



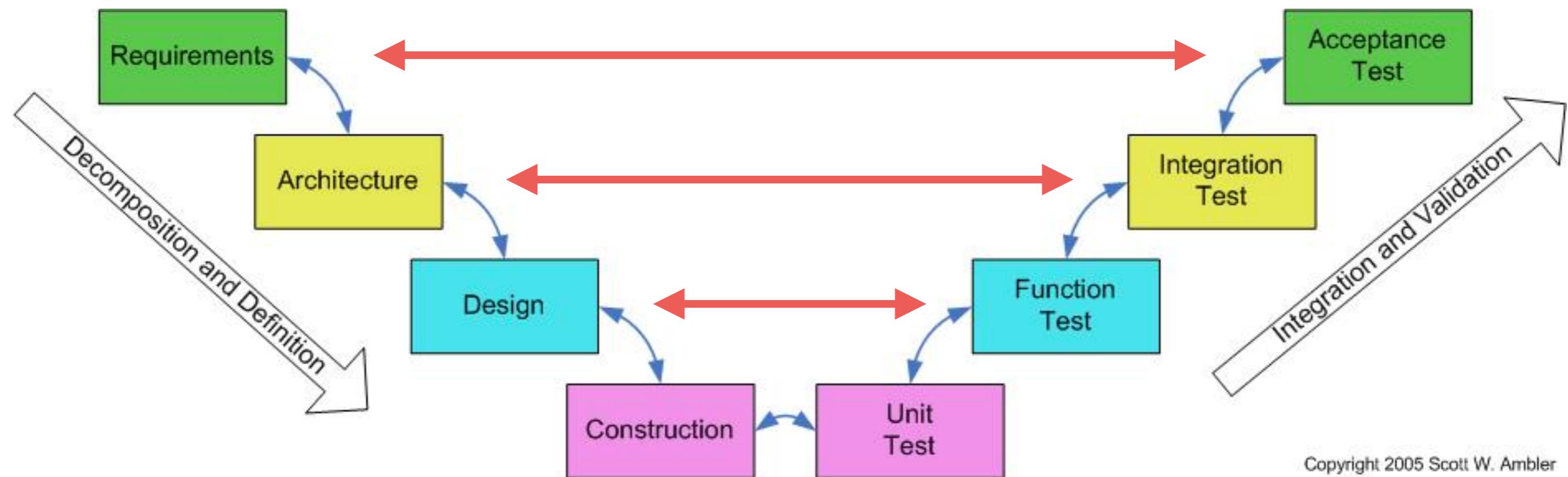
Automated testing with Katalon and Robot framework



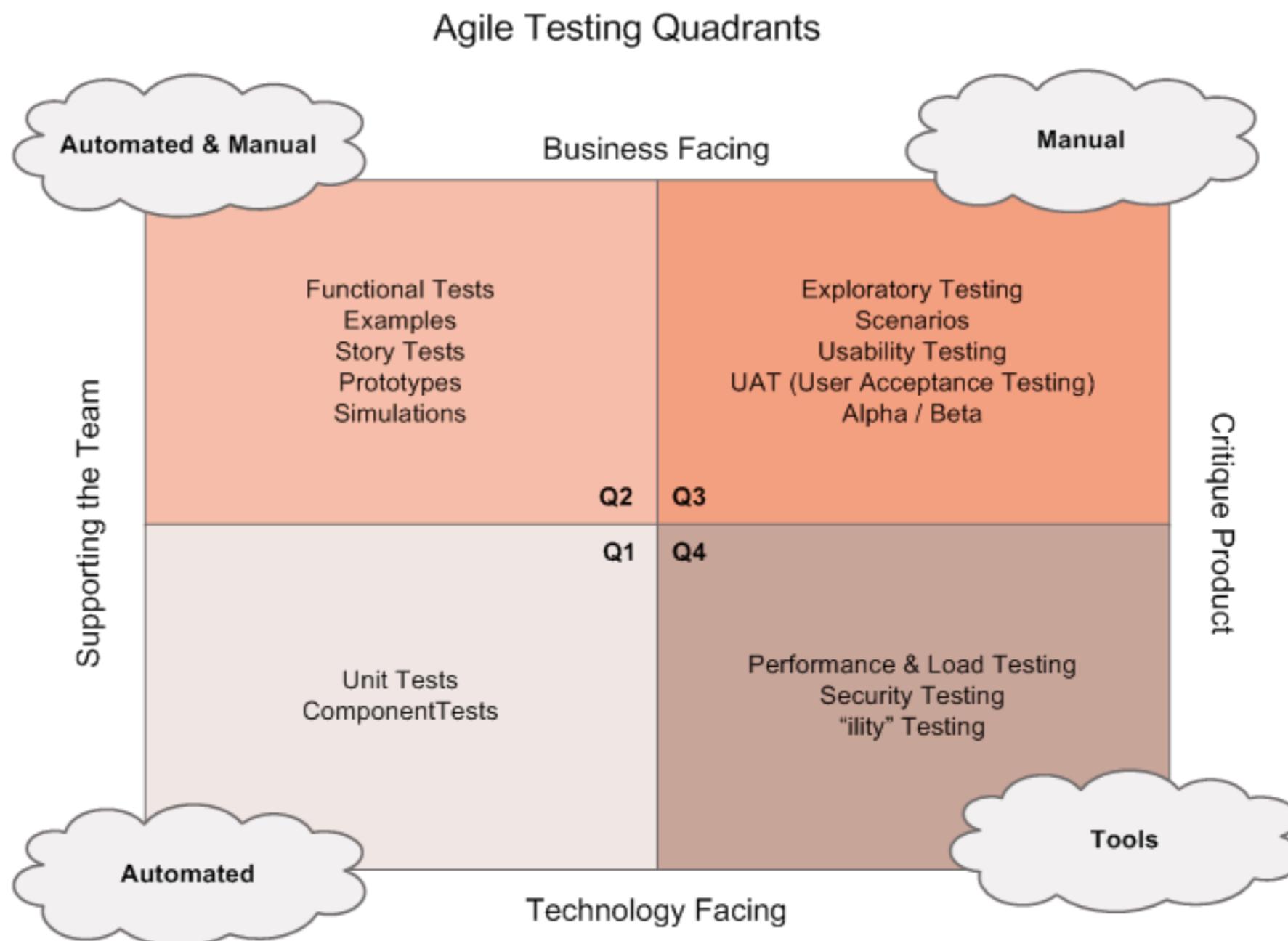
Automation Testing



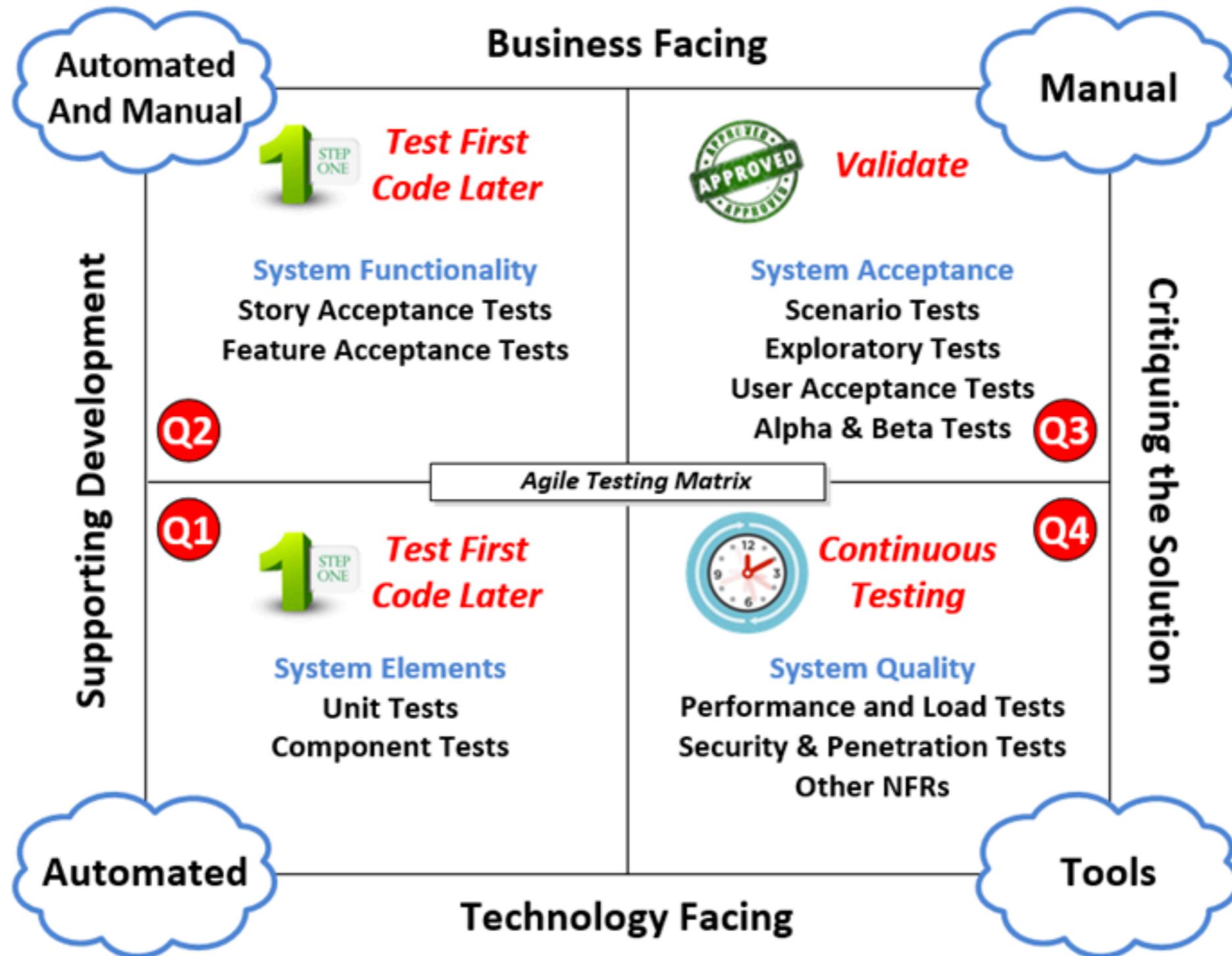
V Model



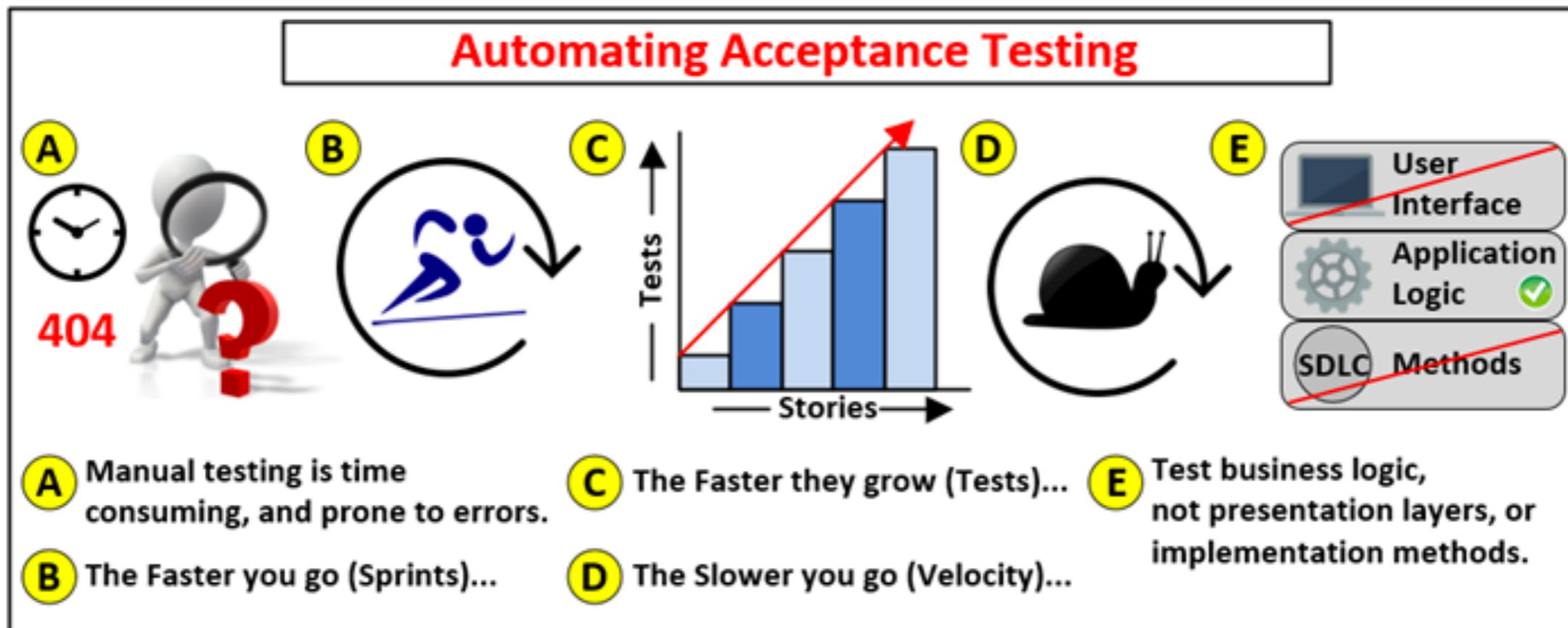
Agile Testing Quadrant



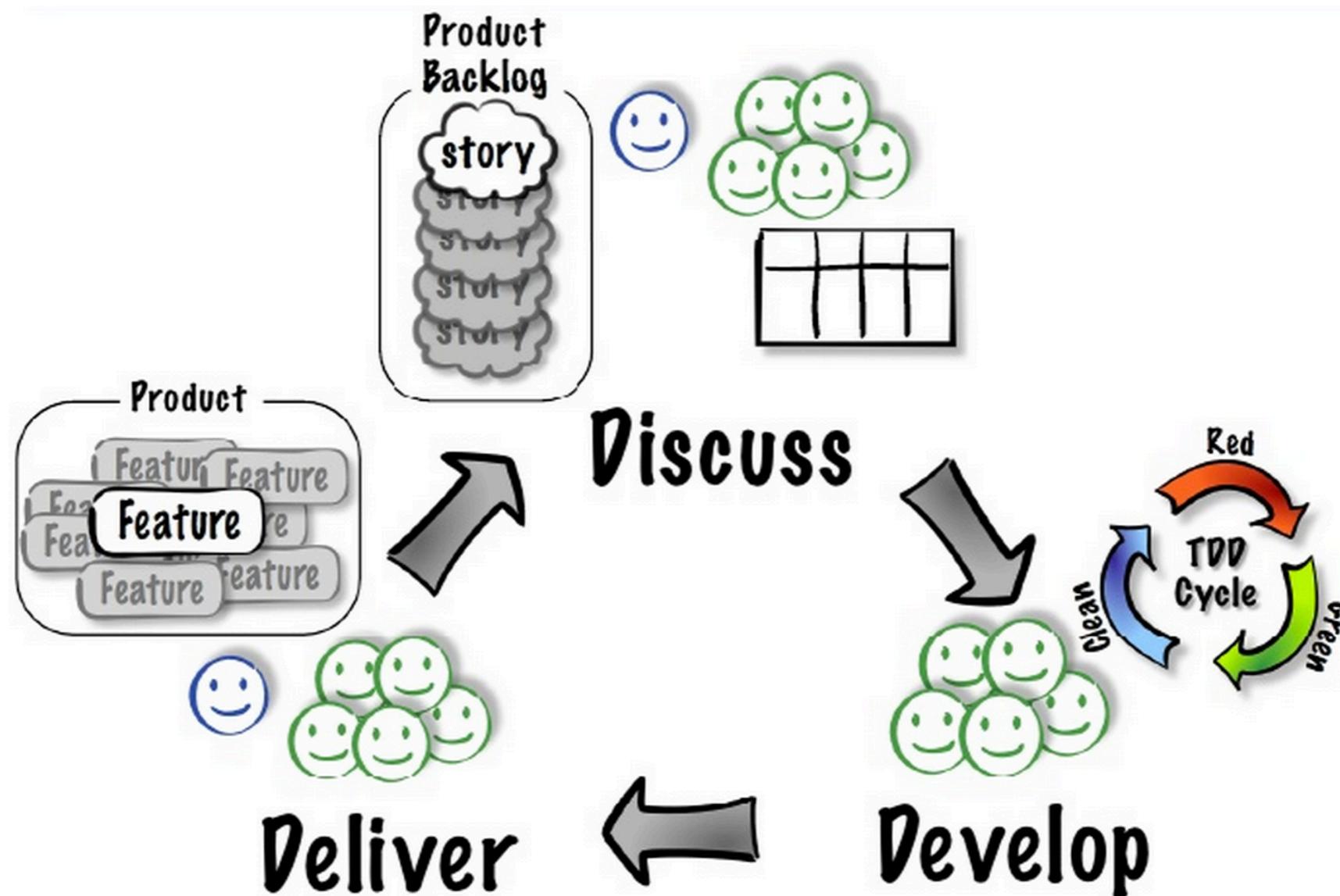
Agile Testing Quadrant



Test automation is essential



Acceptance Test-Driven Development



(Model developed with Pekka Klärck, Bas Vodde, and Craig Larman.)



