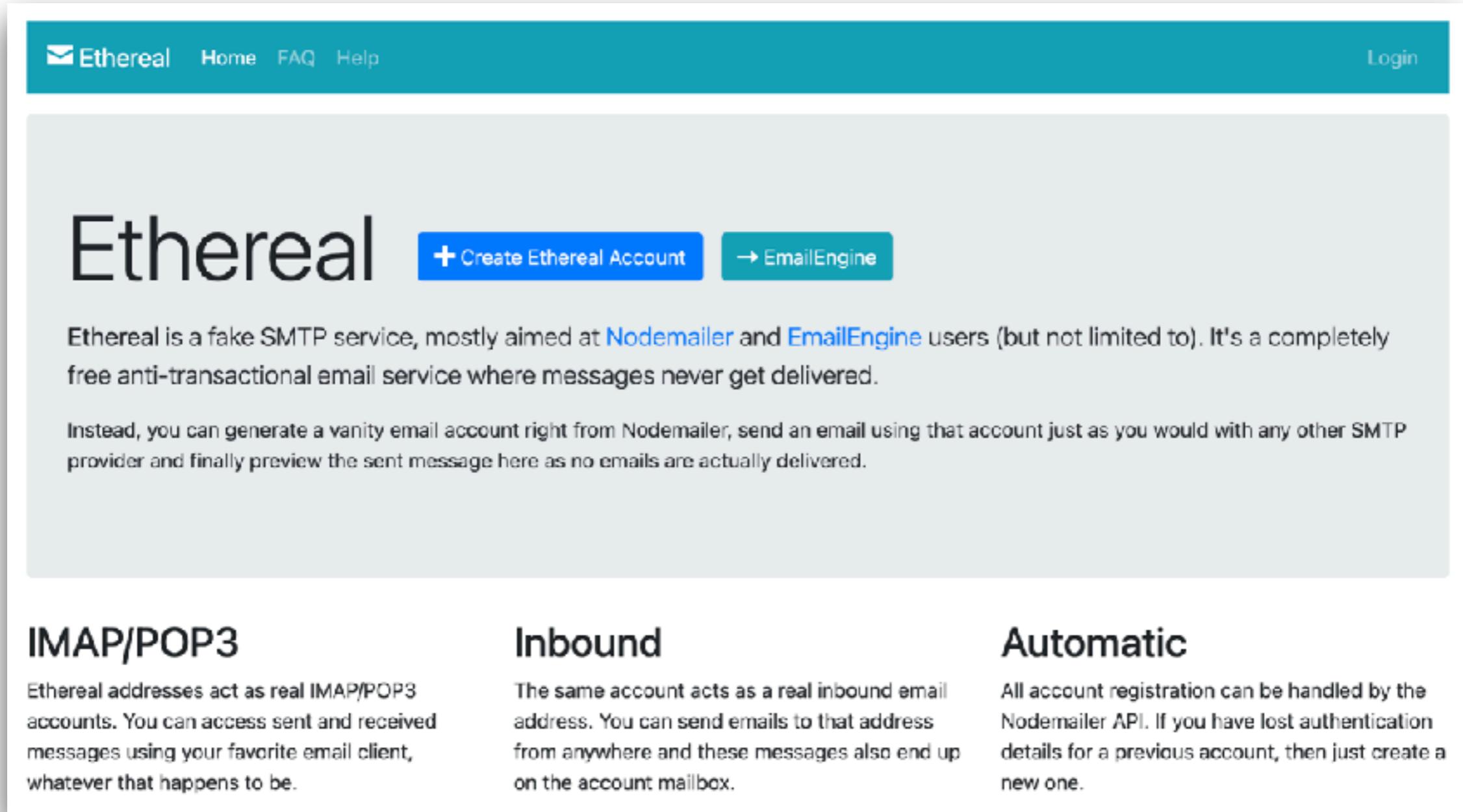
Working with Email

The screenshot shows the Mailosaur homepage with a dark background. At the top, there's a navigation bar with links for Products, Customers, Pricing, Docs, Company, Talk to sales, Sign in, and Start for free. Below the navigation, a large green header reads "Email & SMS testing for any app, product or campaign". A subtext below it says "Improve quality and increase confidence with the number one customer communication testing platform." To the right of the text is a graphic of a smartphone displaying an email interface with various annotations. Annotations include "Logo resolution", "Locks great on:", "Gmail", "iOS", "Outlook", "Correct name", "Hi Jane", "Your promo code is XY672", "Use your promotion", "Working promo code", "Your verification code is 7817. Do not share.", "Valid one-time password", and "Promo code" with a numeric input field showing "XY672". At the bottom left is a "Enter your email" input field and a purple "Try it free" button.

<https://mailosaur.com/>



Working with Email



The screenshot shows the homepage of the Ethereal website. At the top, there is a teal header bar with the Ethereal logo, a Home link, a FAQ link, a Help link, and a Login button. Below the header, the word "Ethereal" is prominently displayed in large black letters. To the right of "Ethereal" are two buttons: a blue one labeled "+ Create Ethereal Account" and a teal one labeled "→ EmailEngine". A descriptive text block follows, stating that Ethereal is a fake SMTP service aimed at Nodemailer and EmailEngine users, providing a free anti-transactional email service where messages never get delivered. It explains that users can generate vanity email accounts from Nodemailer, send emails using them, and preview the sent messages here. Below this text, there are three sections: "IMAP/POP3", "Inbound", and "Automatic", each with a brief description.

Ethereal

+ Create Ethereal Account → EmailEngine

Ethereal is a fake SMTP service, mostly aimed at [Nodemailer](#) and [EmailEngine](#) users (but not limited to). It's a completely free anti-transactional email service where messages never get delivered.

Instead, you can generate a vanity email account right from Nodemailer, send an email using that account just as you would with any other SMTP provider and finally preview the sent message here as no emails are actually delivered.

IMAP/POP3

Ethereal addresses act as real IMAP/POP3 accounts. You can access sent and received messages using your favorite email client, whatever that happens to be.

Inbound

The same account acts as a real inbound email address. You can send emails to that address from anywhere and these messages also end up on the account mailbox.

