Agile Testing, Test Automation & BDD

OCTOBER 25, 2016

SINGAPORE

ORGANIZED BY TESTINGMIND

About Myself

Quick Bio: “Test Automation Consultant having more than 11 years of experience in Software Automated Testing space.”

Twitter: @sahajamait

Github: https://github.com/sahajamit

P.S. : *All the opinions given in this talk are completely personal and has nothing to do with my employer.*

Talk Abstract:

“What goes into the selection of right Test Automation Framework for your application? the Application type (Mobile, Desktop, Web), the scripting language(Java, Ruby, Python) or the tools (Selenium, Appium, UFT). No, there are many more factors to consider before finalising your “ideal” automation framework and if you get this decision wrong then it can have a cascading effect to your entire test strategy. In this rapidly changing Agile environment, the automation framework should be extremely flexible and agnostic of external factors like tools and languages. In this talk we will be covering this subject more deeply with some real life examples.”

Test Automation Frameworks – Assumptions, Concepts and Tools

By : Amit Rawat

Who can help you to build your Test Automation Framework ??

Some Assumptions:

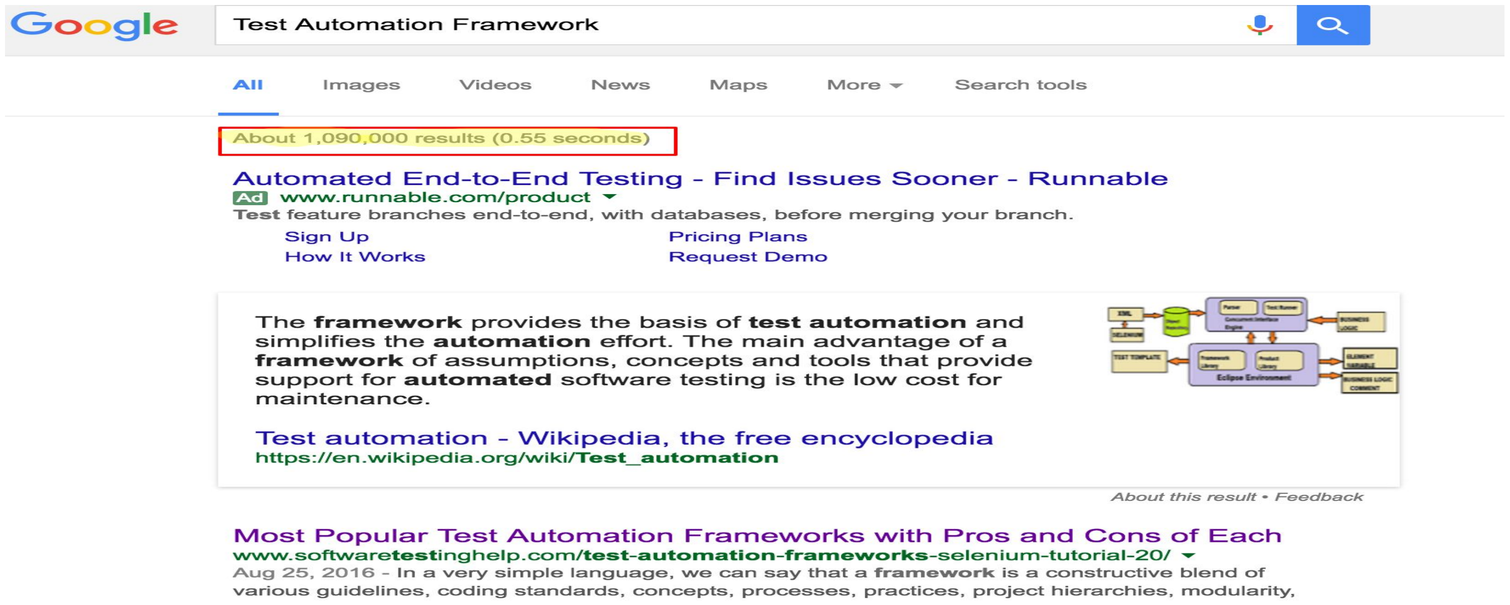
- Why to build (already so many open-source frameworks available)
- I will hire a Selenium/Automation Architect
- I will buy a Licensed tool

Who can help you to build your Test Automation Framework ??