ATDD

Common language
Common and share understanding
Executable requirements or examples
Living document



Acceptance Tests

=

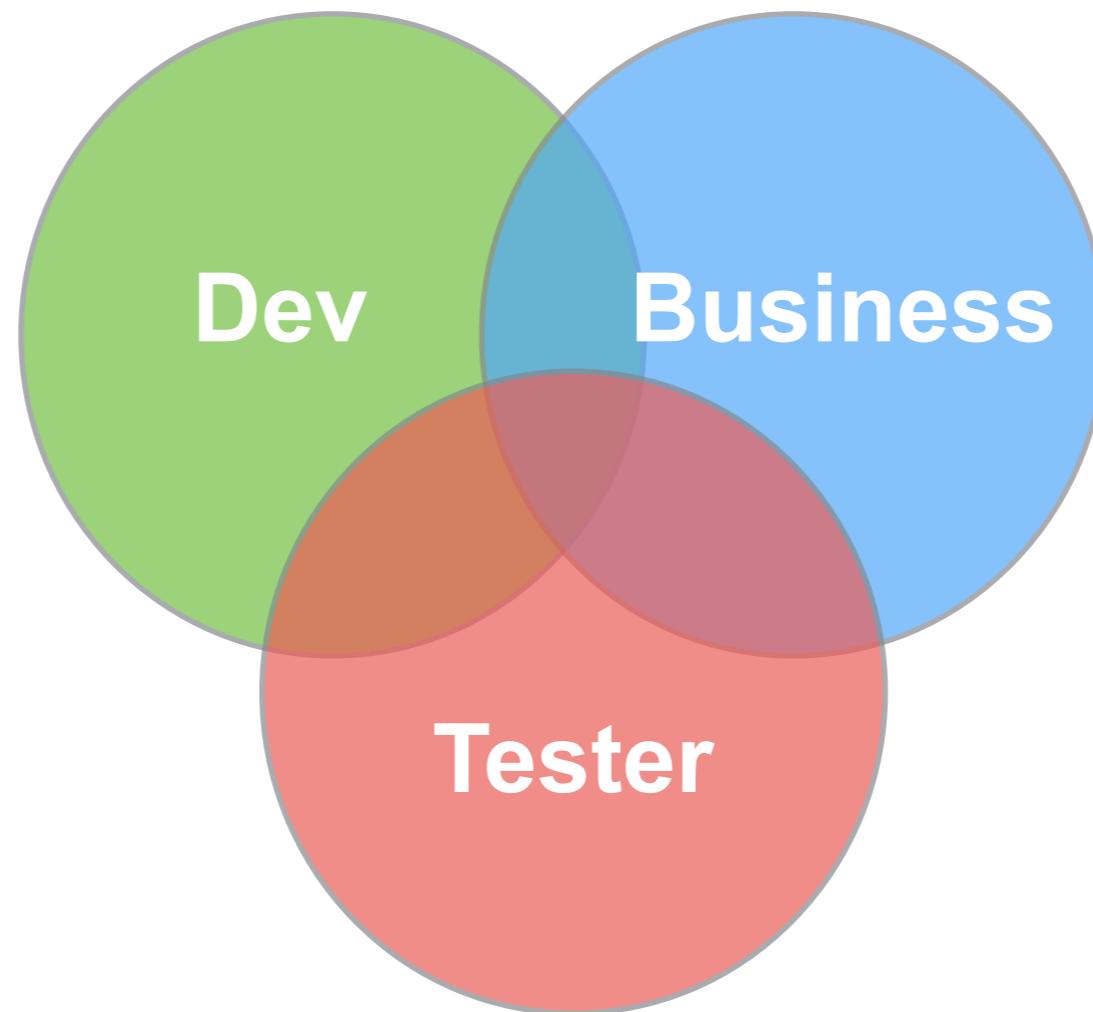
Business Criteria

+

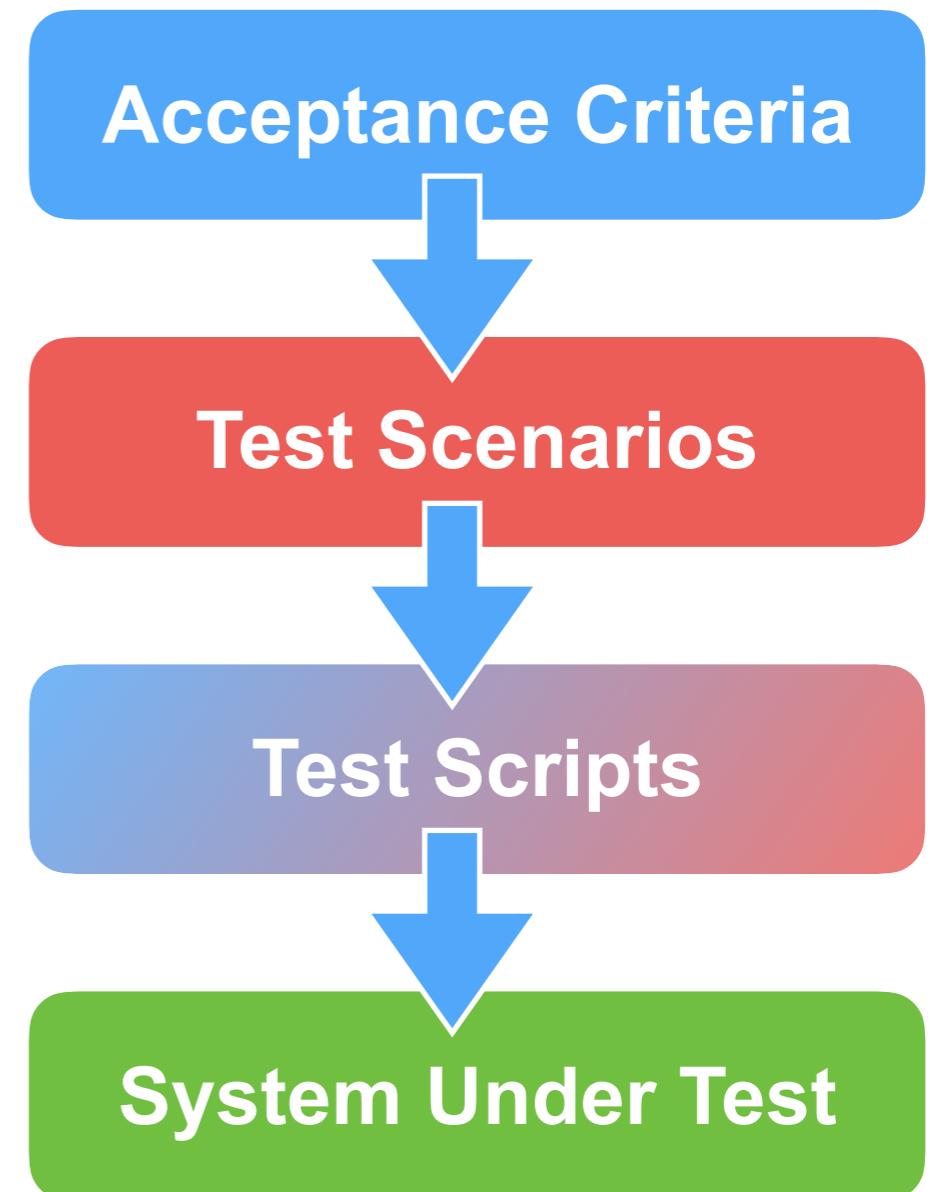
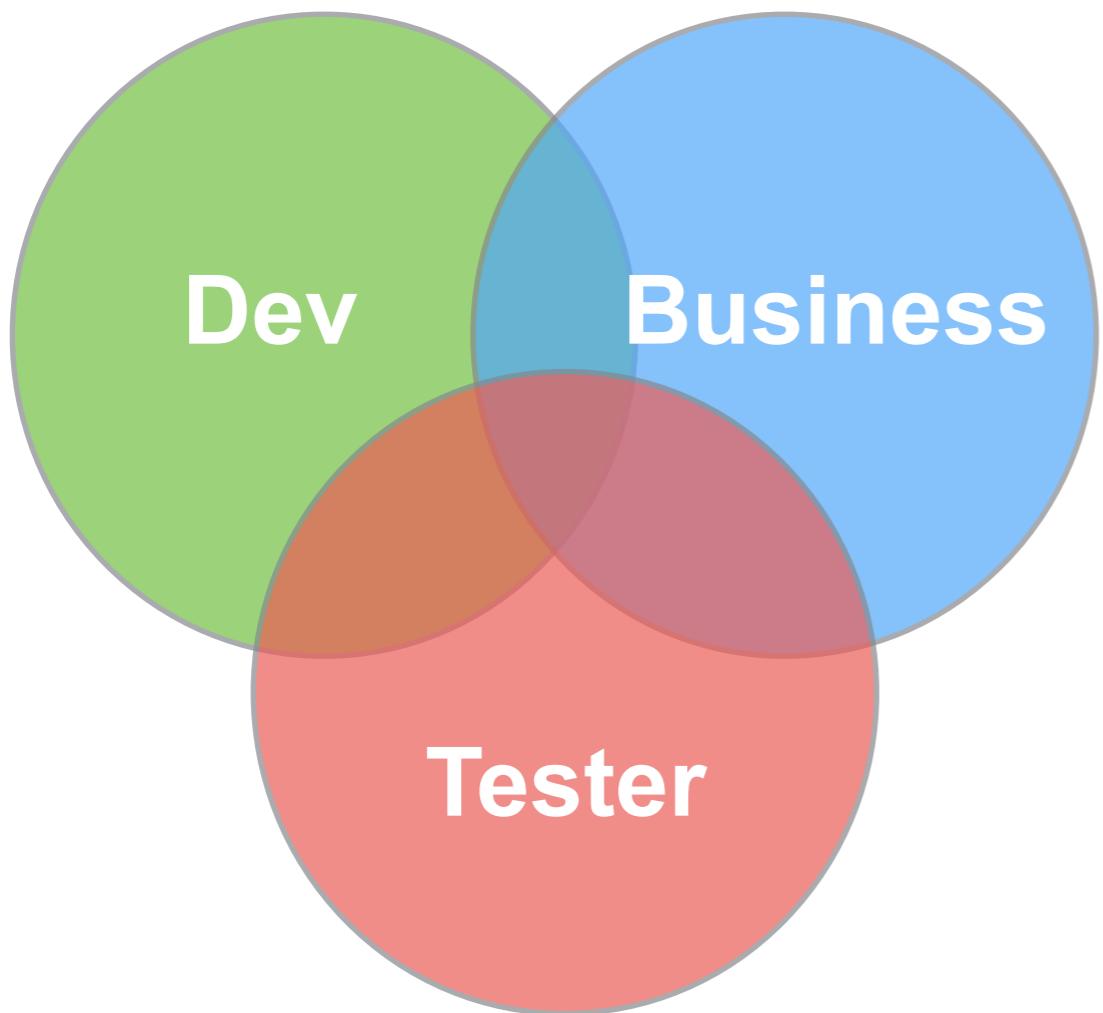
Examples (data)