Automatic

All account registration can be handled by the Nodemailer API. If you have lost authentication details for a previous account, then just create a new one.

<https://ethereal.email/>



Working with Data files

CSV

Excel

JSON

<https://github.com/Snooz82/robotframework-datariver>



RPA framework

Collection of open source libraries

Email

Excel

HTTP

FTP

MFA

OpenAI

PDF

SAP

Database

<https://rpaframework.org/>



Workshop



Scaling Testing



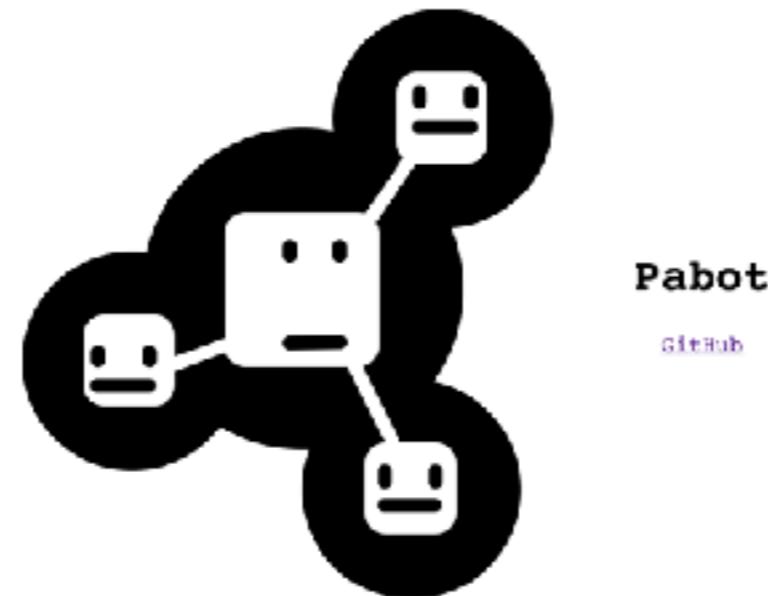
Scaling Testing

- Pabot
- Selenium Grid

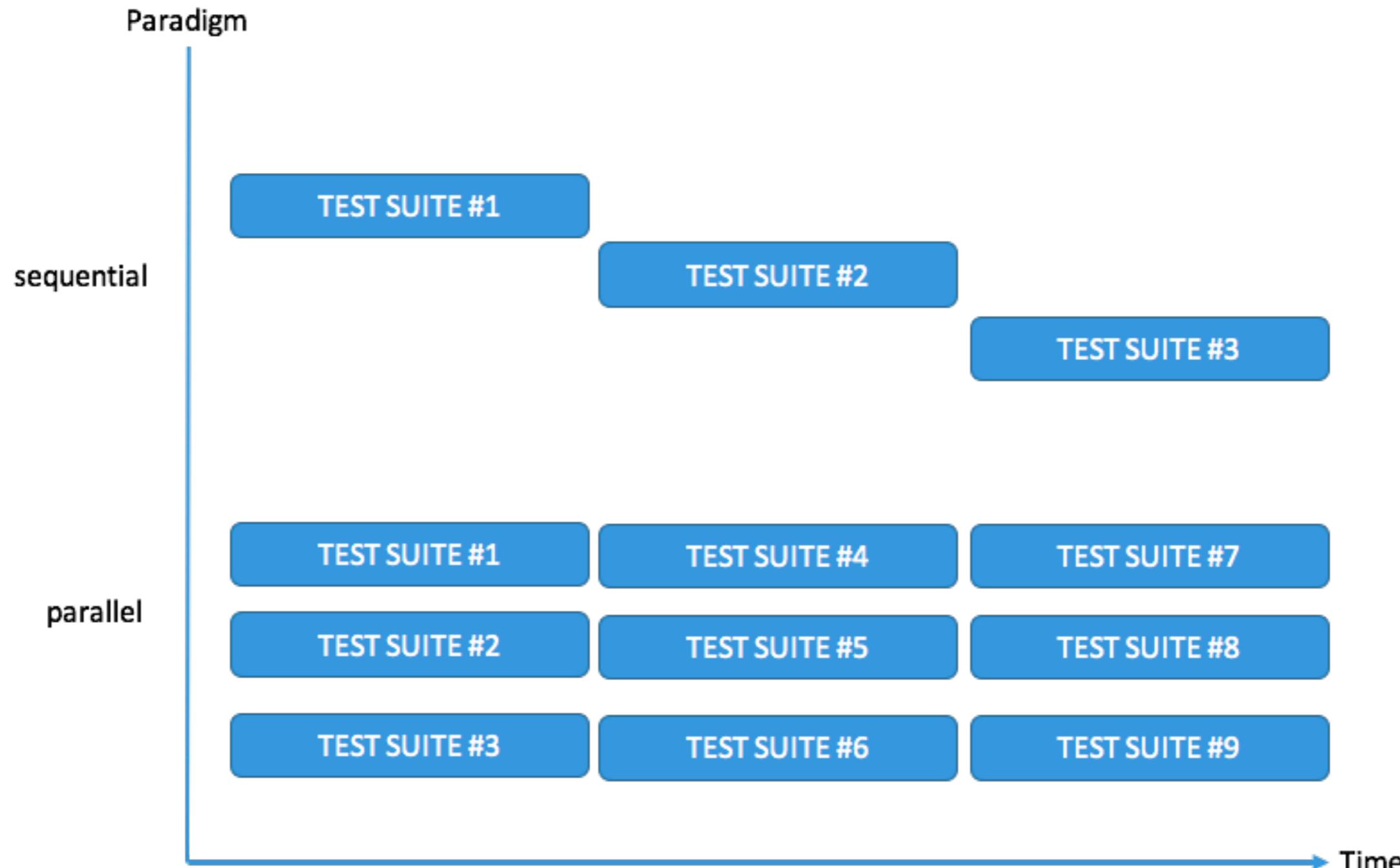


Pabot

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 --testlevelssplit  
flow_dress_sorting.robot
```



Workshop with Pabot



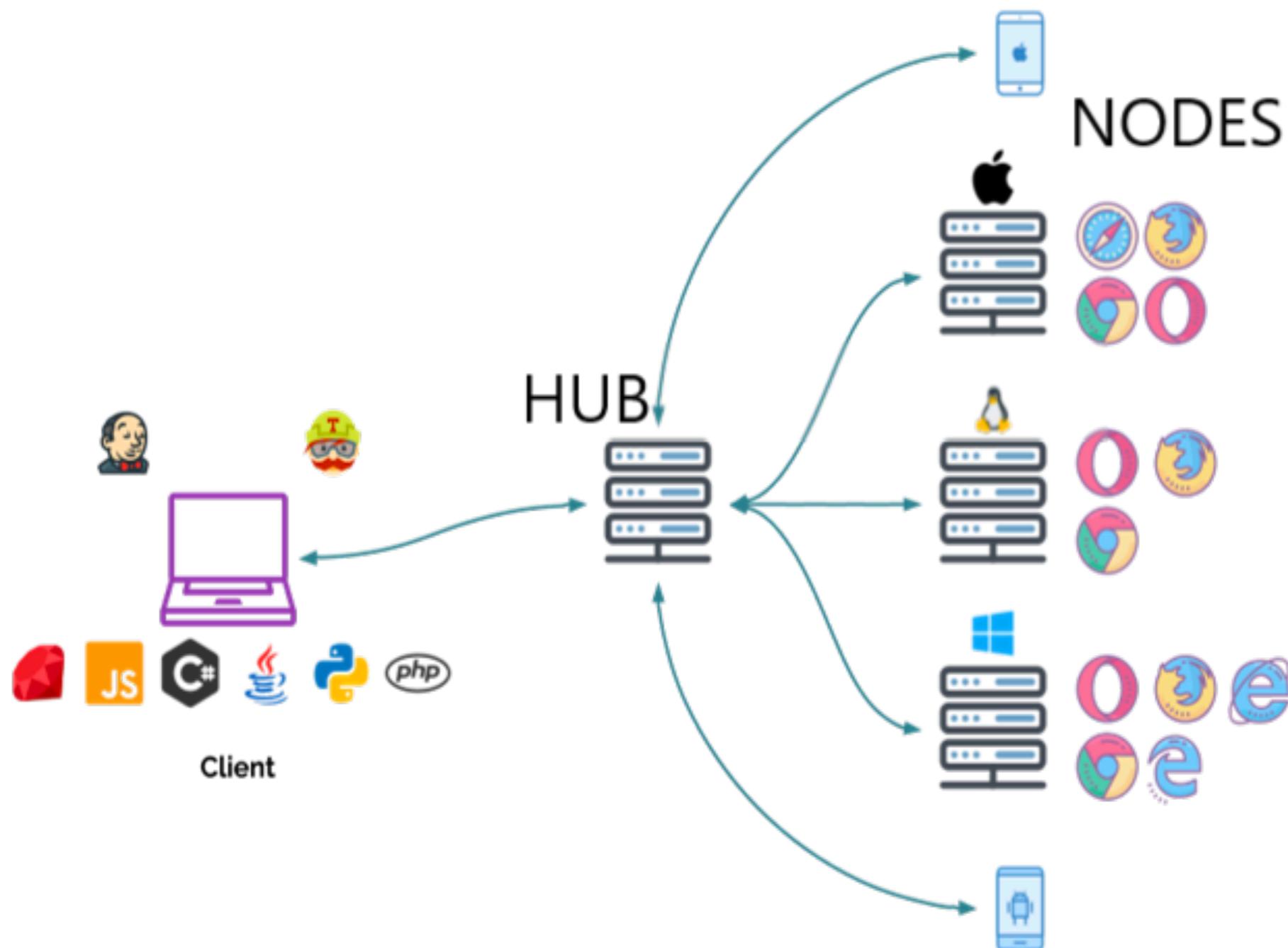
Selenium grid

Run test cases on different machines
Distributed testing

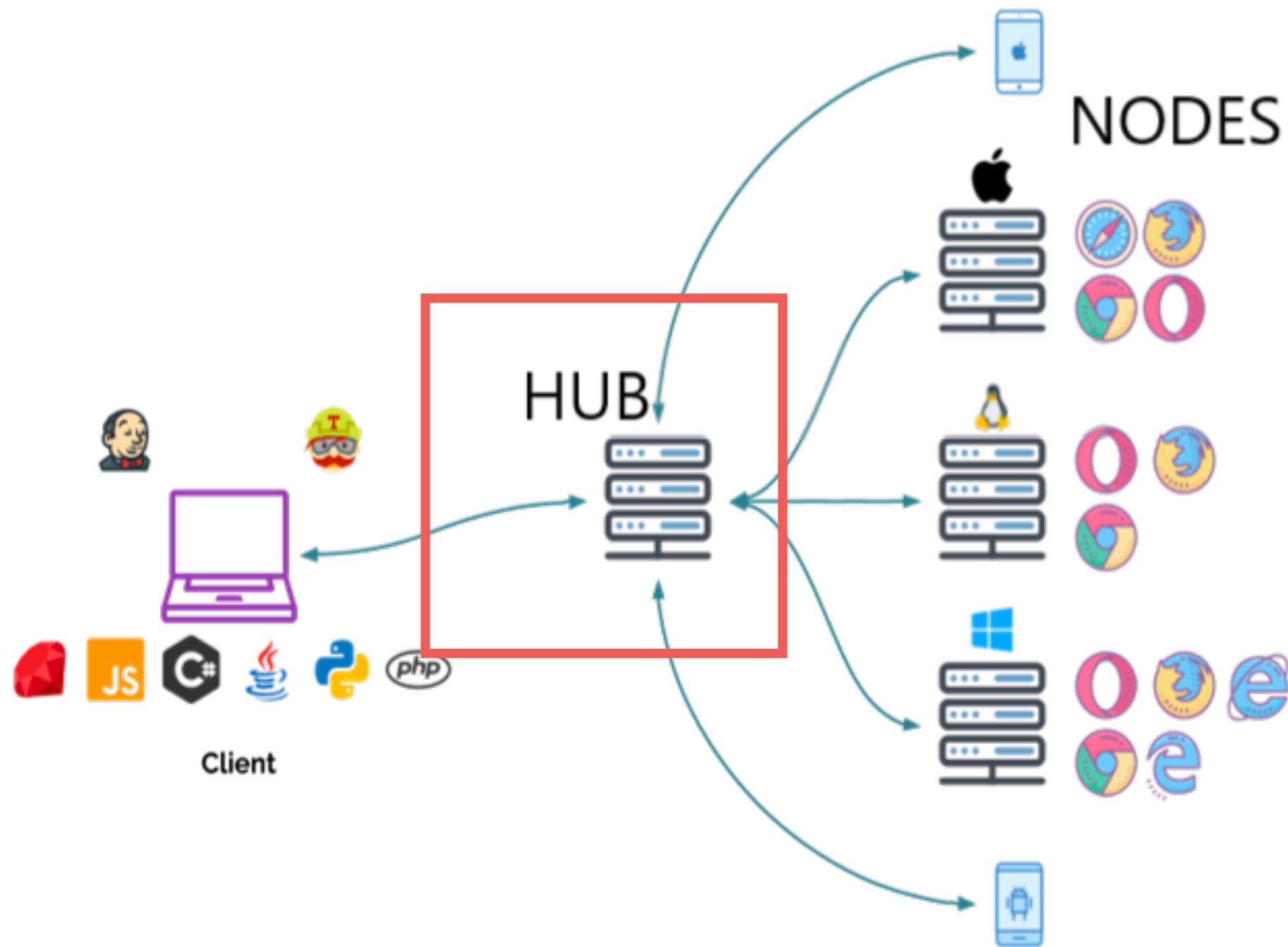