Reality:

How easy is to design your Automation Framework

Google yield more than million results for the query "Test Automation Framework"



Test Automation Framework

All Images Videos News Maps More Search tools

About 1,090,000 results (0.55 seconds)

Automated End-to-End Testing - Find Issues Sooner - Runnable
Ad www.runnable.com/product
Test feature branches end-to-end, with databases, before merging your branch.
Sign Up Pricing Plans
How It Works Request Demo

The **framework** provides the basis of **test automation** and simplifies the **automation** effort. The main advantage of a **framework** of assumptions, concepts and tools that provide support for **automated** software testing is the low cost for maintenance.

Test automation - Wikipedia, the free encyclopedia
https://en.wikipedia.org/wiki/**Test_automation**

About this result • Feedback

Most Popular Test Automation Frameworks with Pros and Cons of Each
www.softwaretestinghelp.com/**test-automation-frameworks**-selenium-tutorial-20/
Aug 25, 2016 - In a very simple language, we can say that a **framework** is a constructive blend of various guidelines, coding standards, concepts, processes, practices, project hierarchies, modularity,

Test Automation Frameworks – Assumptions, Concepts and Tools

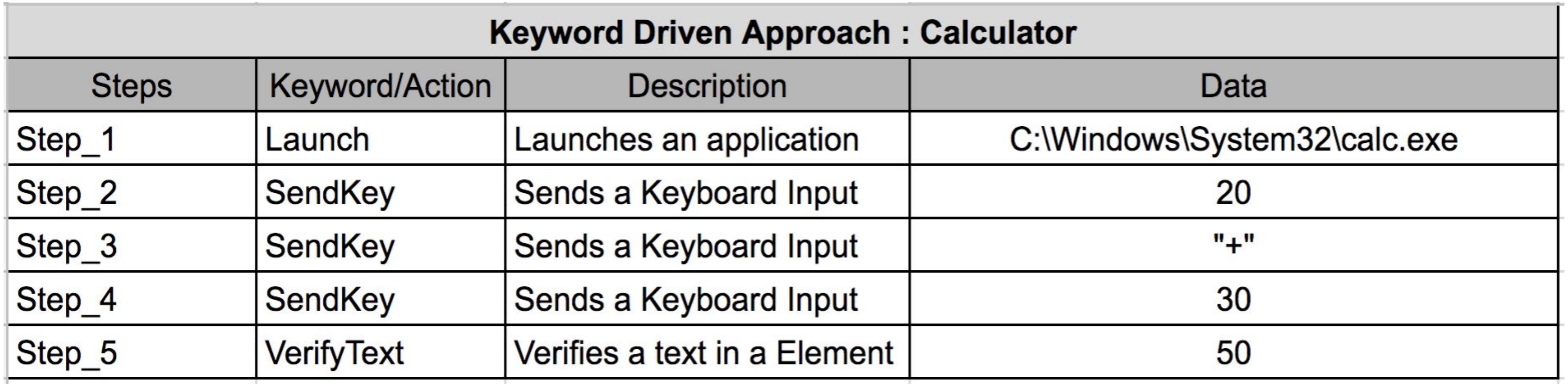
What is a Test Automation Framework?

It is a supporting structure or a harness that provides a conducive environment to execute and maintain the automation scripts effectively. It defines a single standard of doing things which can result in highly-reusable automation scripts and that can lead to very low cost of maintenance.