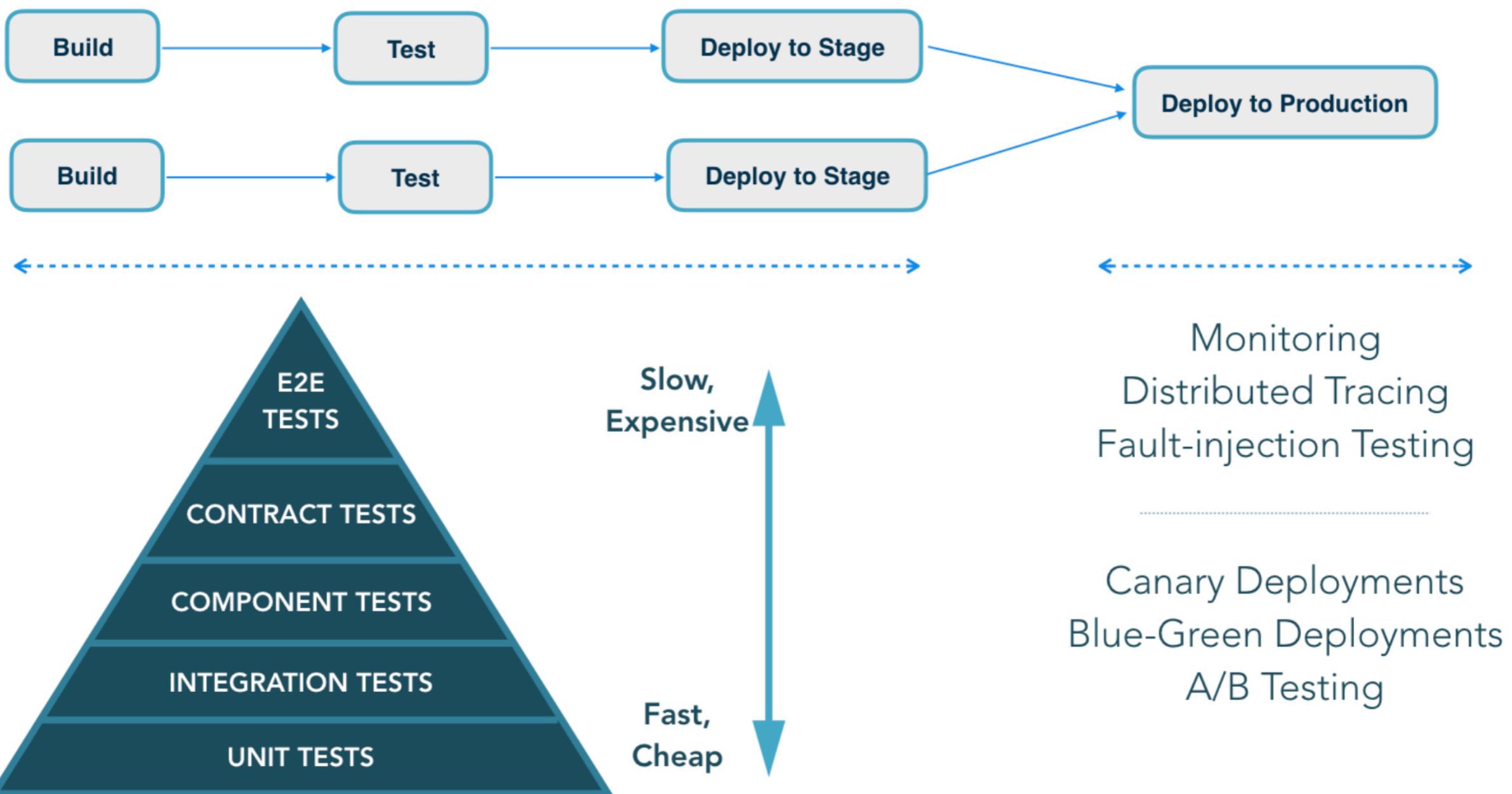
Acceptance testing



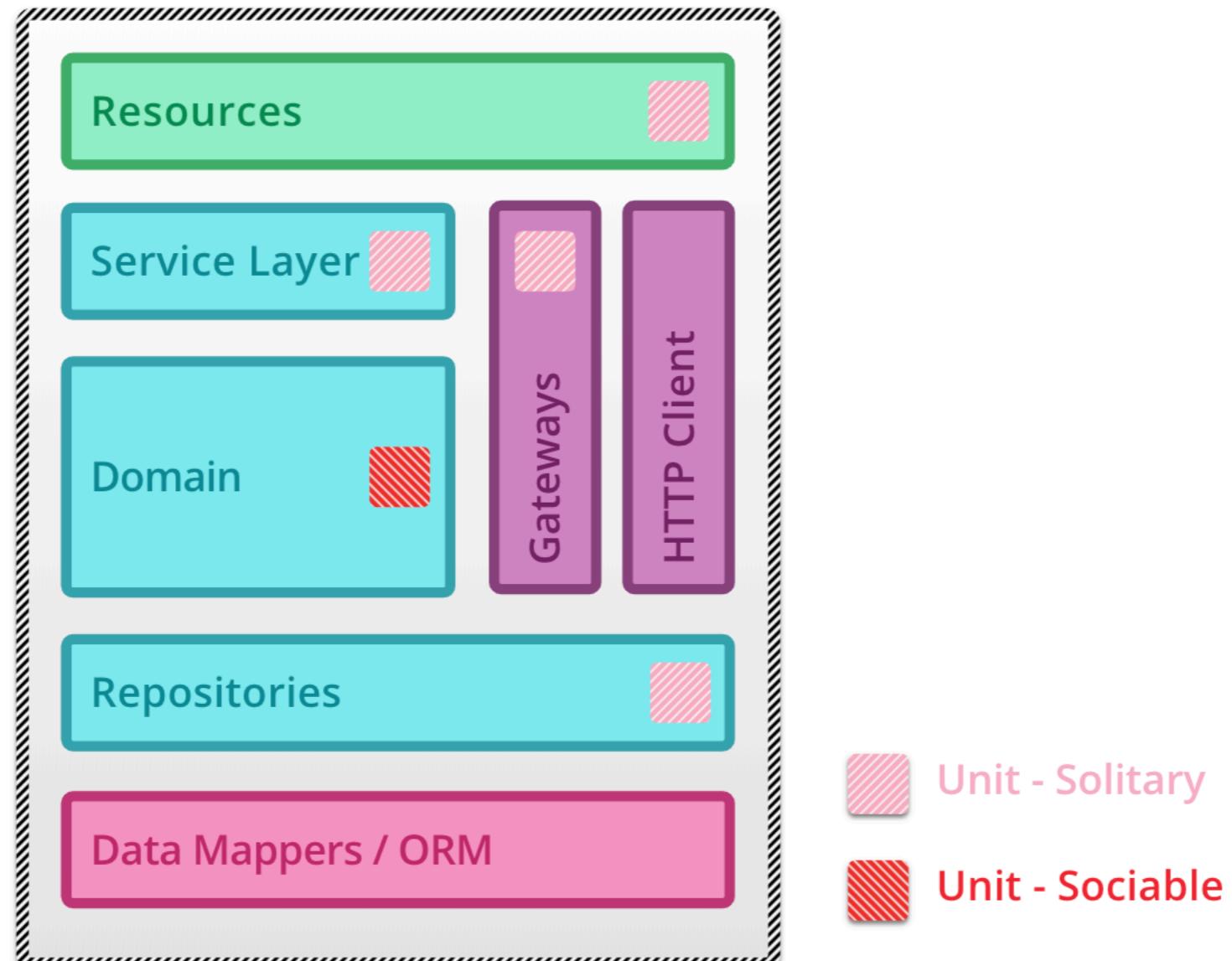
Acceptance testing



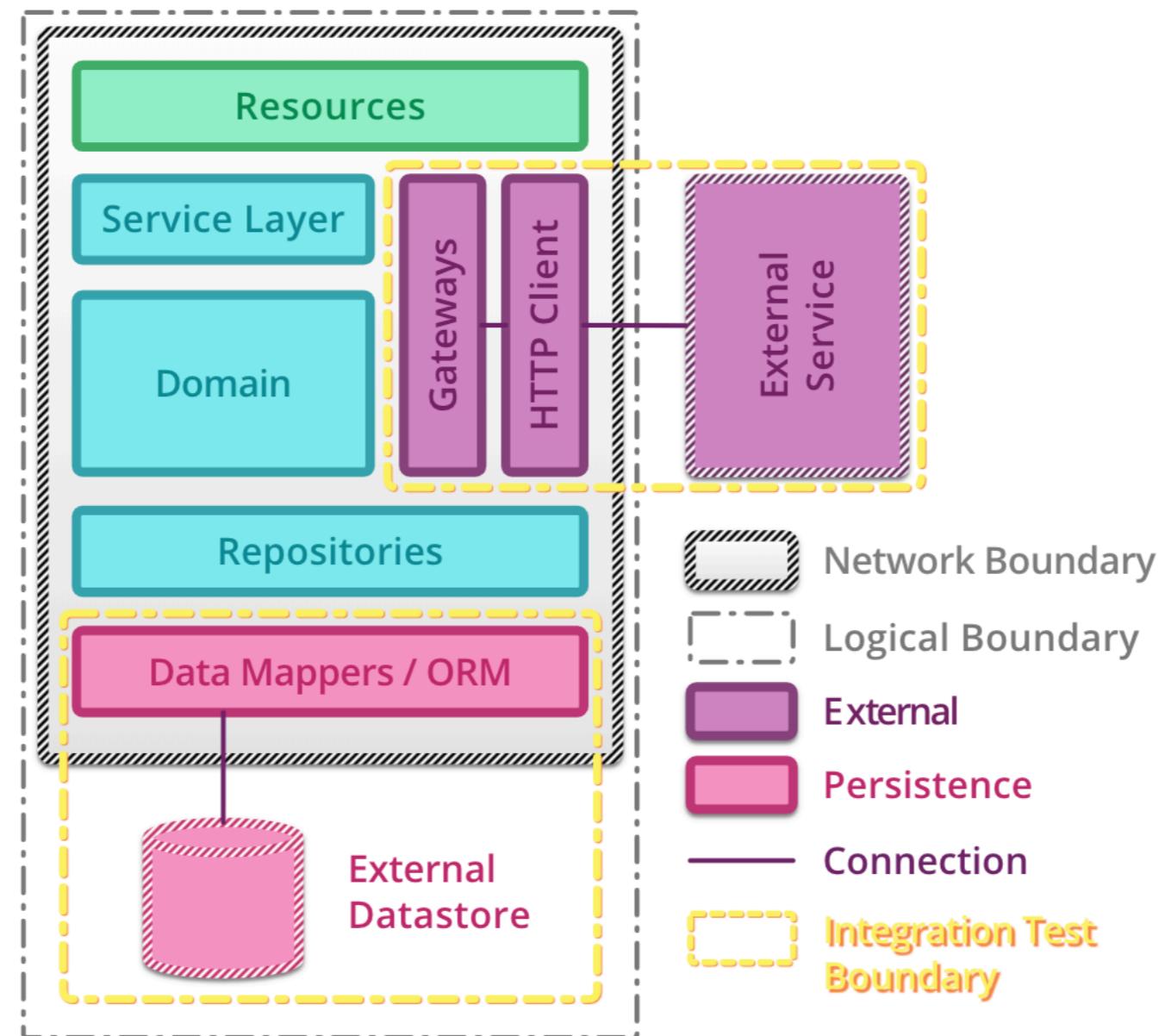
Test strategy



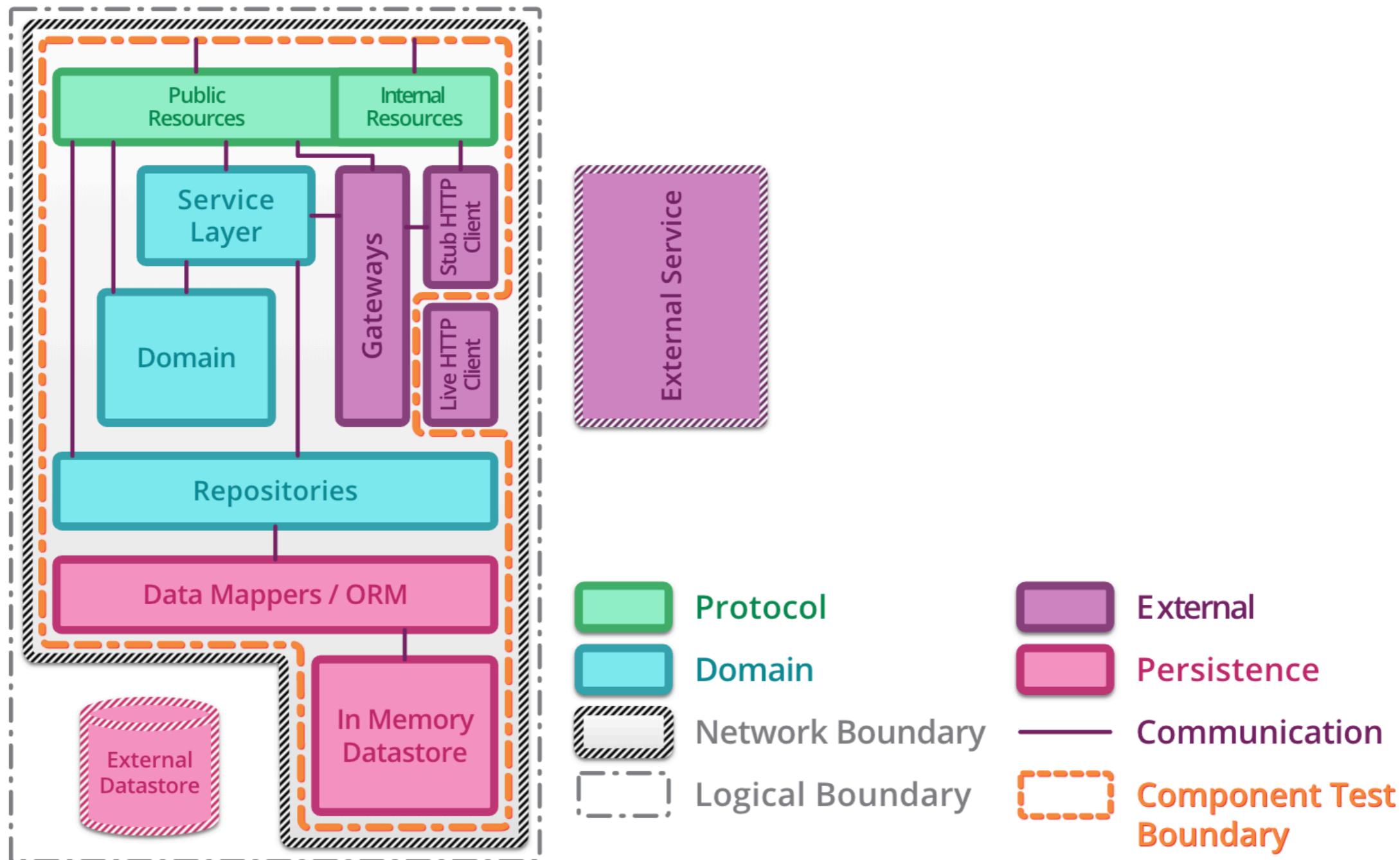
Unit testing



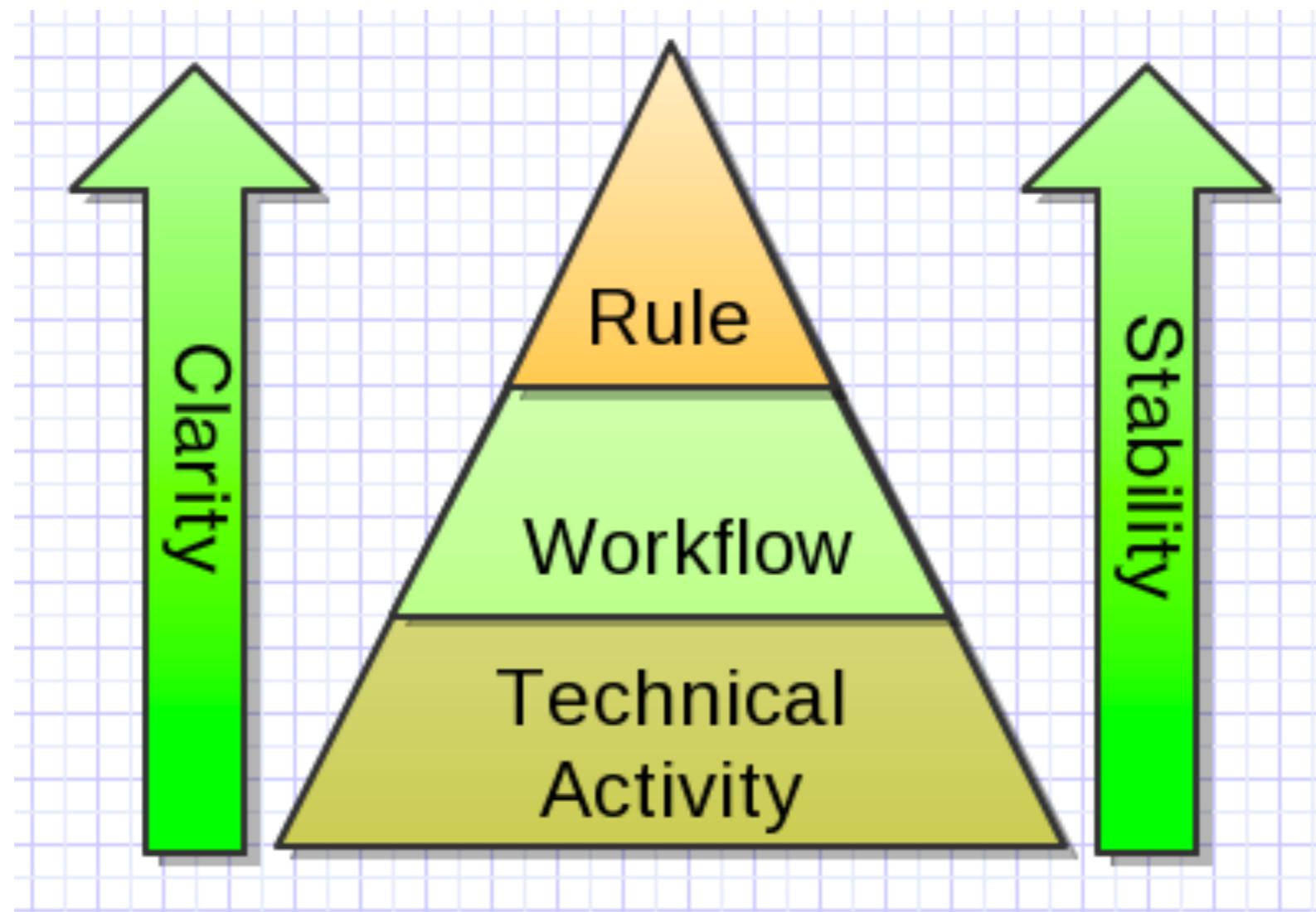
Integration testing



Component testing



3 levels of UI test automation



3 levels of UI test automation

Business rule/functionality level

what is this test demonstrating or exercising

User interface workflow level

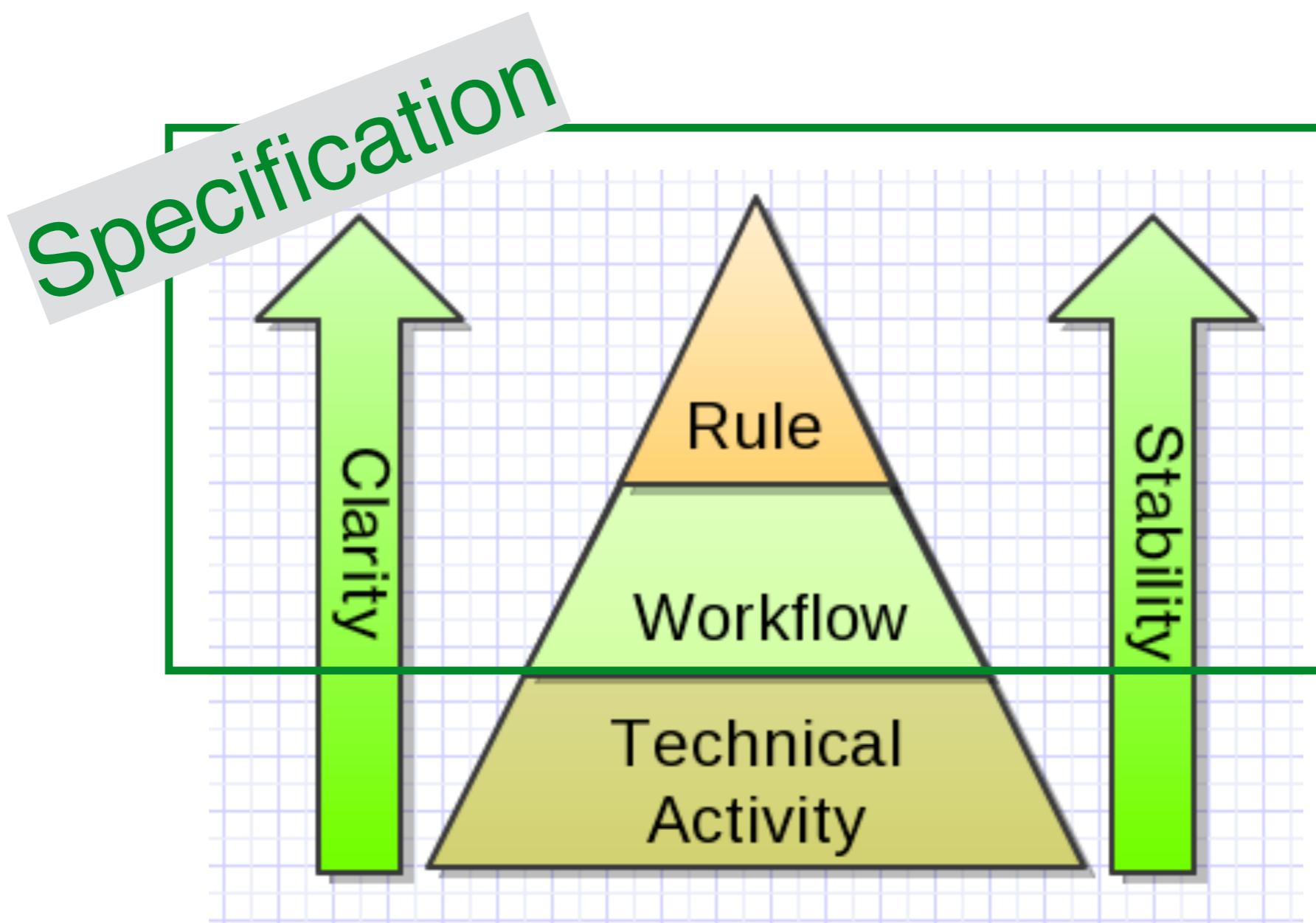
what does a user have to do to exercise the functionality through the UI

Technical activity level

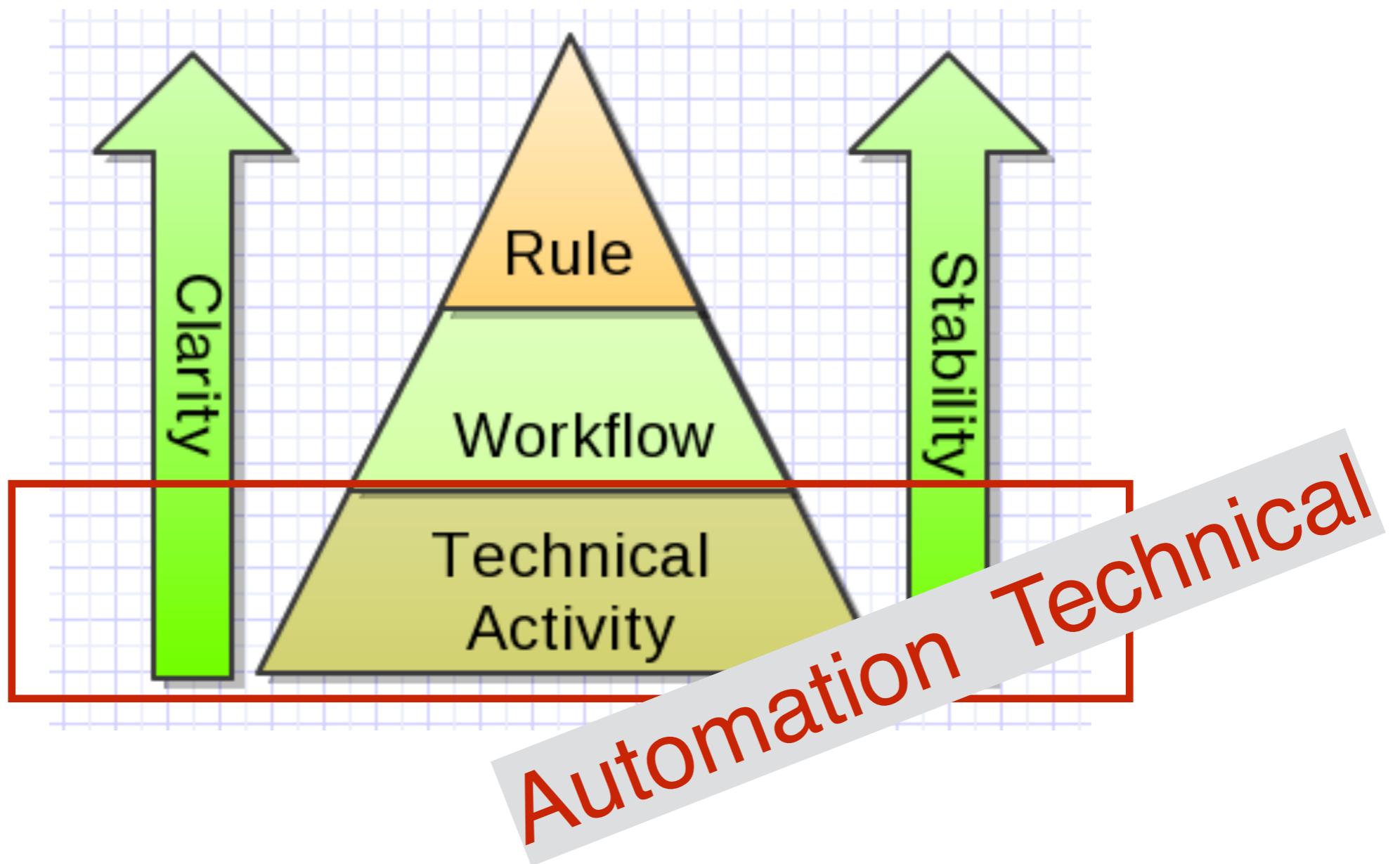
what are the technical steps required to exercise the functionality



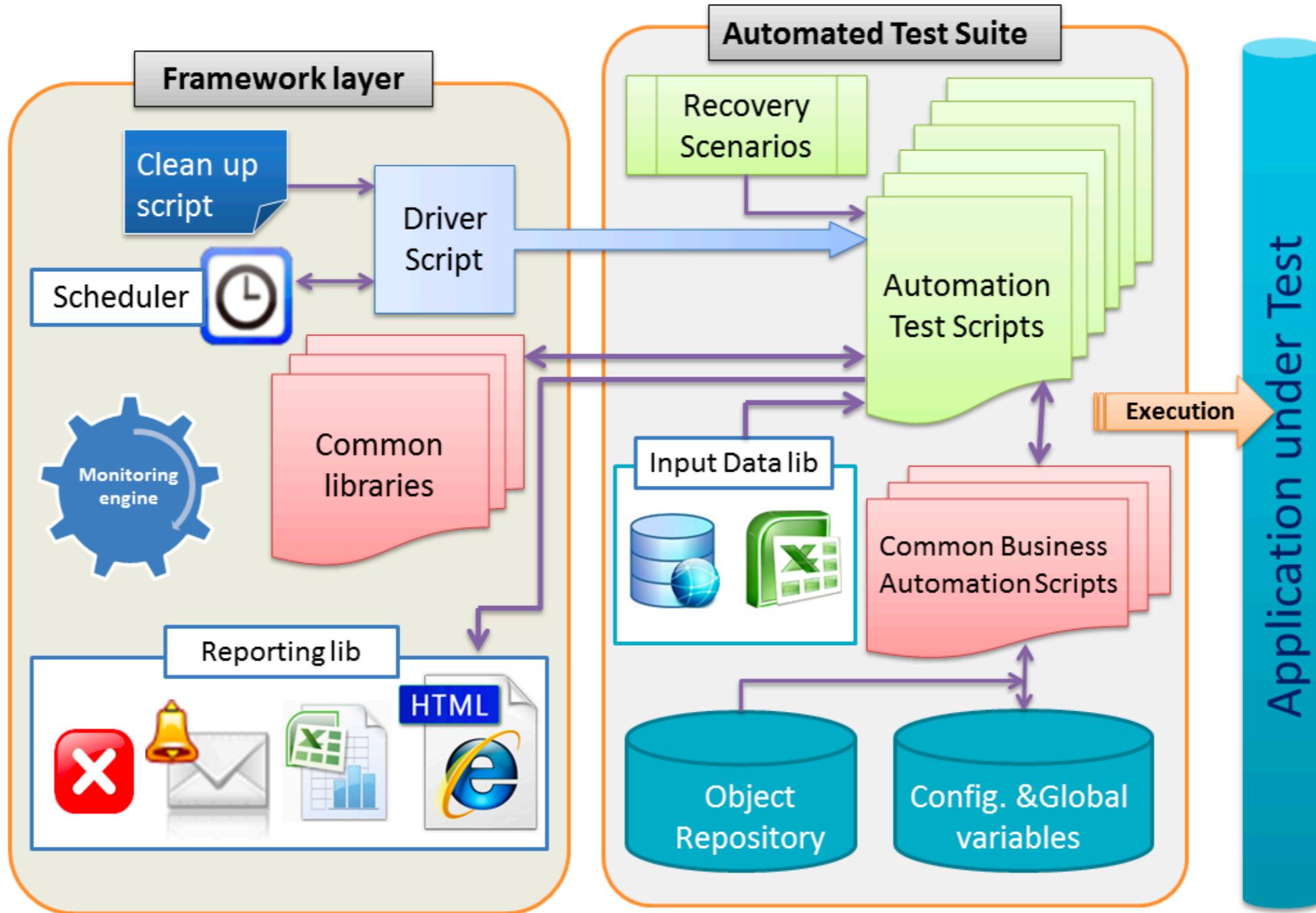
Good to start



Good to start



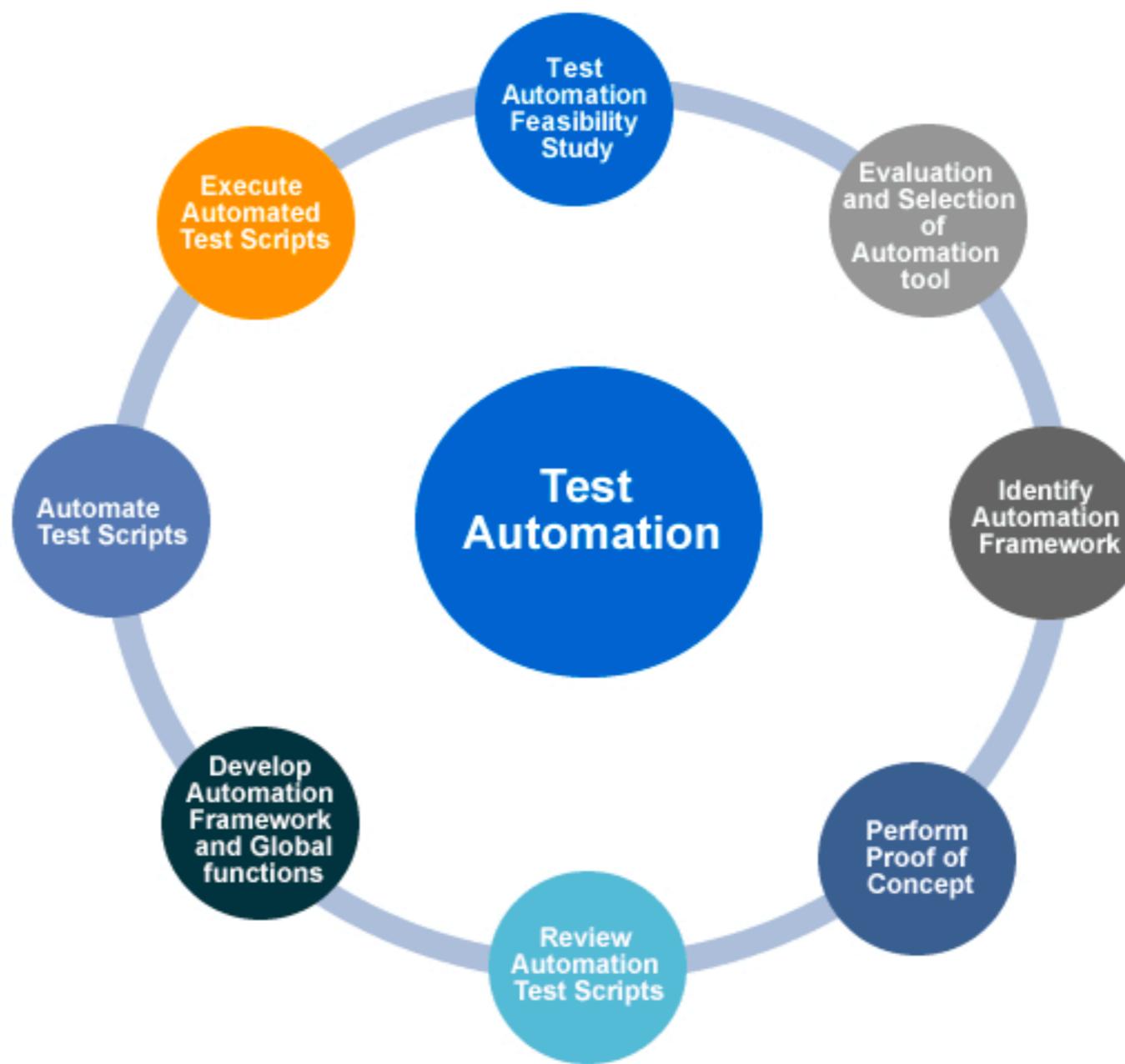
Test Framework



Test Automation strategy



Test Automation selection



Robot Framework



<https://robotframework.org/>



Official website



INTRODUCTION
EXAMPLES
LIBRARIES
TOOLS
DOCUMENTATION
SUPPORT
FOUNDATION
SHOP
ROBOCON

ROBOT FRAME WORK /

INTRODUCTION

Robot Framework is a generic test automation framework for acceptance testing and acceptance test-driven development (ATDD). It has easy-to-use tabular test data