<https://www.selenium.dev/documentation/grid/>



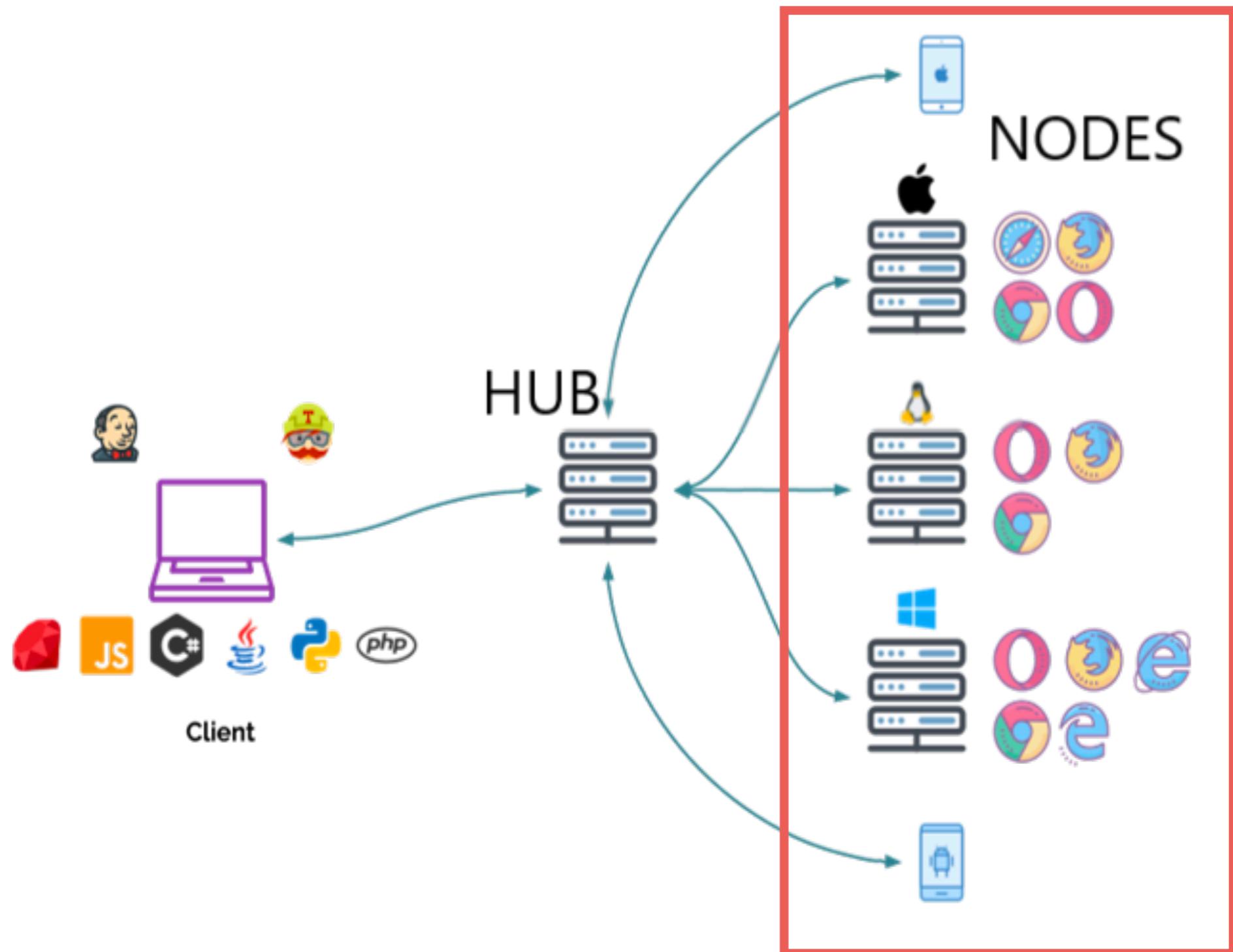
Selenium grid architecture



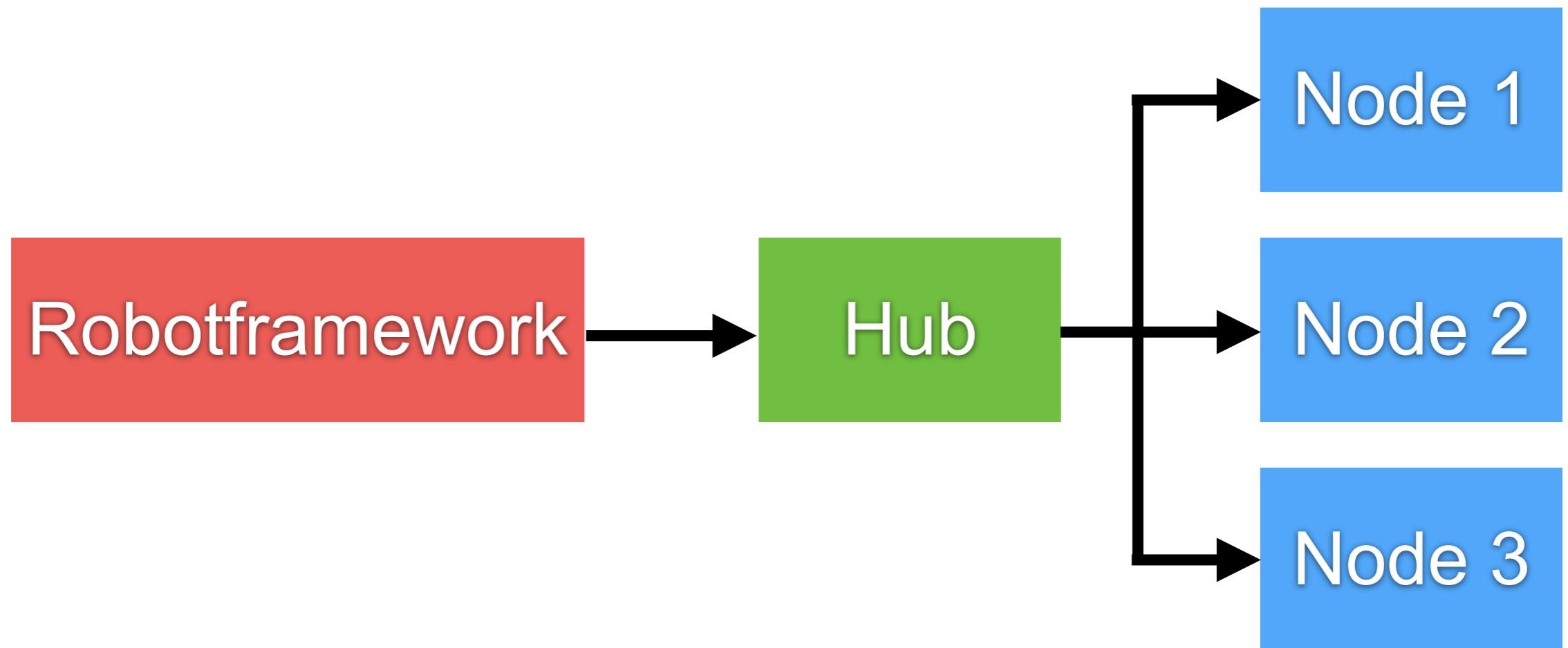
Selenium grid architecture



Selenium grid architecture



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

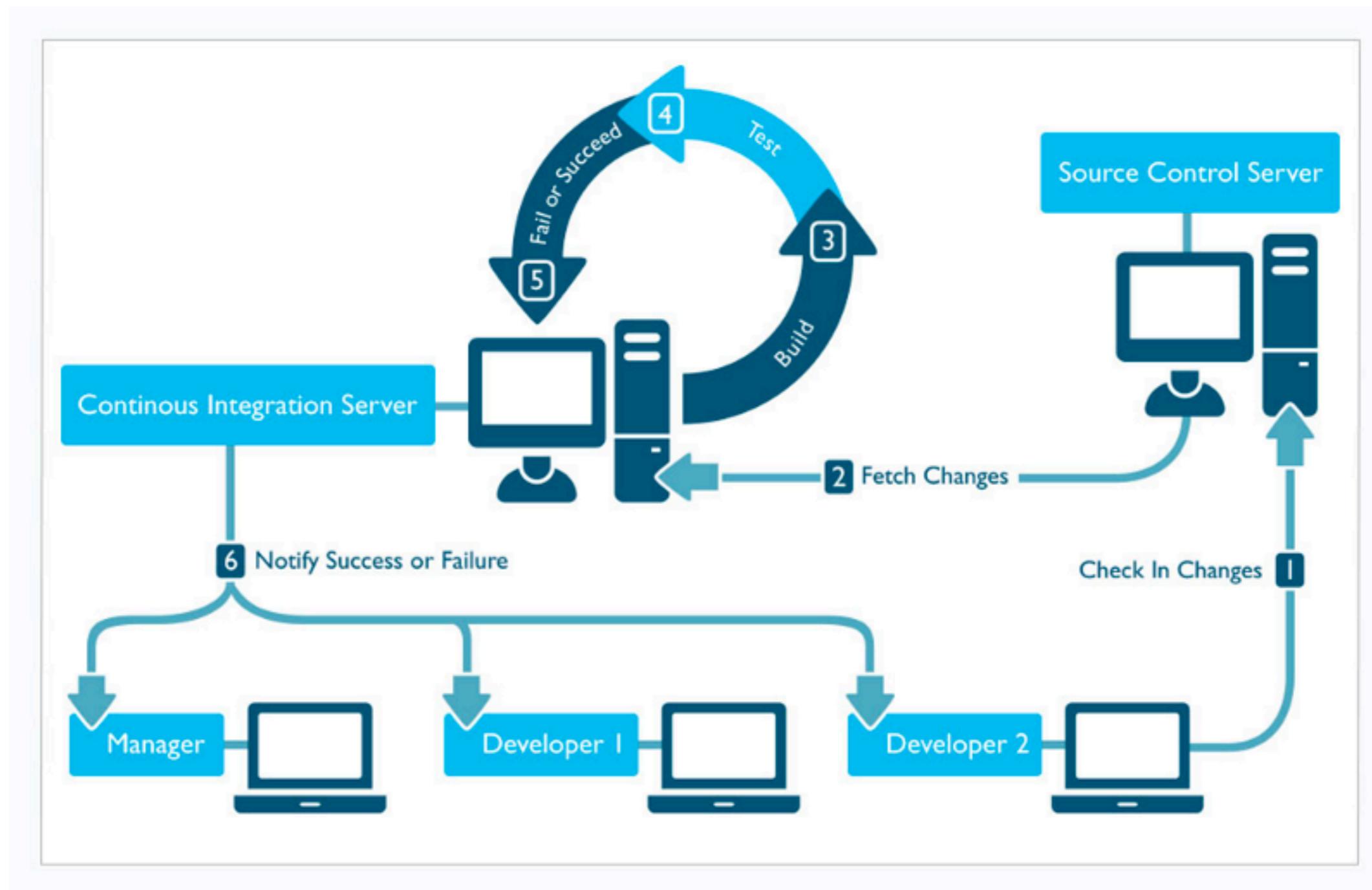
The screenshot shows the Selenium Grid Console interface. At the top, there is a logo consisting of four colored squares (orange, purple, yellow, green) arranged in a 2x2 grid, followed by the text "Grid Console v.4.0.0-alpha-2". Below this, a status bar displays "DefaultRemoteProxy (version : 4.0.0-alpha-2)" and "id : http://192.168.1.33:29618, OS : MAC". The main area has two tabs: "Browsers" (which is active) and "Configuration". Under the "Browsers" tab, there is a section for "WebDriver" with three rows of browser icons. The first row contains five Firefox icons. The second row contains one Internet Explorer icon and one Chrome icon. The third row contains five Chrome icons. The third row is highlighted with a red border. At the bottom left, there is a link labeled "View Config".



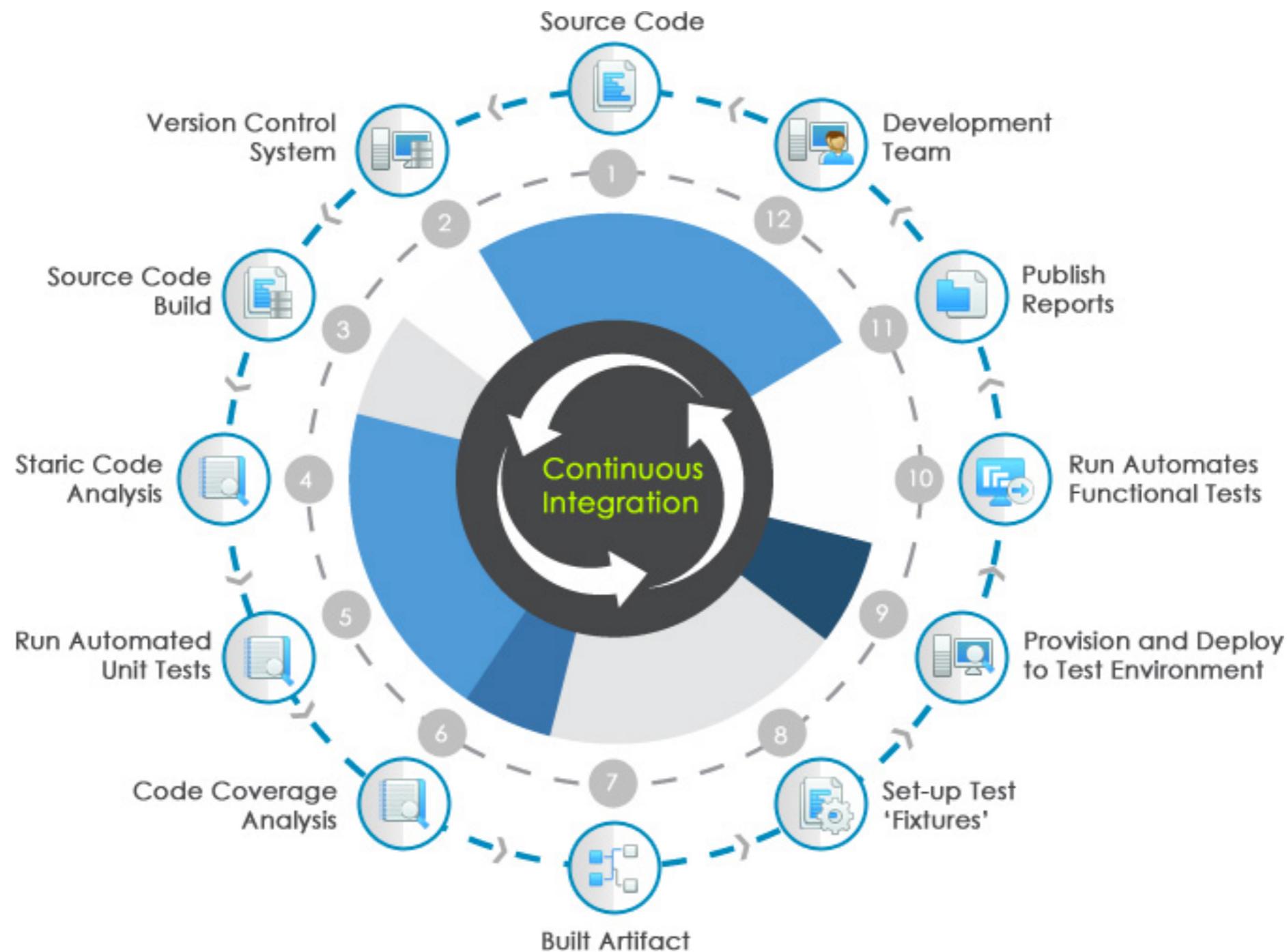
Continuous Integration



Continuous Integration process



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

Build APIs together

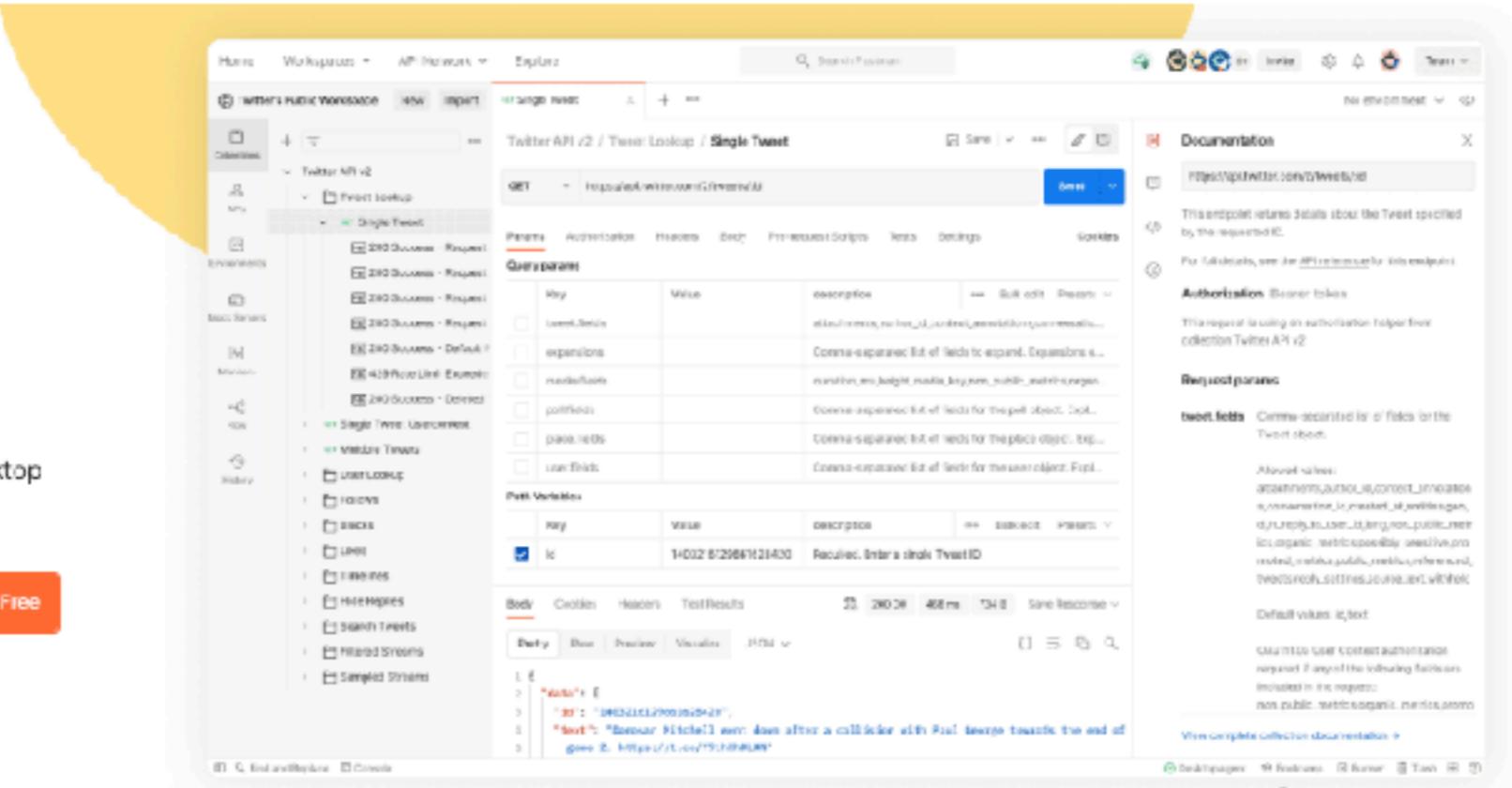