Some common types of Automation Frameworks

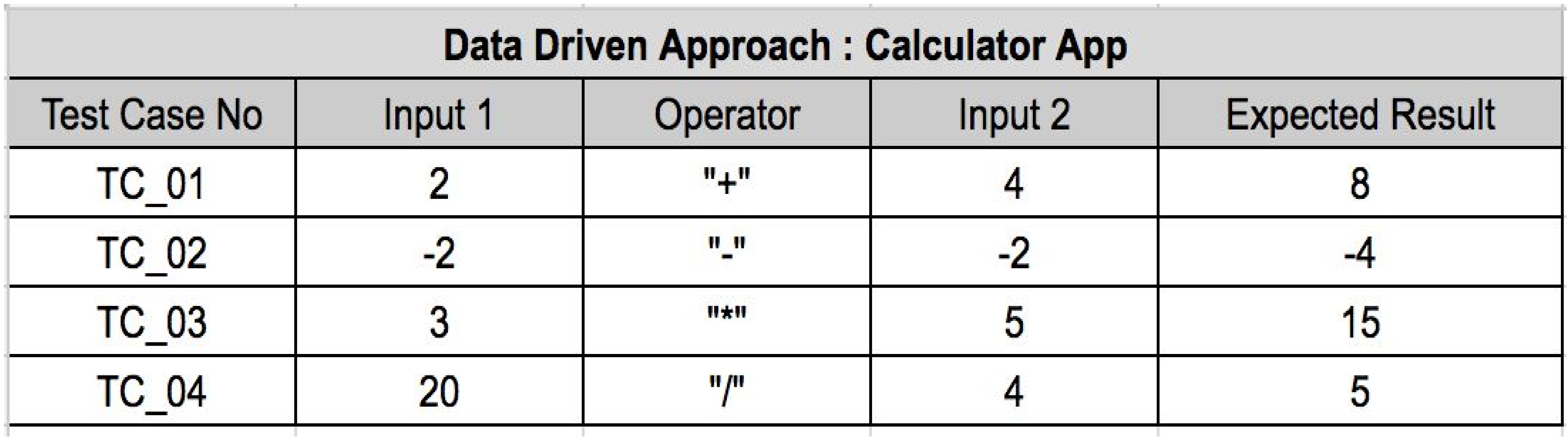
- Linear
- Test Script Modularity
- Keyword-driven
- Behavior-driven(BDD)
- Hybrid
- Agile Automation Framework

Keyword Driven Approach (Script Less Automation)



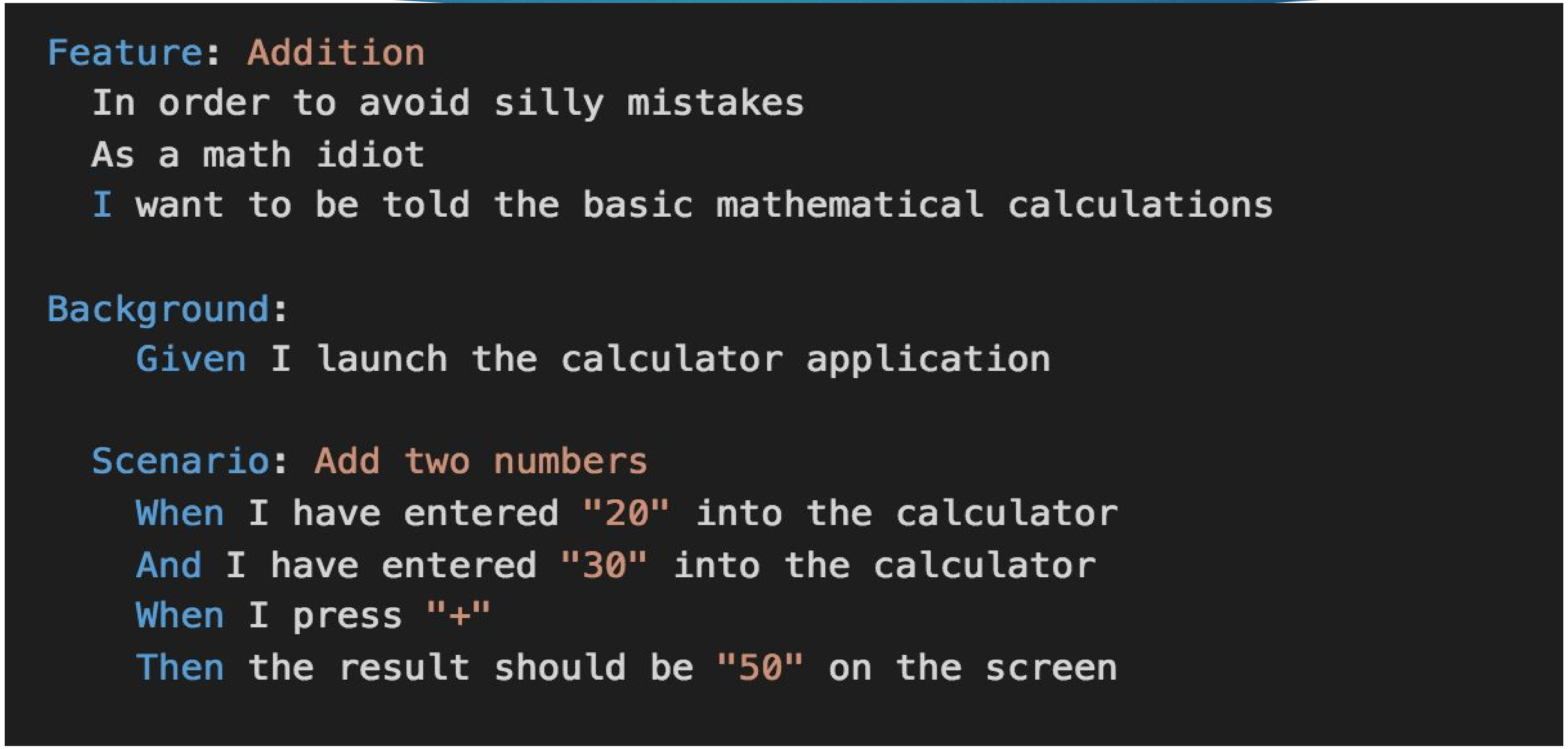
Keyword Driven Approach : Calculator			
Steps	Keyword/Action	Description	Data
Step_1	Launch	Launches an application	C:\Windows\System32\calc.exe
Step_2	SendKey	Sends a Keyboard Input	20
Step_3	SendKey	Sends a Keyboard Input	"+"
Step_4	SendKey	Sends a Keyboard Input	30
Step_5	VerifyText	Verifies a text in a Element	50