Robot Framework

Test automation framework

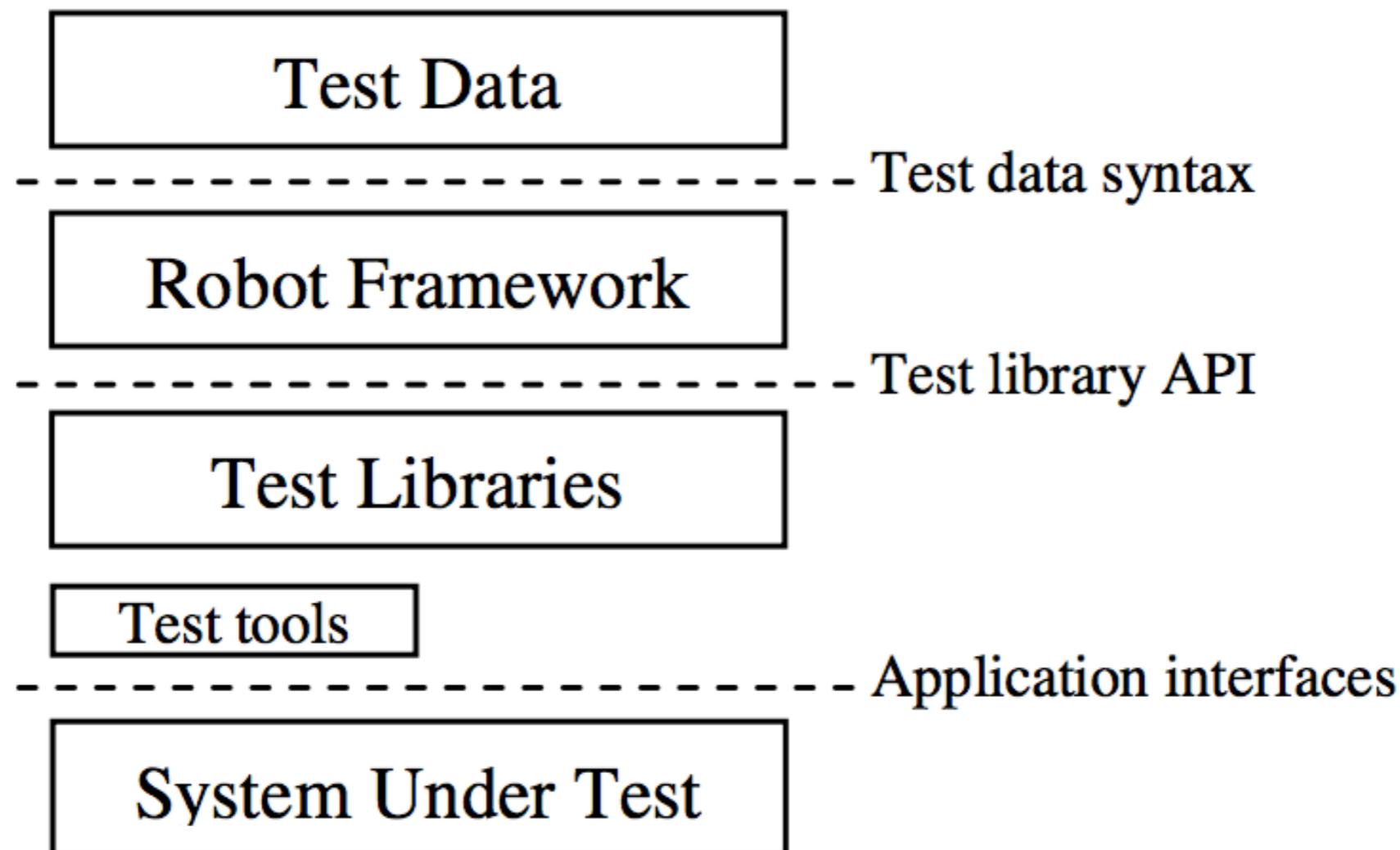
Designed for Acceptance testing(ATDD)

Developed with Python

Compatible with Java (Jython)



Robot Framework architecture



Robot Framework architecture (1)

Robot Framework Core
(standard libraries, reporting)

External library

External library

Resource
file 1

Resource
file 2

Resource
file n

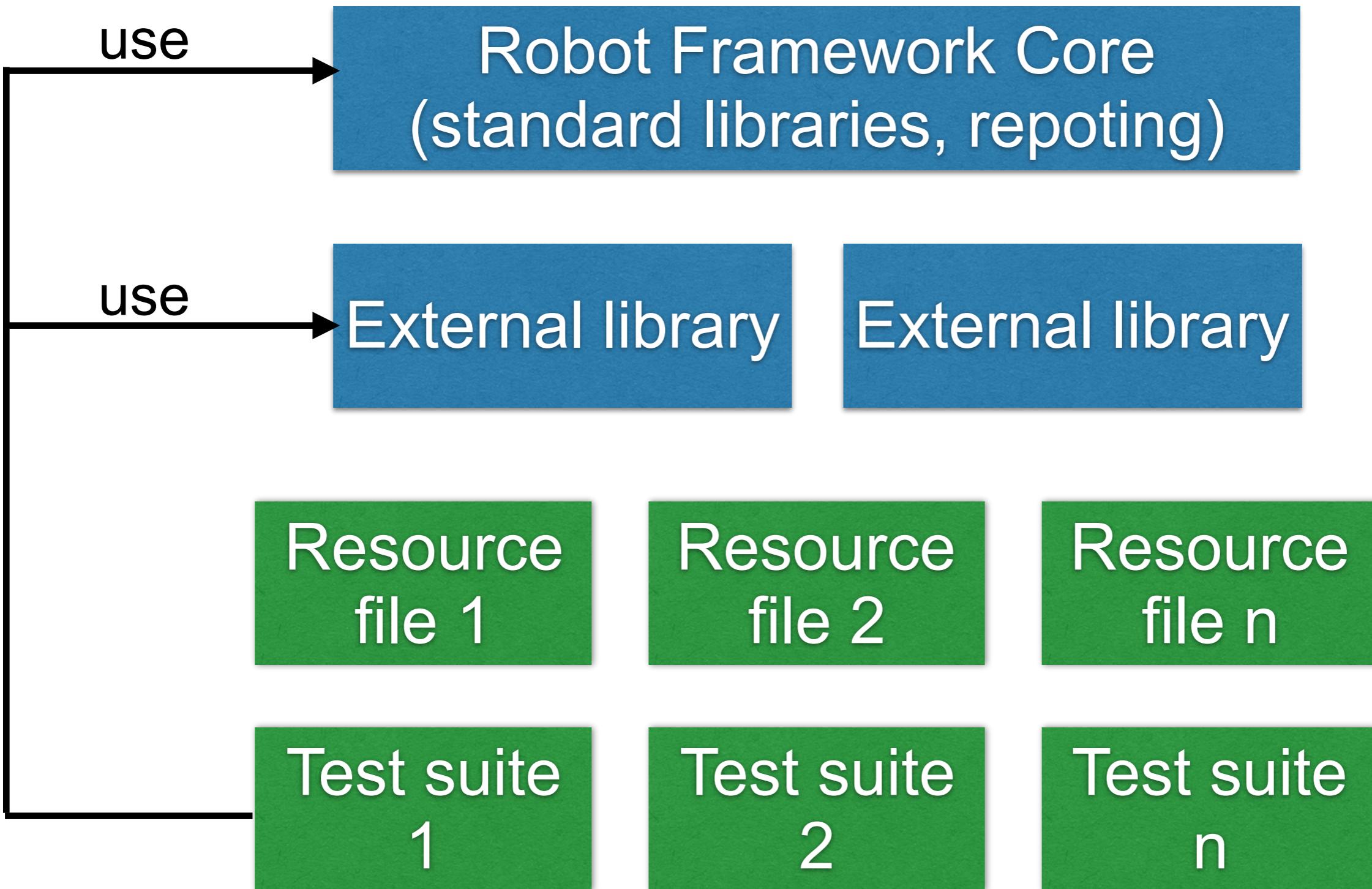
Test suite
1

Test suite
2

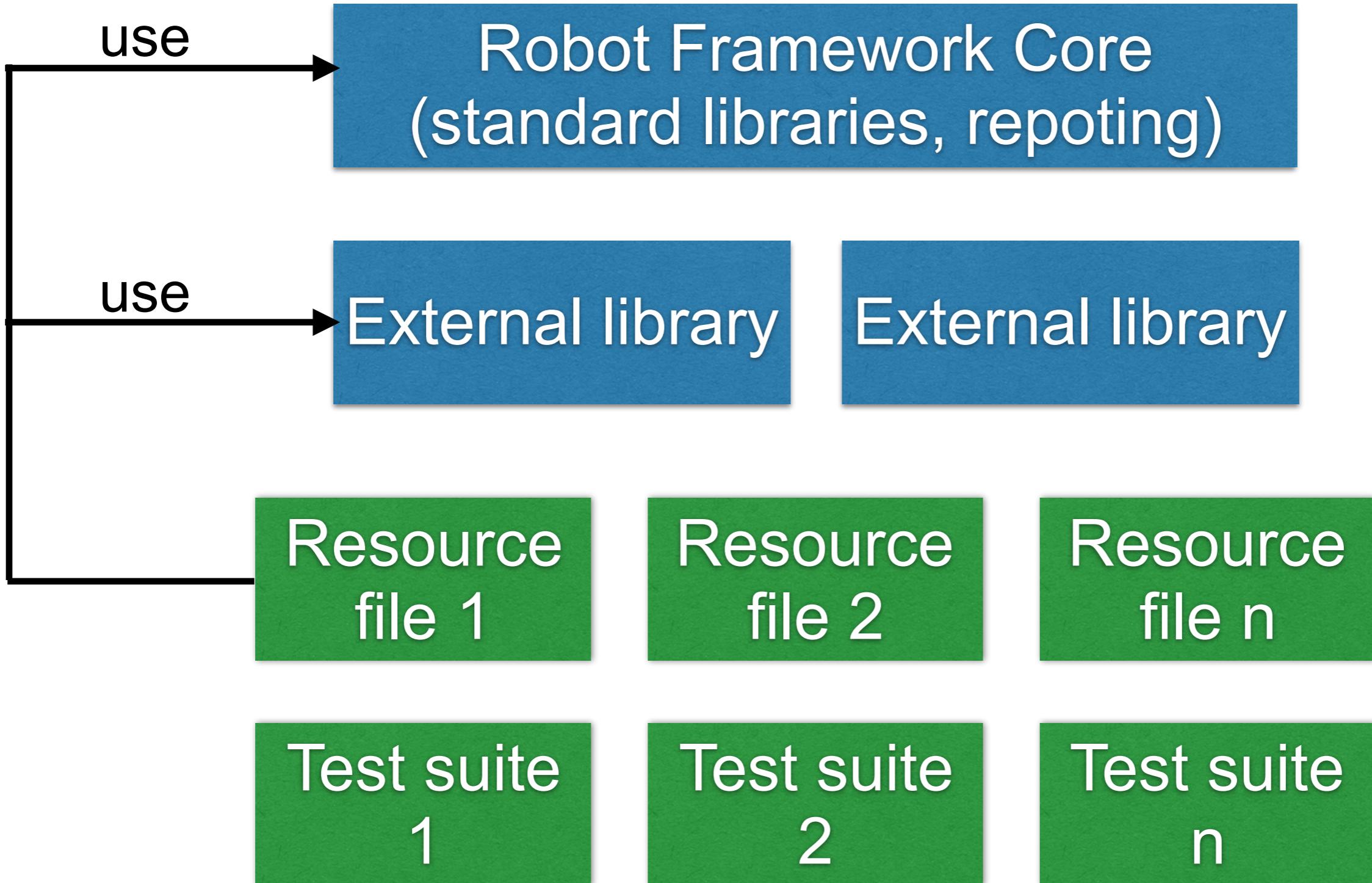
Test suite
n



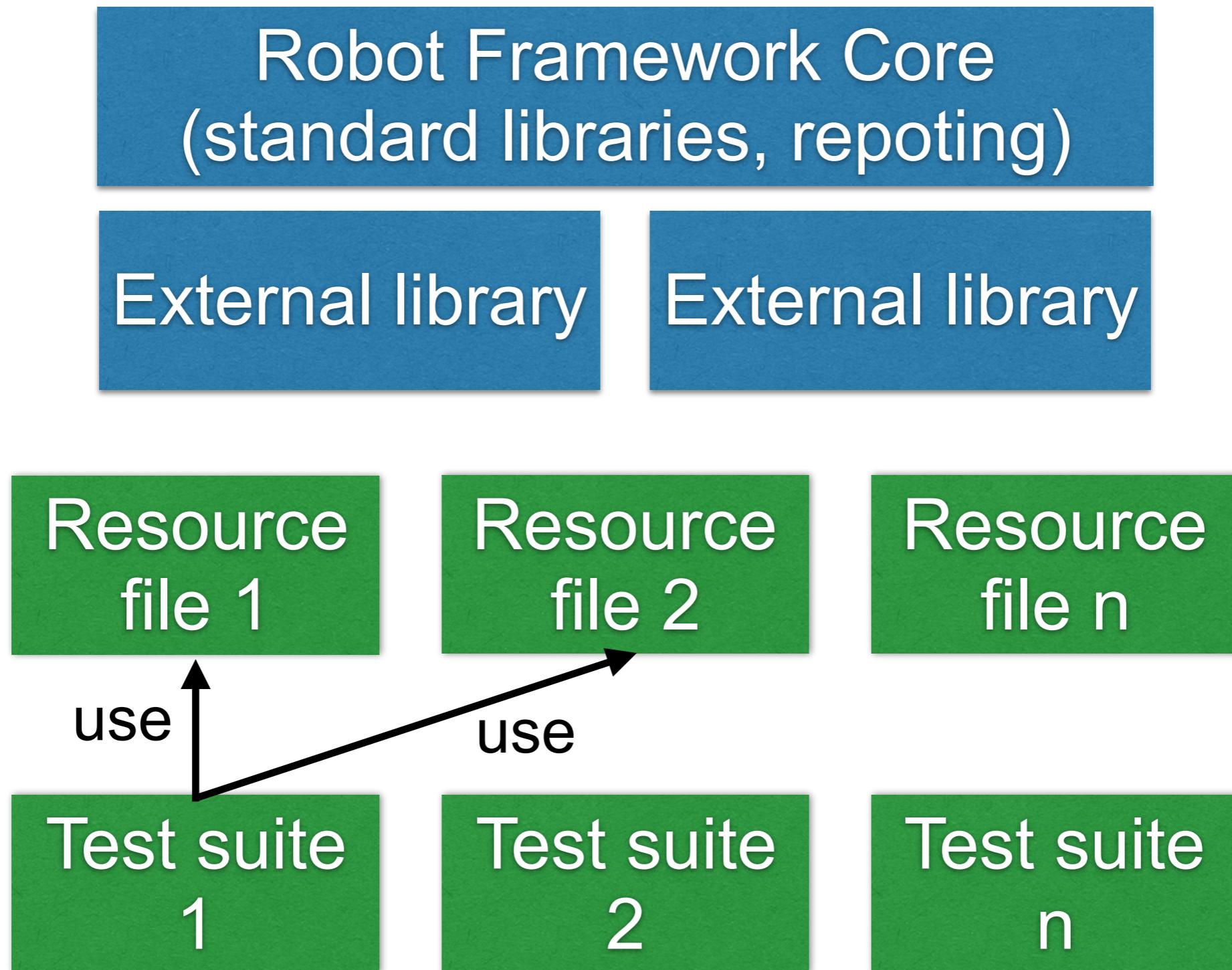
Robot Framework architecture (2)



Robot Framework architecture (3)



Robot Framework architecture (4)



Recommended Libraries

Selenium Library

Web testing library that uses popular Selenium tool internally

HTTP Library (requests)

Library for HTTP level testing using Request internally

REST Instance

Robot Framework test library for HTTP JSON APIs

Faker Library

Library for Faker, a fake test data generator



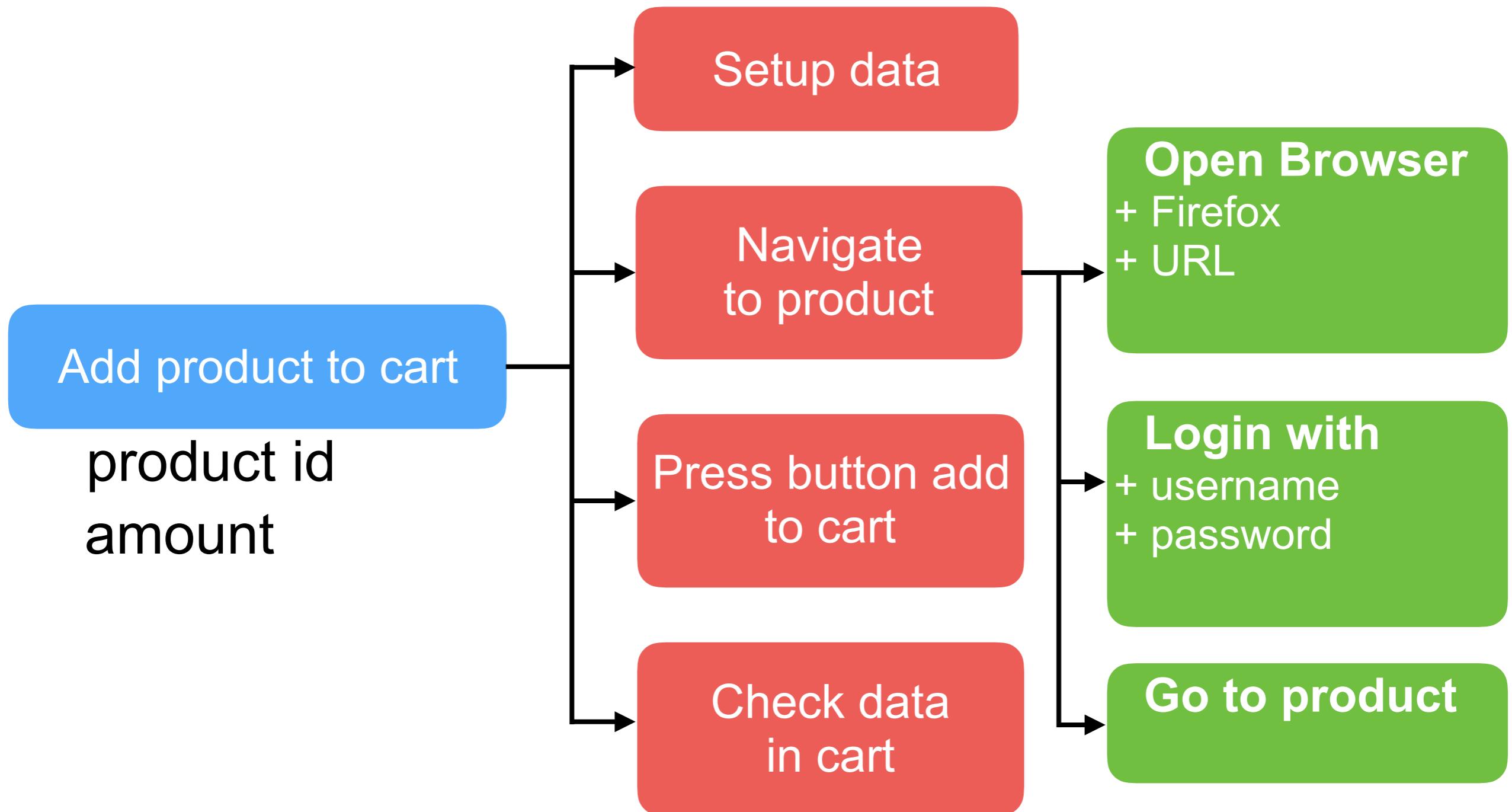
Keyword-driven development

“Add product to cart”

The screenshot shows the Amazon product page for the book "Robot Framework Test Automation" by Sumit Bisht. The page includes the Amazon header with navigation links like Departments, Prime, Video, Music, and a search bar. The main content features the book's title, author, and a 4-star rating from 9 reviews. A "Look inside" button is visible. The book cover image shows a grid of mechanical components. Below the cover, the subtitle "Community Experience Distilled" and the tagline "Create test suites and automated acceptance tests from scratch" are displayed. The product details section shows Kindle (\$10.52), Paperback (\$29.99), and Other Sellers (from \$23.44). The Paperback option is highlighted. The "Buy new" section indicates the item is in stock and ships from Amazon.com with gift-wrap available. A note specifies shipping to Samsennai, Thailand, with options for 15 hrs 29 mins via AmazonGlobal Priority Shipping. An "Add to Cart" button is prominently displayed at the bottom right.