Over 20 million developers use Postman. Get started by signing up or downloading the desktop app.

[Sign Up for Free](#)

Download the desktop app   

What is Postman?

Postman is an API platform for building and using APIs. Postman simplifies each step of the API lifecycle and streamlines collaboration so you can create better APIs—“APIs for Humans.”



<https://www.postman.com/>



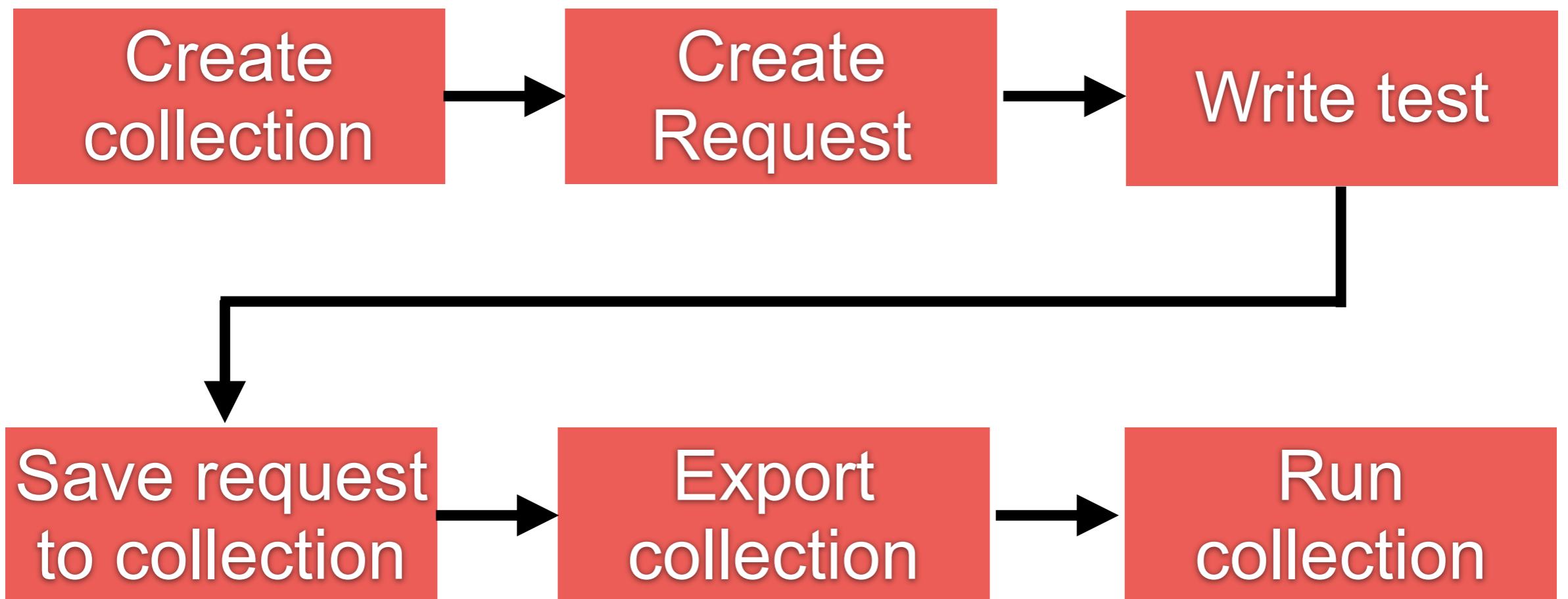
Postman in command line