Data Driven Approach



Data Driven Approach : Calculator App				
Test Case No	Input 1	Operator	Input 2	Expected Result
TC_01	2	"+"	4	8
TC_02	-2	"-"	-2	-4
TC_03	3	"*"	5	15
TC_04	20	"/"	4	5

Behavior Driven Approach (BDD) : Keyword Driven

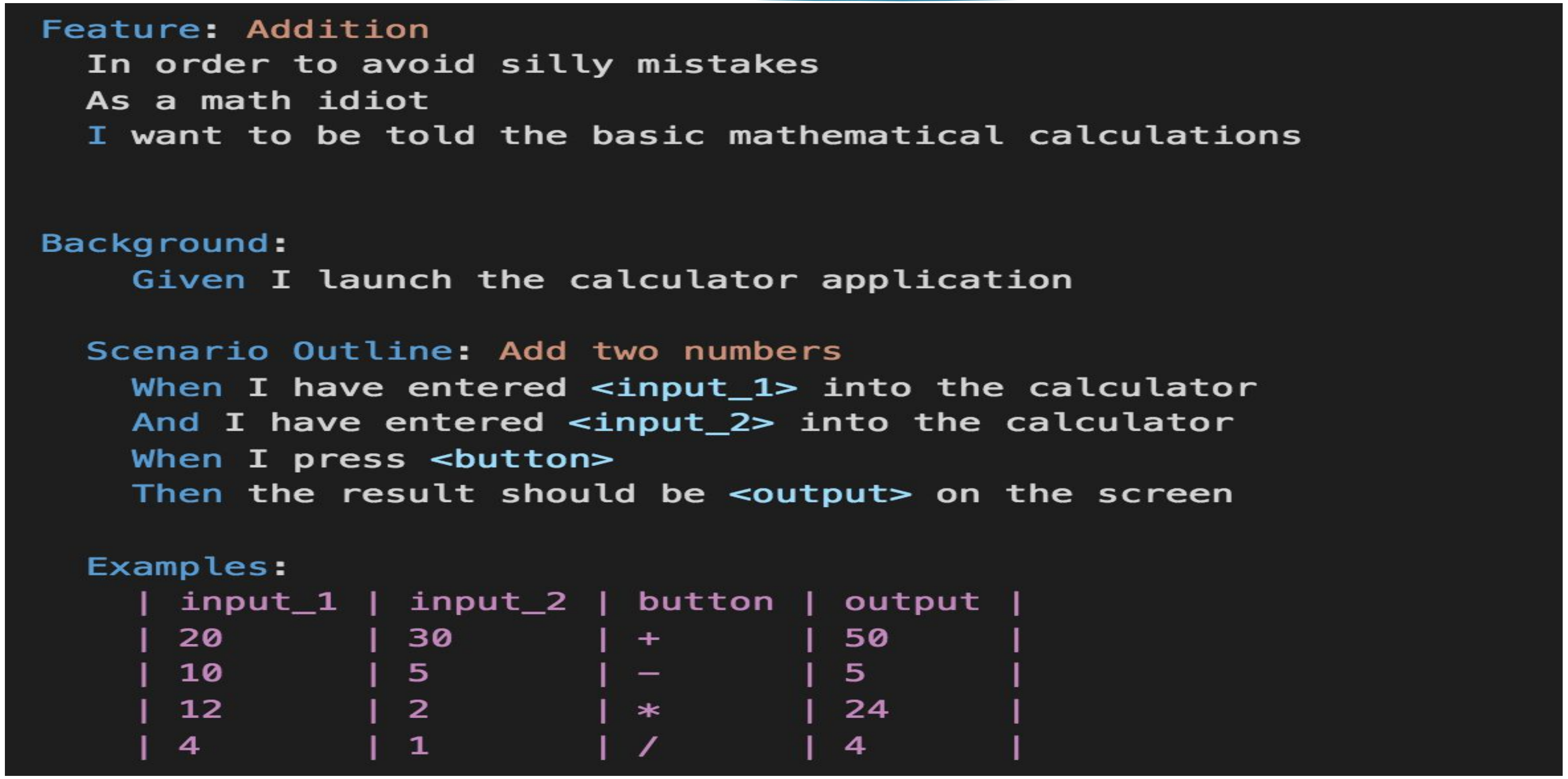


```
Feature: Addition
  In order to avoid silly mistakes
  As a math idiot
  I want to be told the basic mathematical calculations

Background:
    Given I launch the calculator application

  Scenario: Add two numbers
    When I have entered "20" into the calculator
    And I have entered "30" into the calculator
    When I press "+"
    Then the result should be "50" on the screen
```