Add product to cart



Working with web browser



Working with Web Browser

Robot Framework IDE/Editor



Robot Framework

Build-in
library

Selenium
library

Faker
library

Python

PIP



Web Driver (ChromeDriver, GeckoDriver, IEDriver)

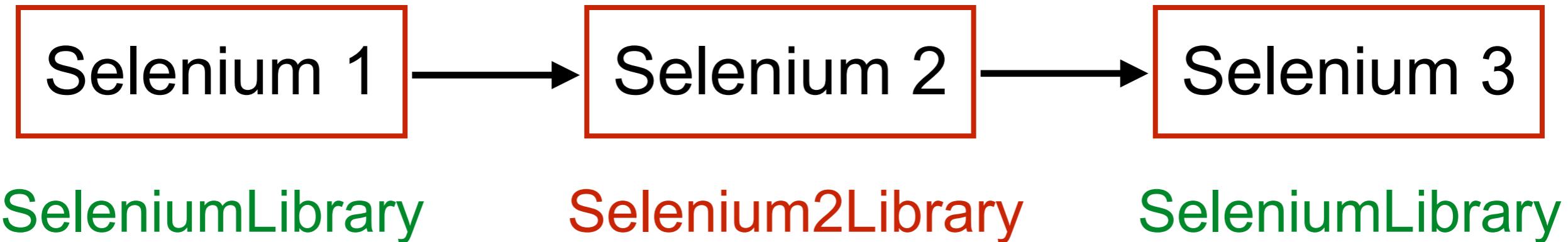


Web Browser (Google Chrome, Firefox, IE)



Install SeleniumLibrary

\$pip install robotframework-seleniumlibrary



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



SeleniumLibrary keywords

Checkbox Should Be Selected · **C**heckbox Should Not Be Selected · **C**hoose File · **C**lear Element Text · **C**lick Button · **C**lick Element · **C**lick Element At Coordinates · **C**lick Image · **C**lick Link · **C**lose All Browsers · **C**lose Browser · **C**lose Window · **C**over Element · **C**reate Webdriver · **C**urrent Frame Should Contain · **C**urrent Frame Should Not Contain · **D**elete All Cookies · **D**elete Cookie · **D**ouble Click Element · **D**rag And Drop · **D**rag And Drop By Offset · **E**lement Attribute Value Should Be · **E**lement Should Be Disabled · **E**lement Should Be Enabled · **E**lement Should Be Focused · **E**lement Should Be Visible · **E**lement Should Contain · **E**lement Should Not Be Visible · **E**lement Should Not Contain · **E**lement Text Should Be · **E**lement Text Should Not Be · **E**xecute Async Javascript · **E**xecute Javascript · **F**rame Should Contain · **G**et All Links · **G**et Browser Aliases · **G**et Browser Ids · **G**et Cookie · **G**et Cookies · **G**et Element Attribute · **G**et Element Count · **G**et Element Size · **G**et Horizontal Position · **G**et List Items · **G**et Location · **G**et Locations · **G**et Selected List Label · **G**et Selected List Labels · **G**et Selected List Value · **G**et Selected List Values · **G**et Selenium Implicit Wait · **G**et Selenium Speed · **G**et Selenium Timeout · **G**et Session Id · **G**et Source · **G**et Table Cell · **G**et Text · **G**et Title · **G**et Value · **G**et Vertical Position · **G**et WebElement · **G**et WebElements · **G**et Window Handles · **G**et Window Identifiers · **G**et Window Names · **G**et Window Position · **G**et Window Size · **G**et Window Titles · **G**o Back · **G**o To · **H**andle Alert · **I**nput Password · **I**nput Text · **I**nput Text Into Alert · **L**ist Selection Should Be · **L**ist Should Have No Selections · **L**ocation Should Be · **L**ocation Should Contain · **L**ocator Should Match X Times · **L**og Location · **L**og Source · **L**og Title · **M**aximize Browser Window · **M**ouse Down · **M**ouse Down On Image · **M**ouse Down On Link · **M**ouse Out · **M**ouse Over · **M**ouse Up · **O**pen Browser · **O**pen Context Menu · **P**age Should Contain · **P**age Should Contain Button · **P**age Should Contain Checkbox · **P**age Should Contain Element · **P**age Should Contain Image · **P**age Should Contain Link · **P**age Should Contain List · **P**age Should Contain Radio Button · **P**age Should Contain Textfield · **P**age Should Not Contain · **P**age Should Not Contain Button · **P**age Should Not Contain Checkbox · **P**age Should Not Contain Element · **P**age Should Not Contain Image · **P**age Should Not Contain Link · **P**age Should Not Contain List · **P**age Should Not Contain Radio Button · **P**age Should Not Contain Textfield · **P**ress Key · **P**ress Keys · **R**adio Button Should Be Set To · **R**adio Button Should Not Be Selected · **R**egister Keyword To Run On Failure · **R**eload Page · **R**emove Location Strategy · **S**croll Element Into View · **S**elect All From List · **S**elect Checkbox · **S**elect Frame · **S**elect From List By Index · **S**elect From List By Label · **S**elect From List By Value · **S**elect Radio Button · **S**elect Window · **S**et Browser Implicit Wait · **S**et Focus To Element · **S**et Screenshot Directory · **S**et Selenium Implicit Wait · **S**et Selenium Speed · **S**et Selenium Timeout · **S**et Window Position · **S**et Window Size · **S**imulate Event · **S**ubmit Form · **S**witch Browser · **S**witch Window · **T**able Cell Should Contain · **T**able Column Should Contain · **T**able Footer Should Contain · **T**able Header Should Contain · **T**able Row Should Contain · **T**able Should Contain · **T**extarea Should Contain · **T**extarea Value Should Be · **T**extfield Should Contain · **T**extfield Value Should Be · **T**itle Should Be · **U**nselect All From List · **U**nselect Checkbox · **U**nselect Frame · **U**nselect From List By Index · **U**nselect From List By Label · **U**nselect From List By Value · **W**ait For Condition · **W**ait Until Element Contains · **W**ait Until Element Does Not Contain · **W**ait Until Element Is Enabled · **W**ait Until Element Is Not Visible · **W**ait Until Element Is Visible · **W**ait Until Location Contains · **W**ait Until Location Is · **W**ait Until Page Contains · **W**ait Until Page Contains Element · **W**ait Until Page Does Not Contain · **W**ait Until Page Does Not Contain Element

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



Open Browser

Browser	Name(s)
Firefox	firefox, ff
Google Chrome	googlechrome, chrome, gc
Headless Firefox	headlessfirefox
Headless Chrome	headlesschrome
Internet Explorer	internetexplorer, ie
Edge	edge
Safari	safari
Opera	opera
Android	android
Iphone	iphone
PhantomJS	phantomjs
HTMLUnit	htmlunit
HTMLUnit with Javascript	htmlunitwithjs

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



Locator strategies

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>
jquery	jQuery expression.	<code>jquery:div.example</code>
default	Keyword specific default behavior.	<code>default:example</code>

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



Verify elements in browser

Using keyword that have prefix “Wait Until”

Wait Until Element Contains	<code>locator, text, timeout=None, error=None</code>	Waits until the element <code>locator</code> contains <code>text</code> . Fails if <code>timeout</code> expires before the text appears. See the Timeouts section for details about the locator syntax. <code>error</code> can be used to override the default error message.
Wait Until Element Does Not Contain	<code>locator, text, timeout=None, error=None</code>	Waits until the element <code>locator</code> does not contain <code>text</code> . Fails if <code>timeout</code> expires before the text disappears. See the Timeouts section for details about the locator syntax. <code>error</code> can be used to override the default error message.
Wait Until Element Is Enabled	<code>locator, timeout=None, error=None</code>	Waits until the element <code>locator</code> is enabled. Element is considered enabled if it is not disabled nor read-only. Fails if <code>timeout</code> expires before the element is enabled. See the Timeouts and elements section for details about the locator syntax. <code>error</code> can be used to override the default error message. Considering read-only elements to be disabled is a new feature in Selenium 3.0.
Wait Until Element Is Not Visible	<code>locator, timeout=None, error=None</code>	Waits until the element <code>locator</code> is not visible. Fails if <code>timeout</code> expires before the element is not visible. See the Timeouts and elements section for details about the locator syntax.

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



First test case

Create file => **google_success.robot**

***** Settings *****

***** Variables *****

***** Test Cases *****

***** Keywords *****



Run test and see result

\$robot google_success.robot



Working with Selenium

*** Settings ***

Library SeleniumLibrary

*** Variables ***

\${URL} http://www.google.com

*** Test Cases ***

ค้นหาคำว่า เหล็กไฟล

[Tags] done

เข้าไปยังหน้าค้นหาของ google

ค้นหาคำว่า "เหล็กไฟล"

จะต้องเจอ "เหล็กไฟล" นะ

*** Keywords ***

ค้นหาคำว่า "\${keyword}"

Input Text

name:q

\${keyword}

Press Keys

name:q

RETURN



Setup web browser driver

Web Browser	Driver
Google Chrome	ChromeDriver
FireFox	GeckoDriver
Internet Explorer	InternetExplorerDriver
Microsoft Edge	EdgeDriver



eg. ChromeDriver

```
$set WEB_DRIVER=<path of driver>
$set PATH=.;%WEB_DRIVER%;%PATH%
```

<https://chromedriver.chromium.org/downloads>



Reporting

Open file report.html

Comments Api Report

Generated
20190925 16:13:40 UTC+07:00
13 days 20 hours ago

Summary Information

Status: All tests passed
Start Time: 20190925 16:13:40.553
End Time: 20190925 16:13:40.961
Elapsed Time: 00:00:00.408
Log File: log.html

Test Statistics

	Total	Pass	Fail	Elapsed	Pass / Fail
Critical Tests	1	1	0	00:00:00	P
All Tests	1	1	0	00:00:00	P

Statistics by Tag

	Total
No Tags	

Statistics by Suite

	Total
Comments Api	1

Test Details

Totals Tags Suites Search

Type: Critical Tests All Tests

Comments Api Log

Generated
20190925 16:13:40 UTC+07:00
13 days 20 hours ago

Test Statistics

	Total	Pass	Fail	Elapsed	Pass / Fail
Critical Tests	1	1	0	00:00:00	P
All Tests	1	1	0	00:00:00	P

Statistics by Tag

	Total	Pass	Fail	Elapsed	Pass / Fail
No Tags					

Statistics by Suite

	Total	Pass	Fail	Elapsed	Pass / Fail
Comments Api	1	1	0	00:00:00	P

Test Execution Log

- **SUITE** Comments Api

Full Name: Comments Api
Source: /Users/somkiat/data/slide/robotframework/aycap-20190924/comments_api.robot
Start / End / Elapsed: 20190925 16:13:40.553 / 20190925 16:13:40.961 / 00:00:00.408
Status: 1 critical test, 1 passed, 0 failed
1 test total, 1 passed, 0 failed

- **TEST** Flow 1

Full Name: Comments Api.Flow 1
Start / End / Elapsed: 20190925 16:13:40.750 / 20190925 16:13:40.960 / 00:00:00.210
Status: PASS (critical)

+ KEYWORD \${email} = Get Comments
+ KEYWORD BuiltIn.Log To Console \${email}



Variables

Define global variables in test suites

```
*** Variables ***
${URL}  http://www.google.com
```



Variables

Set value of variables from command line

*** Variables ***

URL

http://www.google.com



\$robot -v URL:<value> google_success.robot



Build-ins variables

Usage	Description
<code> \${SPACE}</code>	\ \
<code>"\${SPACE}"</code>	" "
<code> \${SPACE * 5}</code>	\ \ \ \ \
<code>"\${SPACE * 5}"</code>	" \ \ \ \ "
<code> \${EMPTY}</code>	\



Tagging test cases

Grouping your test cases

*** Test Cases ***

ค้นหาคำว่า robot

[Tags] done

ค้นหาคำว่า robot

ต้องเจอคำว่า robot

ค้นหาคำว่า thai

[Tags] testing

ค้นหาคำว่า thai

ต้องเจอคำว่า robot



Run tests with specified tags

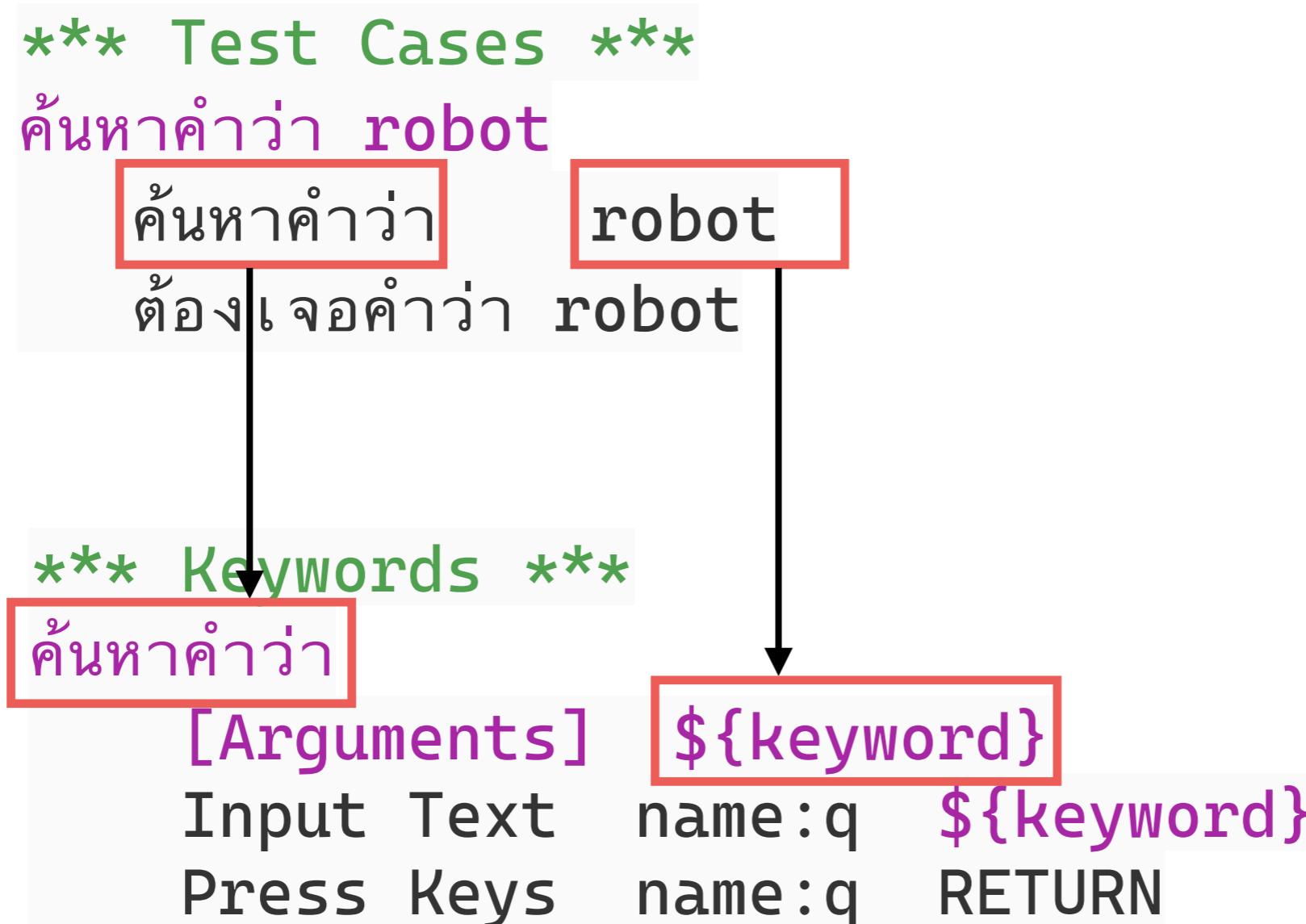
\$robot -i testing google_success.robot

\$robot -e done google_success.robot

-i = Include, -e = Exclude



User keyword arguments



Embedding arguments

*** Test Cases ***

ค้นหาคำว่า robot

ค้นหาคำว่า “robot”

ต้องเจอกำกว่า r~~obot~~bot

*** Keywords ***

ค้นหาคำว่า “\${keyword}”

Input Text name:q \${keyword}

Press Keys name:q RETURN



Improve your test with Test Life Cycle



Test Life Cycle

Suite

Suite Setup

Test Setup
Test case 1
Test Teardown

Test Setup
Test case 2
Test Teardown

Test Setup
Test case 3
Test Teardown

Suite Teardown



Test Life Cycle

*** Settings ***

Suite Setup

เข้า web google

Test Setup

เพิ่มข้อมูลการทดสอบ

Test Teardown

กลับไปยังหน้าแรก

Suite Teardown

Close Browser



Improve your test with Test Template



Test template

One template for all test cases

```
*** Settings ***
Test Template    Search ${keyword} should be found ${expected result}
```

```
*** Test Cases ***
```

Case 01	robot	robot
Case 02	thai	thai

```
*** Keywords ***
```

```
Search ${keyword} should be found ${expected result}
```