```
$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 outfile.xml
```

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



Testing Flow



Workshop API testing



Workshop API testing

<https://jsonplaceholder.typicode.com/>

JSONPlaceholder

Fake Online REST API for Testing and Prototyping

Serving ~350M requests per month

Powered by [JSON Server](#) + [LowDB](#)

| ● BECOME A PATRON



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"  
            }  
        },  
    },
```



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)

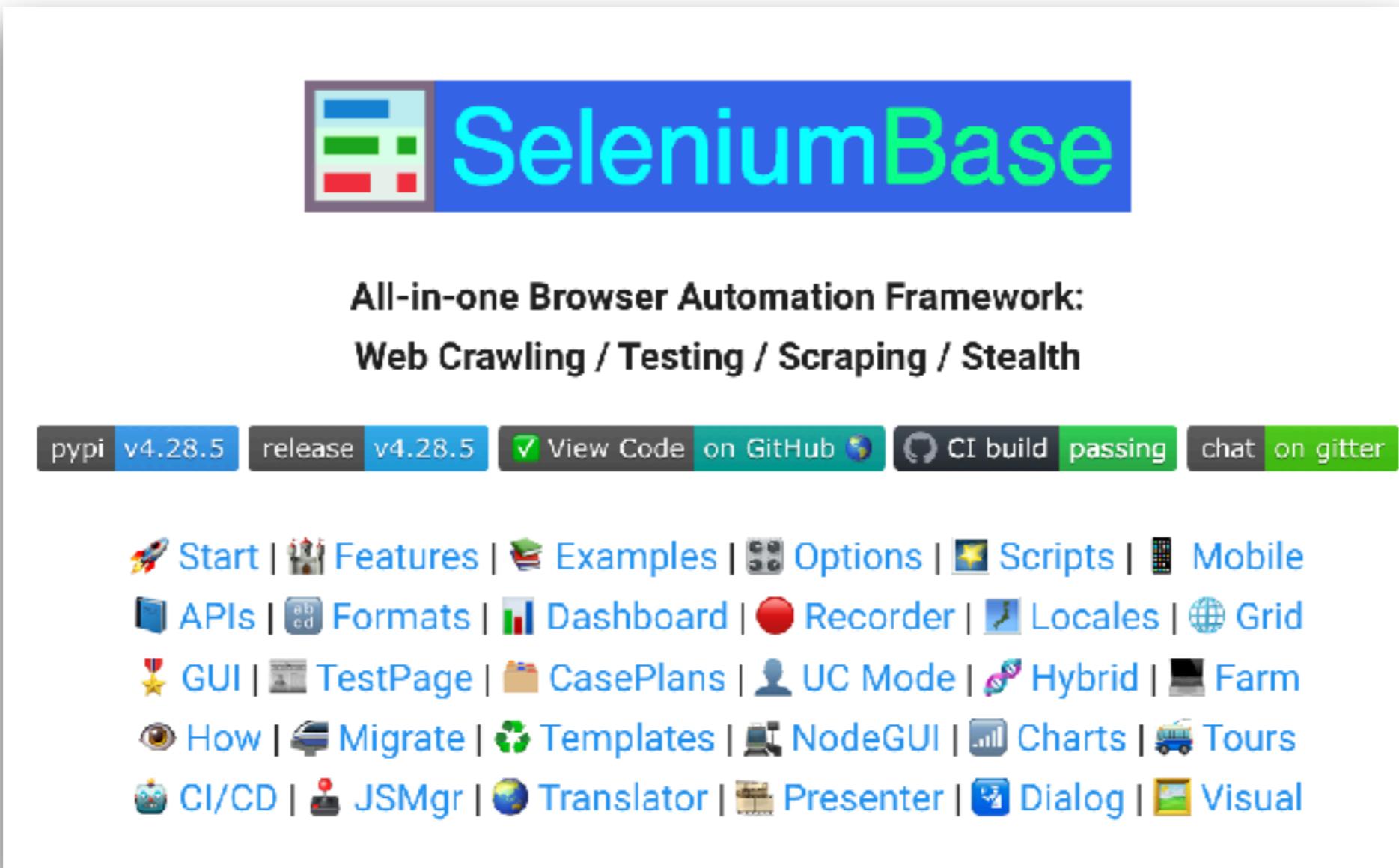
```
1 import csv
2
3 def read(filename):
4     with open(filename) as csvfile:
5         spamreader = csv.reader(csvfile)
6         for row in spamreader:
7             for c in row:
8                 print(c, end=' ')
9         print()
```



Practice



SeleniumBase



The screenshot shows the official website for SeleniumBase. At the top is a large blue header bar with the SeleniumBase logo (a stylized grid of colored squares) and the text "SeleniumBase". Below the header, a main title reads "All-in-one Browser Automation Framework: Web Crawling / Testing / Scraping / Stealth". A navigation bar below the title includes links for "Start", "Features", "Examples", "Options", "Scripts", "Mobile", "APIs", "Formats", "Dashboard", "Recorder", "Locales", "Grid", "GUI", "TestPage", "CasePlans", "UC Mode", "Hybrid", "Farm", "How", "Migrate", "Templates", "NodeGUI", "Charts", "Tours", "CI/CD", "JSMgr", "Translator", "Presenter", "Dialog", and "Visual". At the bottom of the page, there is a URL field containing "https://seleniumbase.io/".

<https://seleniumbase.io/>



Selenium demo



<https://www.selenium.dev/selenium/web/index.html>



QA Hive



HOME ECOMMERCE SWAGGER TODO FORM DATA-GENERATOR

ยินดีต้อนรับสู่ QA Web Demo

เว็บไซต์นี้จัดทำเพื่อการจำลองการทดสอบระบบเท่านั้น

เลือก Test Framework ไม่ถูก?

ลองทำแบบสอบถามนี้ดู

ทำแบบสอบถาม คลิก

สนับสนุนเราได้ เพียงซื้อ

LINE Sticker

PURCHASE



<https://web-demo.qahive.com/>



Robot Framework

© 2017 - 2024 Siam Chamnankit Company Limited. All rights reserved.

Q/A

**How to use in your current project ?
Development and Delivery process ?
Testing process ?**