Behavior Driven Approach (BDD) : Data Driven



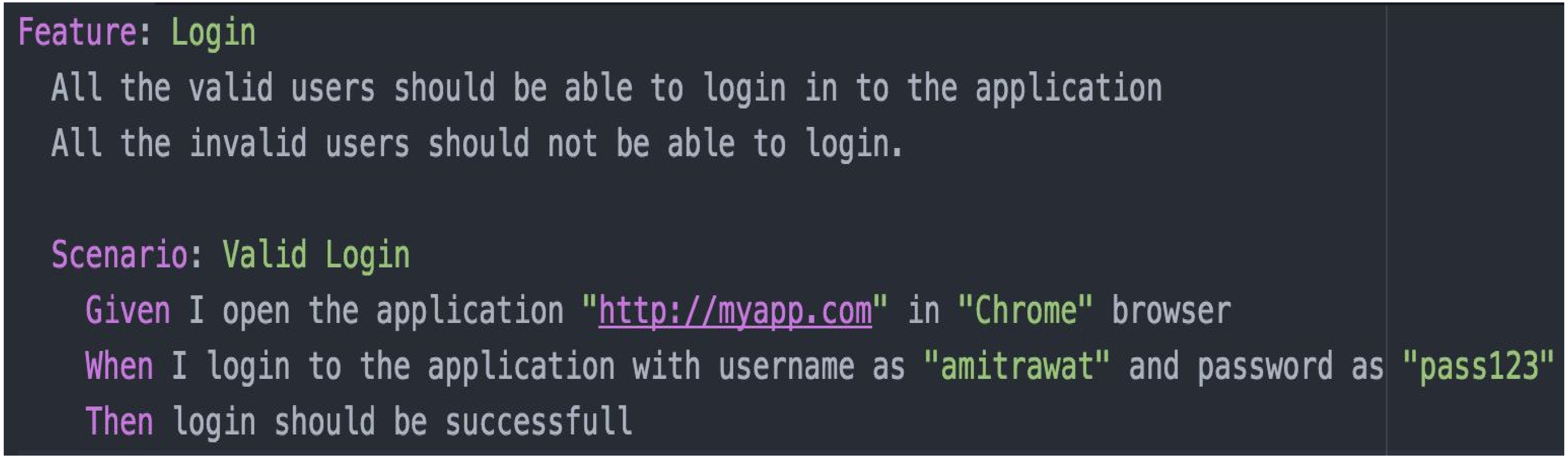
```
Feature: Addition
  In order to avoid silly mistakes
  As a math idiot
  I want to be told the basic mathematical calculations


Background:
    Given I launch the calculator application

  Scenario Outline: Add two numbers
    When I have entered <input_1> into the calculator
    And I have entered <input_2> into the calculator
    When I press <button>
    Then the result should be <output> on the screen

  Examples:
    | input_1 | input_2 | button | output |
    | 20      | 30      | +      | 50     |
    | 10      | 5       | -      | 5      |
    | 12      | 2       | *      | 24     |
    | 4       | 1       | /      | 4      |
```