ค้นหาคำว่า \${keyword}

ต้องเจอคำว่า robot



Test template

Template per test case

*** Test Cases ***

Search from google

[Template] Flow success to search from google

robot robot

thai thai

*** Keywords ***

Flow success to search from google

[Arguments] \${keyword} \${expected result}

Search \${keyword} shoud be found \${expected result}



Reuse keywords/variables with Resource



Using resource

*** Settings ***

Resource resources/sample.robot

Suite Setup sample.เข้า web google

Suite Teardown Close Browser

1. Create folder = resources
2. Create file = resources/sample.robot

*** Settings ***

Library SeleniumLibrary

Library String

*** Variables ***

\\${URL} http://www.google.com

*** Keywords ***

กลับไปยังหน้าแรก

Go To \\${URL}



Page Object Pattern



Problems

Fragile tests

Bad maintainability

Hard to develop new tests

Hard to understand

Test code duplication

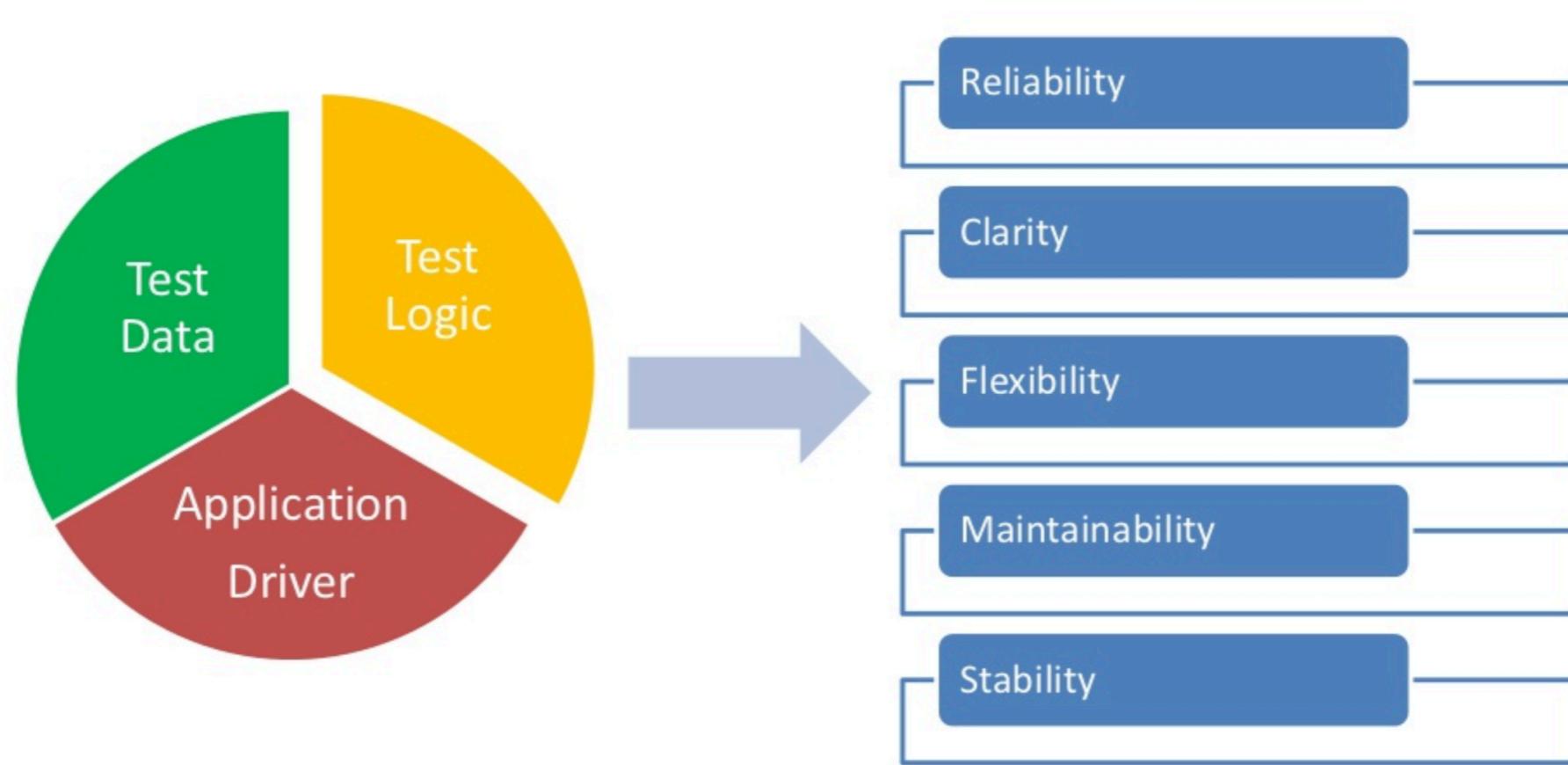
High cost to maintenance tests



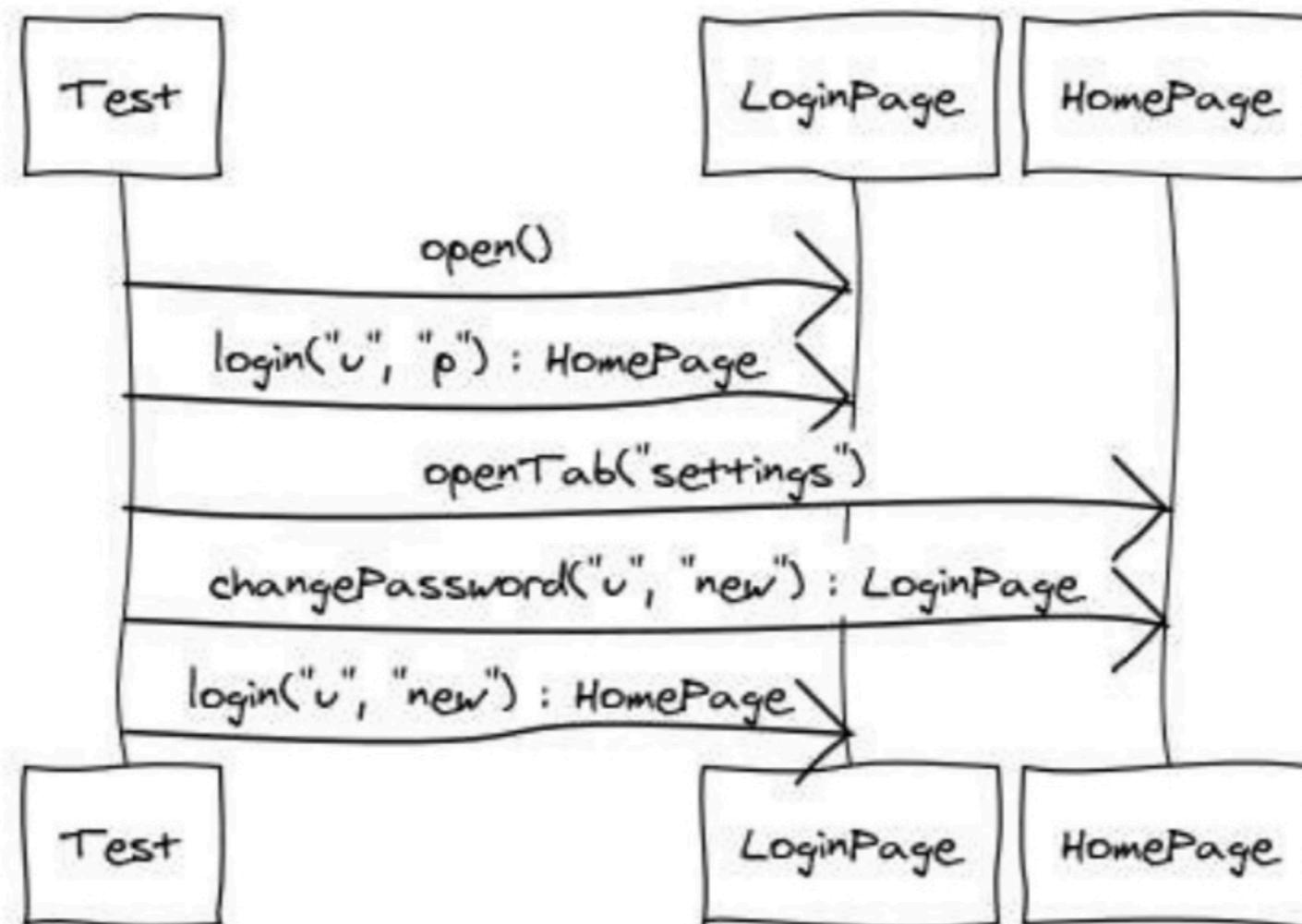
Primary goal ?

To enable reliable stable tests

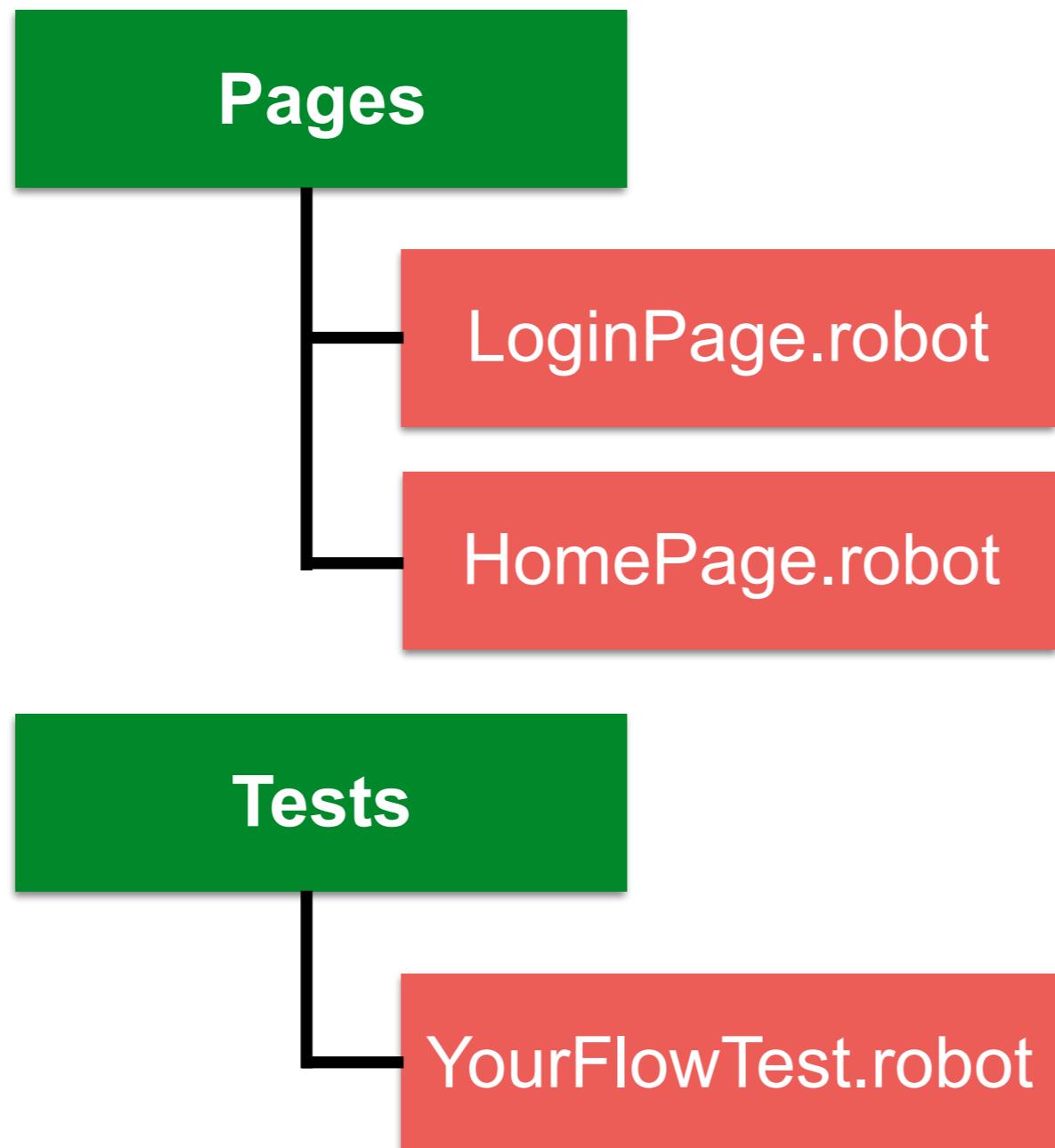
We need to separate all parts as much as possible



Test architecture with Page object



Test architecture with Page object



Tips and Tricks

Basic page with browser details

Expose only what is allowed to do on page

Don't use browser details in tests

Use site map to prepare Page object



Try to improve ...

Reliable tests

Reusable test code

Separation of concepts

Expressive UI structure

Short and clear tests

Tests look more like acceptance tests

Tests are understand by non-technical people

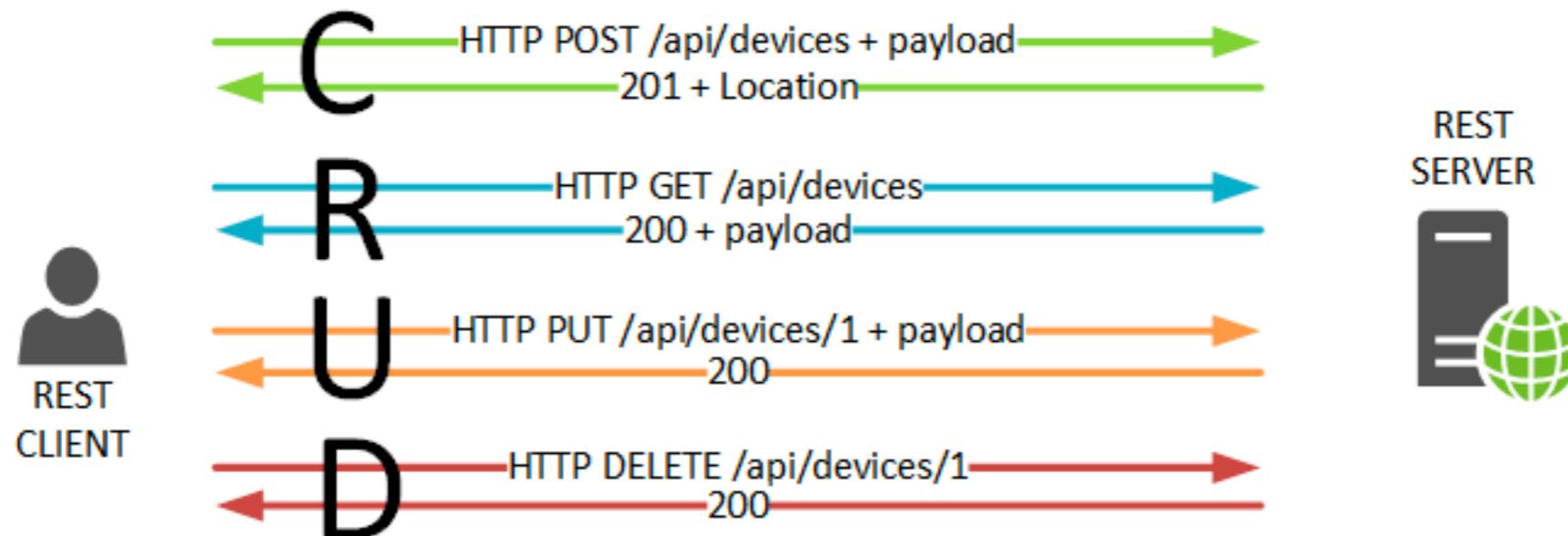


API Testing



Install RequestsLibrary

\$pip install requests
\$pip install robotframework-requests



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



RequestsLibrary

Introduction

RequestsLibrary is a [Robot Framework](#) test library that uses the [Requests](#) HTTP client.

Here is an example testcase

*** Settings ***				
Library	Collections			
Library	RequestsLibrary			
*** Test Cases ***				
Get Requests				
	Create Session	github	http://api.github.com	
	Create Session	google	http://www.google.com	
	\${resp}=	Get Request	google	/
	Should Be Equal As Strings	\${resp.status_code}	200	
	\${resp}=	Get Request	github	/users/bulkan
	Should Be Equal As Strings	\${resp.status_code}	200	
	Dictionary Should Contain Value	\${resp.json()}	Bulkan Savun Evcimen	

Shortcuts

[Create Digest Session](#) · [Create Ntlm Session](#) · [Create Session](#) · [Delete](#) · [Delete All Sessions](#) · [Delete Request](#) · [Get](#) · [Get Request](#) · [Head](#) · [Head Request](#) · [Options](#) · [Options Request](#) · [Patch](#) · [Patch Request](#) · [Post](#) · [Post Request](#) · [Put](#) · [Put Request](#) · [To Json](#)

<https://bulkan.github.io/robotframework-requests/doc/RequestsLibrary.html>



RequestsLibrary keywords

Shortcuts

[Create Client Cert Session](#) · [Create Custom Session](#) · [Create Digest Session](#) · [Create Ntlm Session](#) ·
[Create Session](#) · [Delete All Sessions](#) · [Delete Request](#) · [Get Request](#) · [Head Request](#) · [Options Request](#) ·
[Patch Request](#) · [Post Request](#) · [Put Request](#) · [Session Exists](#) · [To Json](#) · [Update Session](#)

<https://bulkan.github.io/robotframework-requests/doc/RequestsLibrary.html>



Create tests

Connect to service and check status code

```
*** Settings ***
```

```
Library RequestsLibrary
```

```
*** Test Case ***
```

```
Success with /users
```

```
Create Session jph https://jsonplaceholder.typicode.com
```

```
    ${response}= Get Request jph /users
```

```
Should Be Equal ${response.status_code} ${200}
```



Create tests

Check size of result from service

*** Test Case ***

Success with /users

```
Create Session    jph  https://jsonplaceholder.typicode.com
${response}=  Get Request  jph  /users
Should Be Equal  ${response.status_code}  ${200}
```

```
 ${json}=  Set Variable  ${response.json()}
Length Should Be  ${json}  10
```



Create tests

Check first user data from result

*** Test Case ***

Success with /users

```
Create Session    jph  https://jsonplaceholder.typicode.com
${response}=  Get Request  jph  /users
Should Be Equal  ${response.status_code}  ${200}
```

```
${json}=  Set Variable  ${response.json()}
Length Should Be  ${json}  10
```

```
Should Be Equal  ${json[0]["name"]}  Leanne Graham
Should Be Equal  ${json[0]["username"]}  Bret
```



Working with HTTP POST

Add product id=1000 to empty basket

```
Create Session      baskets      http://localhost:8882
&{headers}= Create Dictionary Content-Type=application/json
&{data}= Create Dictionary
...  product_id=${1000}
...  product_name=Adidas
...  product_price=${1500}
...  product_image=http://xxx.jpg
...  quantity=${1}
${response}= Post Request baskets /baskets
...  data=${data}  headers=${headers}
Should Be Equal As Strings ${response.status_code} 200
```



Parallel testing with pabot



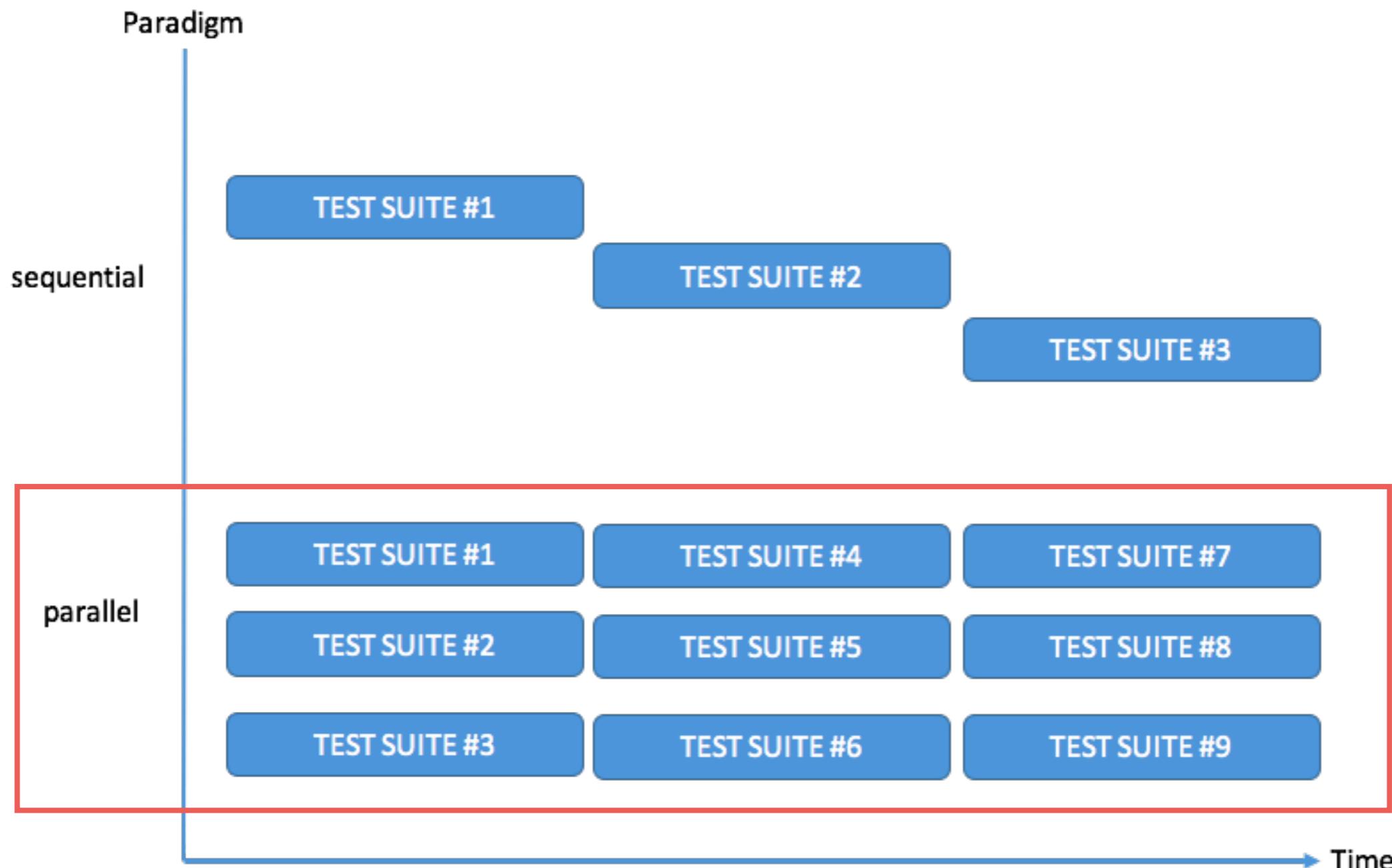
Problems

More test cases more problems

More test cases more testing time

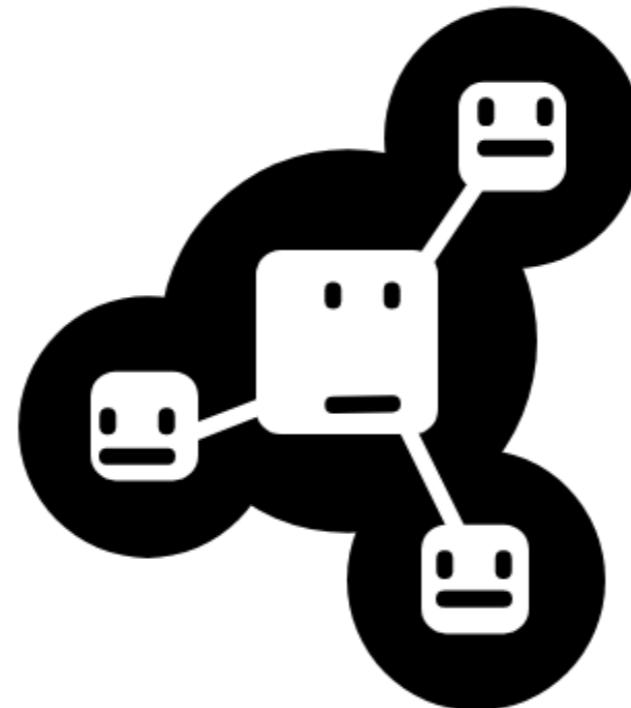


Parallel testing



Install Pabot

```
$pip install robotframework-pabot  
$pabot
```



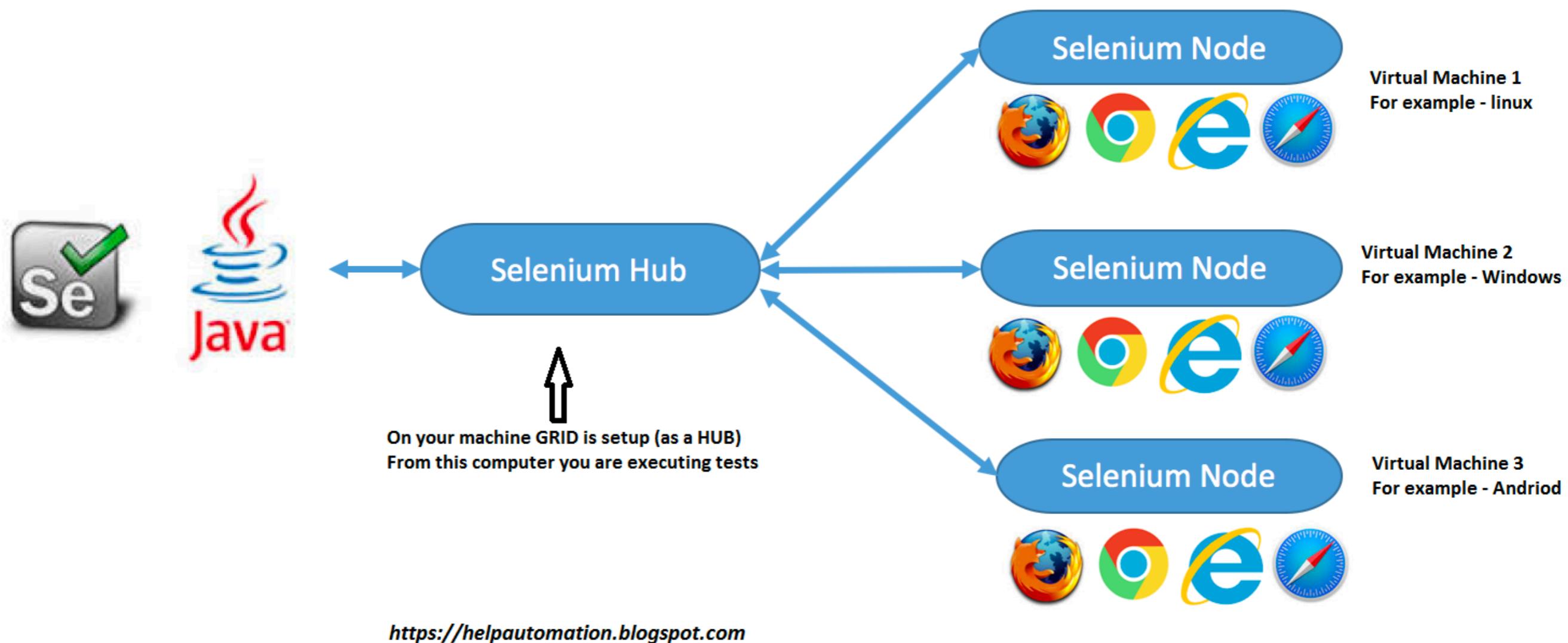
<https://pabot.org/>



Scaling UI testing with Selenium Grid



Selenium Grid



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



Using from Robot Framework

```
Open Browser    http://localhost:8080/demo/
...  browser=chrome
...  remote_url=http://localhost:4444/wd/hub
...  desired_capabilities=browserName:chrome
```



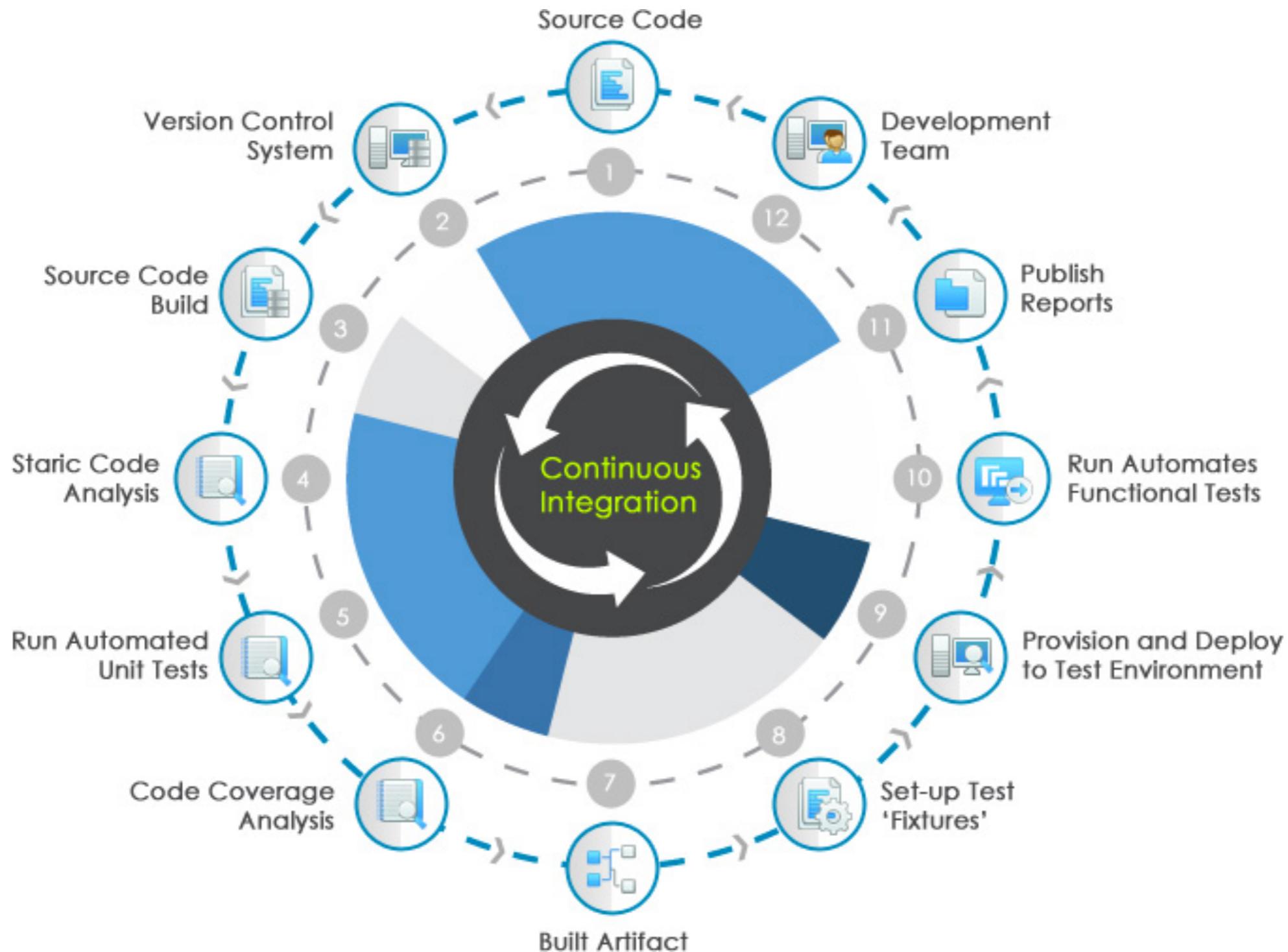
Continuous Testing



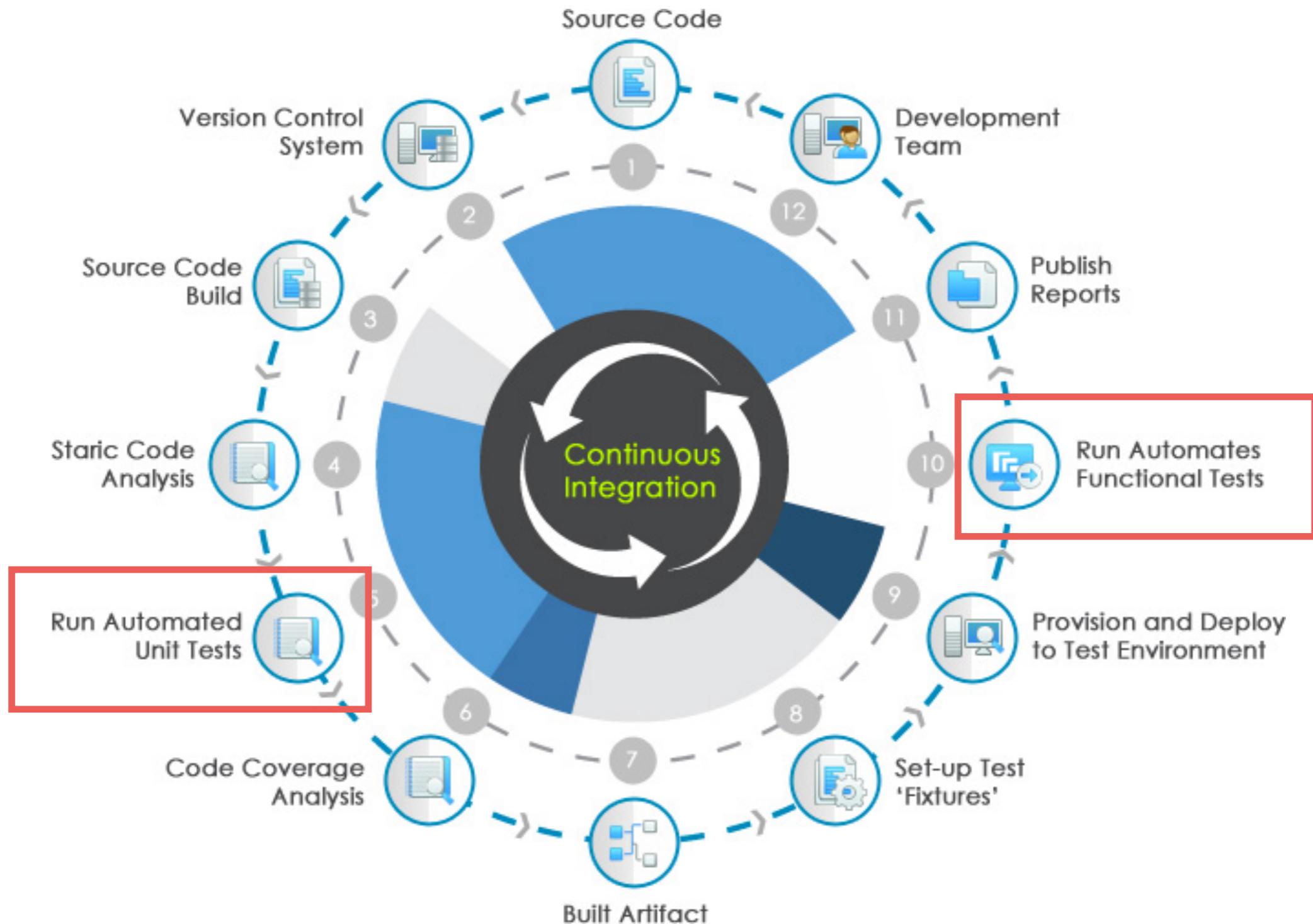
<https://jenkins.io/>



Continuous Integration



Continuous Integration



Continuous Integration

Discipline to integrate frequently



Continuous Integration

Strive to make **small change**

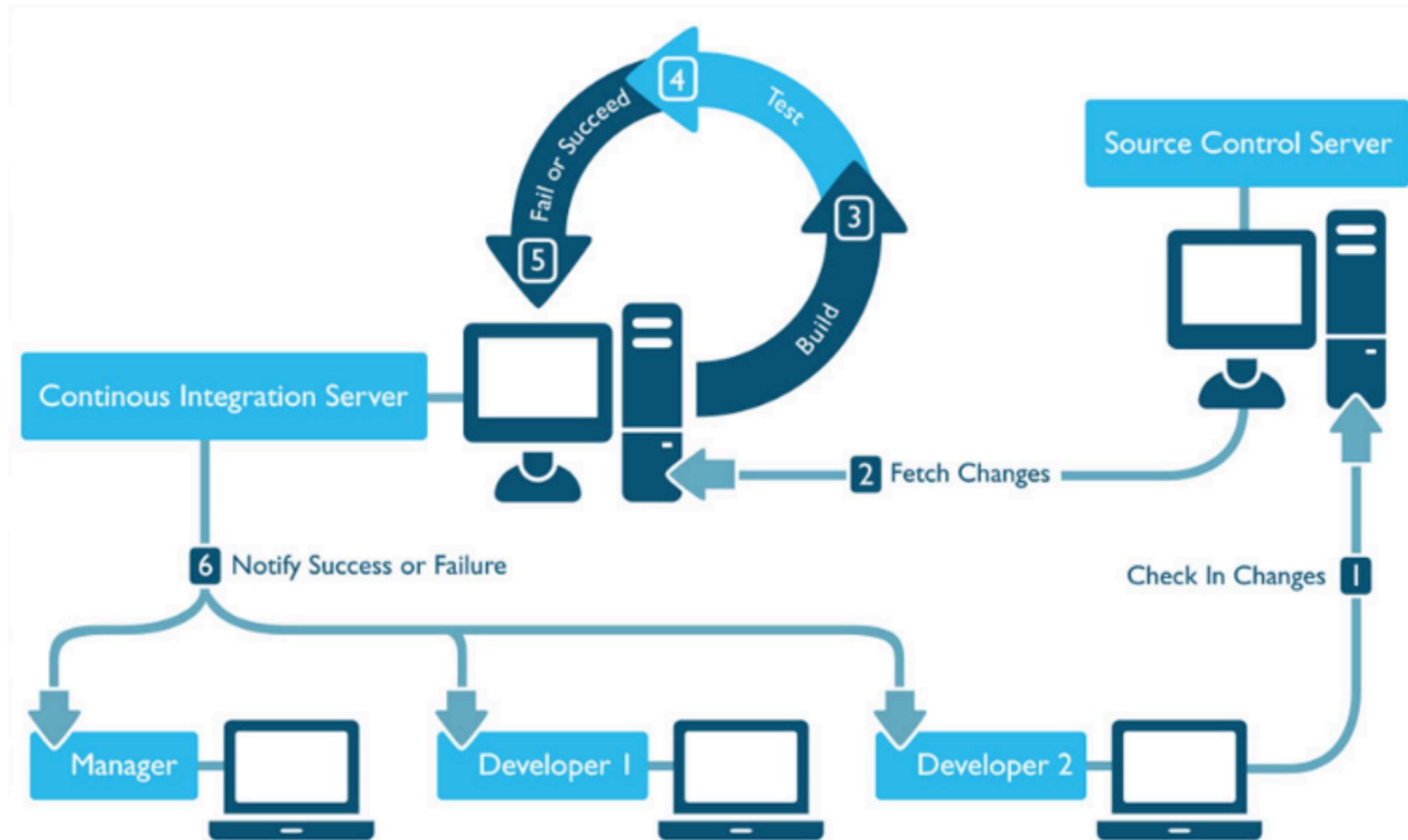


Continuous Integration

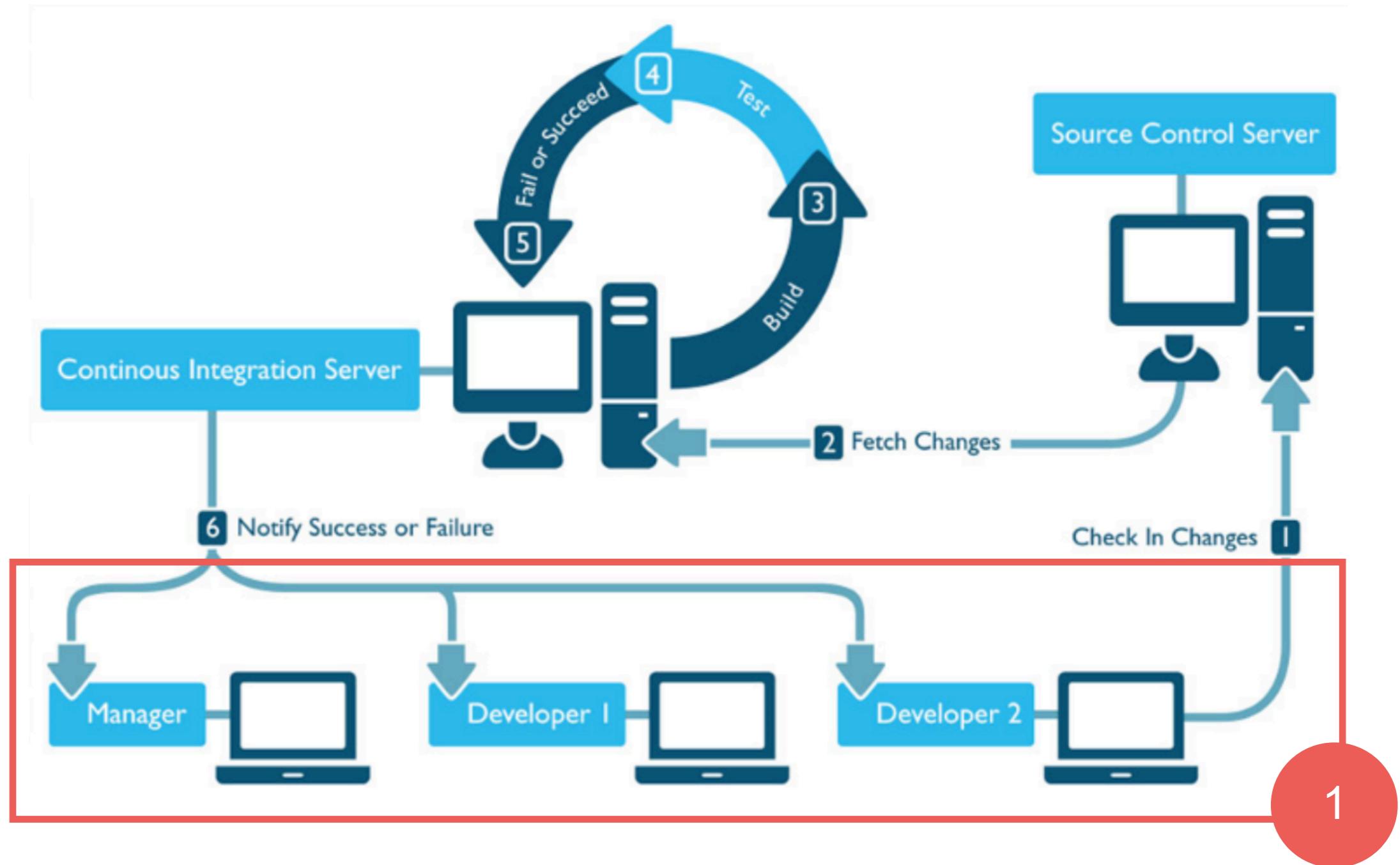
Strive for **fast feedback**



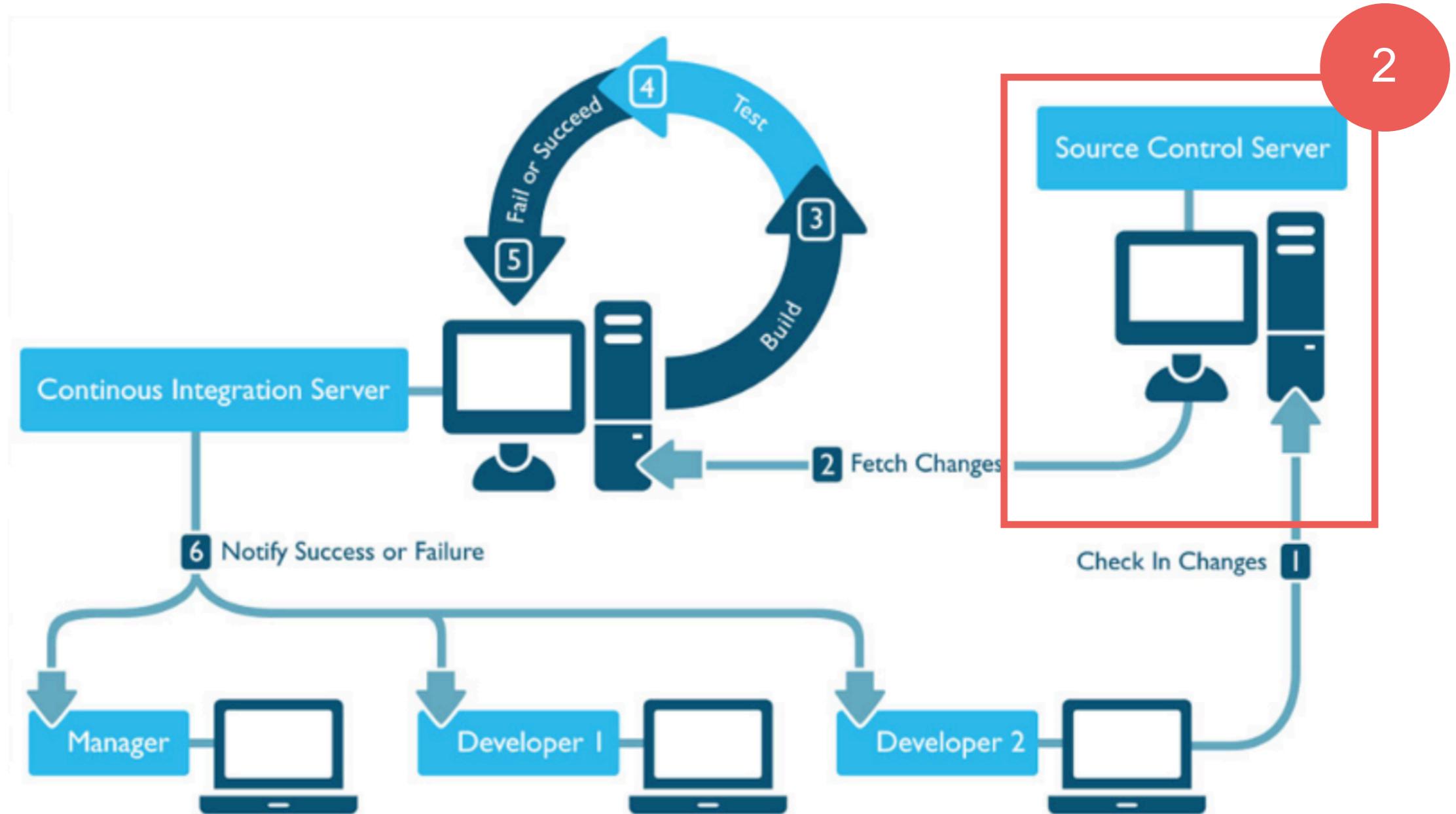
Continuous Integration and Testing



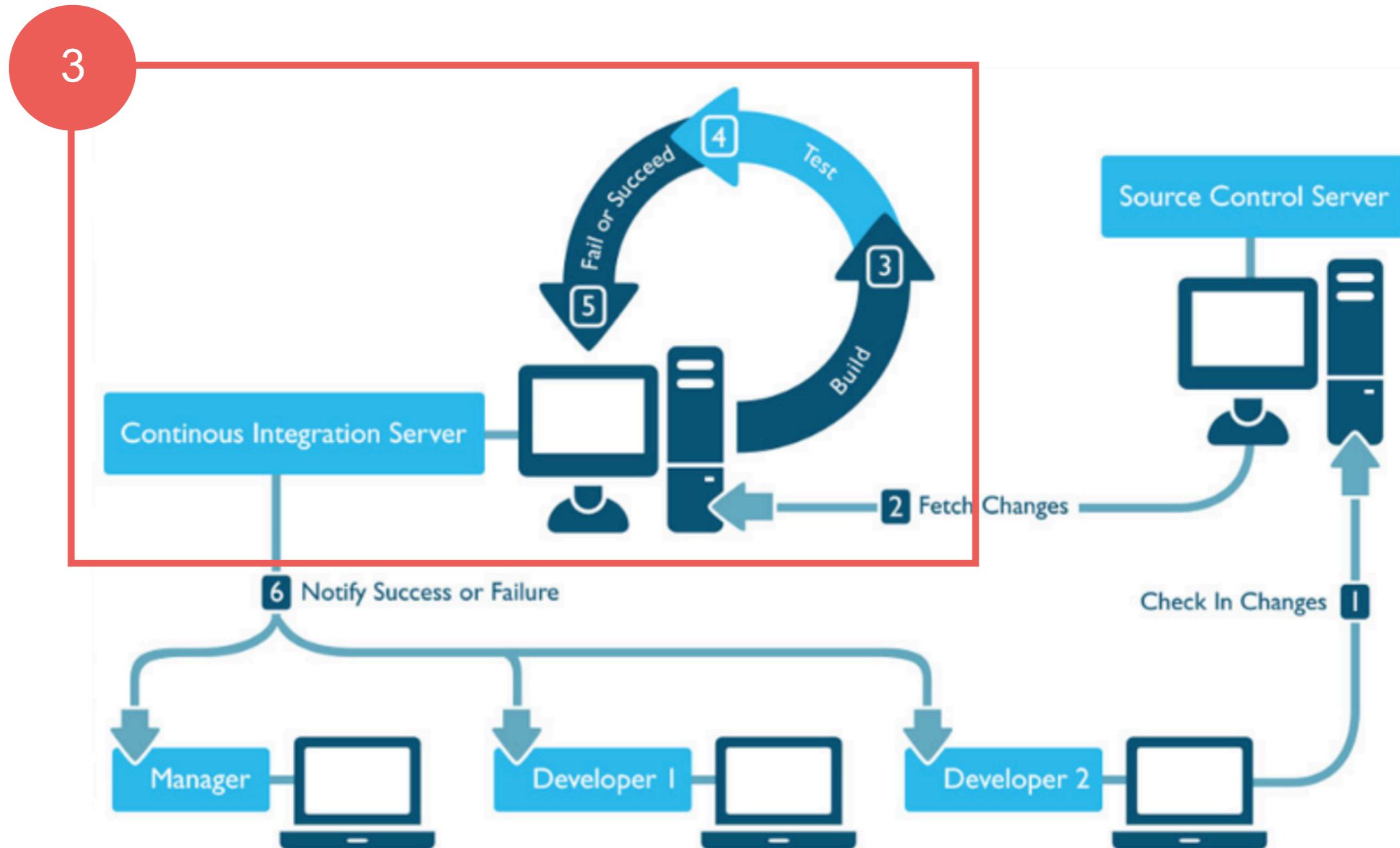
Continuous Integration and Testing



Continuous Integration and Testing



Continuous Integration and Testing



Start Jenkins server

```
$java -jar jenkins.war
```

Jenkins initial setup is required. An admin user
been created and a password generated.
Please use the following password to proceed to :
llation:

```
07be76a843f54c3bae1d259ee38ed2d9
```

This may also be found at: /Users/somkiat/data/s
robot-framework/thaibev-20180806/ci/v2/secrets/i
lAdminPassword



Open in browser

http://localhost:8080

The screenshot shows the Jenkins 'Getting Started' screen. The main title is 'Unlock Jenkins'. A sub-instruction says: 'To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:'. Below this, a code block highlights the path: '/Users/somkiat/data/slide/robot-framework/thaibev-20180806/ci/v2/secrets/initialAdminPassword'. A placeholder for the 'Administrator password' is provided, and a 'Continue' button is at the bottom right.

Getting Started

Unlock Jenkins

To ensure Jenkins is securely set up by the administrator, a password has been written to the log ([not sure where to find it?](#)) and this file on the server:

```
/Users/somkiat/data/slide/robot-framework/thaibev-  
20180806/ci/v2/secrets/initialAdminPassword
```

Please copy the password from either location and paste it below.

Administrator password

Continue



Install suggested plugins

Getting Started X

Customize Jenkins

Plugins extend Jenkins with additional features to support many different needs.

**Install suggested
plugins**

Install plugins the Jenkins community finds most useful.

**Select plugins to
install**

Select and install plugins most suitable for your needs.

Jenkins 2.121.3



Ready to start !!

The screenshot shows the Jenkins dashboard. At the top, there's a navigation bar with a cartoon Jenkins head icon and the word "Jenkins". Below it is a sidebar with links: "New Item", "People", "Build History", "Manage Jenkins", "My Views", "Credentials", and "New View". The main content area has a large "Welcome to Jenkins!" message with a call to action: "Please [create new jobs](#) to get started." Below this, there are two sections: "Build Queue" (which says "No builds in the queue.") and "Build Executor Status" (which lists "1 Idle" and "2 Idle").



1. Create new job

Enter an item name

Fill in job name

» Required field

Freestyle project
 This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Pipeline
 Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
 Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder
 Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a separate namespace, so you can have multiple things of the same name as long as they are in different folders.

GitHub Organization
 Scans a GitHub organization (or user account) for all repositories matching some defined markers.

Multibranch Pipeline
 Creates a set of Pipeline projects according to detected branches in one SCM repository.

OK



2. Source code management

General **Source Code Management** Build Triggers Build Environment Build Post-build Actions

Source Code Management

None
 Git
 Subversion 

Build Triggers

Trigger builds remotely (e.g., from scripts) 
 Build after other projects are built 
 Build periodically 
 GitHub hook trigger for GITScm polling 
 Poll SCM 



3. Build trigger

General **Source Code Management** Build Triggers Build Environment Build Post-build Actions

Source Code Management

None
 Git
 Subversion ?

Build Triggers

Trigger builds remotely (e.g., from scripts) ?
 Build after other projects are built ?
 Build periodically ?
 GitHub hook trigger for GITScm polling ?
 Poll SCM ?



4. Build trigger with poll SCM

Poll SCM 

Schedule 

* * * * *

No schedules so will only run due to SCM changes if triggered by a post-commit hook

This field follows the syntax of cron (with minor differences). Specifically, each line consists of 5 fields separated by TAB or whitespace:

MINUTE HOUR DOM MONTH DOW

MINUTE Minutes within the hour (0–59)

HOUR The hour of the day (0–23)

DOM The day of the month (1–31)

MONTH The month (1–12)

DOW The day of the week (0–7) where 0 and 7 are Sunday.

To specify multiple values for one field, the following operators are available. In the order of precedence,

- * specifies all valid values
- M–N specifies a range of values
- M–N/X or */X steps by intervals of X through the specified range or whole valid range
- A, B, . . . , Z enumerates multiple values



5. Build your job

Build

Add build step ▾

- Execute Windows batch command
- Execute shell
- Invoke Ant
- Invoke Gradle script
- Invoke top-level Maven targets
- Run with timeout
- Set build status to "pending" on GitHub commit



6. Post build actions

Post-build Actions

Add post-build action ▾

- Aggregate downstream test results
- Archive the artifacts
- Build other projects
- Publish JUnit test result report
- Publish Robot Framework test results**
- Record fingerprints of files to track usage
- Git Publisher
- E-mail Notification
- Editable Email Notification
- Set GitHub commit status (universal)
- Set build status on GitHub commit [deprecated]
- Delete workspace when build is done



7. Add post build with Robot framework

Post-build Actions

Publish Robot Framework test results

Directory of Robot output

Path to directory containing robot xml and html files (relative to build workspace)

Advanced...

Thresholds for build result

Yellow % **Entry must be percentage value between 0-100**

Blue % **Entry must be percentage value between 0-100**

Use thresholds for critical tests only

Add post-build action ▾



8. Install Robot framework plugin

The screenshot shows the Jenkins 'Manage Jenkins' interface. On the left, there's a sidebar with links: 'People', 'Build History', 'Manage Jenkins' (which is highlighted with a red box and a red circle containing the number 1), 'My Views', 'Credentials', and 'New View'. Below this is a 'Build Queue' section stating 'No builds in the queue.' and a 'Build Executor Status' section showing '1 Idle' and '2 Idle'. The main content area is titled 'Manage Jenkins' and contains several management options, each with an icon and a brief description:

- Configure System**: Configure global settings and paths.
- Configure Global Security**: Secure Jenkins; define who is allowed to access/use the system.
- Configure Credentials**: Configure the credential providers and types.
- Global Tool Configuration**: Configure tools, their locations and automatic installers.
- Reload Configuration from Disk**: Discard all the loaded data in memory and reload everything from file system. Useful when you modify disk.
- Manage Plugins**: Add, remove, disable or enable plugins that can extend the functionality of Jenkins. This option is highlighted with a red box and a red circle containing the number 2.
- System Information**: Displays various environmental information to assist trouble-shooting.
- System Log**: System log captures output from `java.util.logging` related to Jenkins.



8. Install Robot framework plugin

Filter:

Updates Available Installed Advanced

Install ↓	Name	Version
Robot Framework <input type="checkbox"/> This publisher stores Robot Framework test reports for builds and shows summaries of them in project and build views along with trend graph.		1.6.5

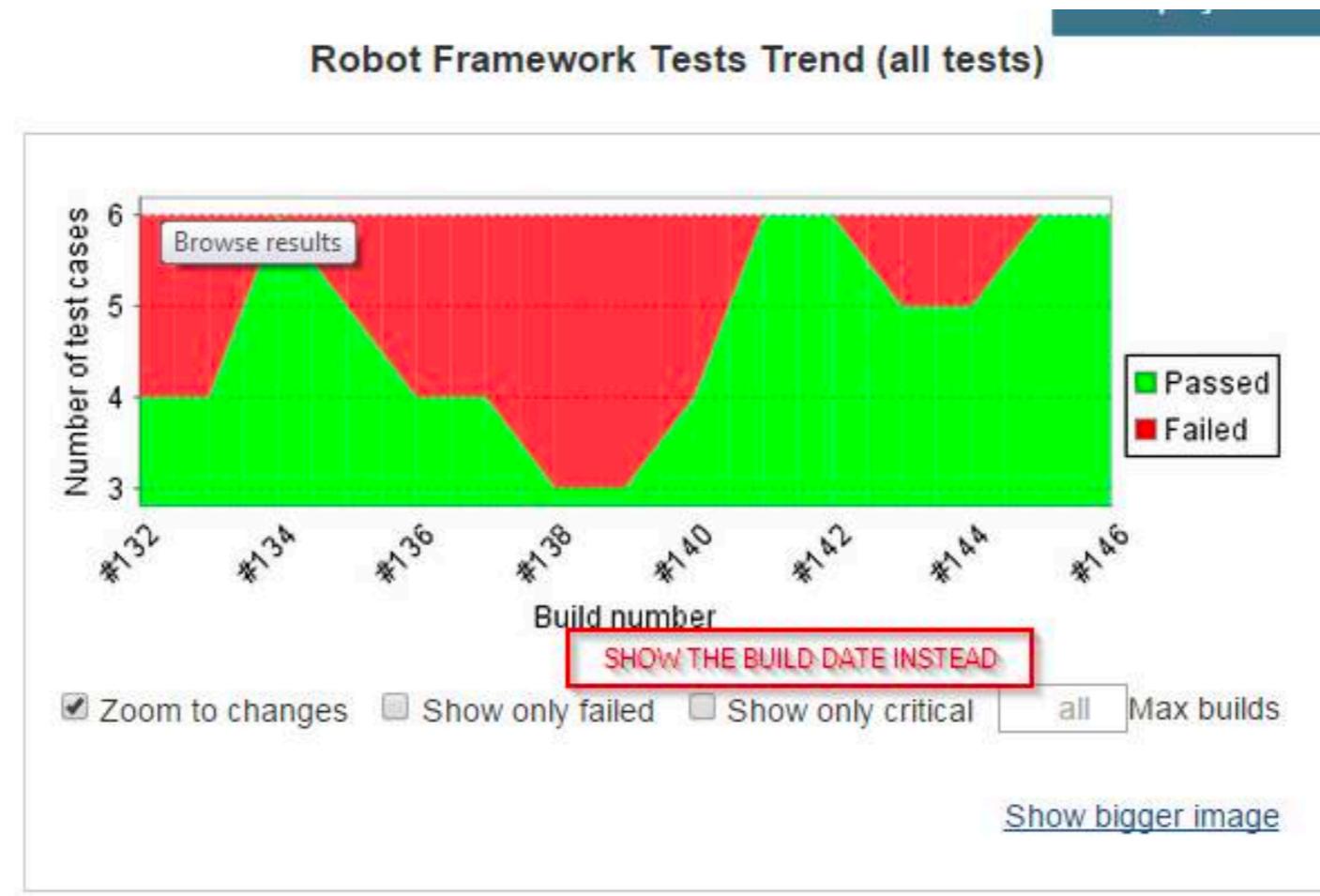
Install without restart Download now and install after restart Check now

Update information obtained: 3 hr 51 min ago

<https://plugins.jenkins.io/robot>



Robot framework report



<https://plugins.jenkins.io/robot>