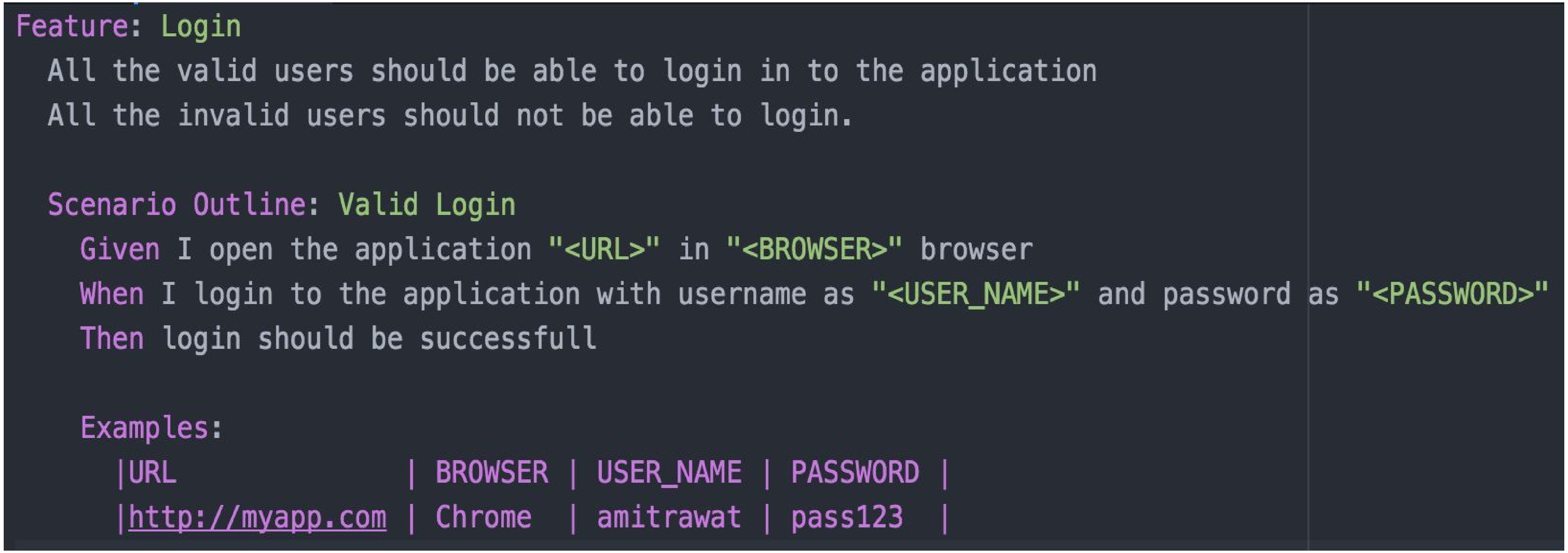
Behavior Driven Approach (BDD) : Functionality Driven



```
Feature: Login
  All the valid users should be able to login in to the application
  All the invalid users should not be able to login.

  Scenario: Valid Login
    Given I open the application "http://myapp.com" in "Chrome" browser
    When I login to the application with username as "amitrawat" and password as "pass123"
    Then login should be successfull
```

Behavior Driven Approach (BDD) : Data Driven



```
Feature: Login
  All the valid users should be able to login in to the application
  All the invalid users should not be able to login.

  Scenario Outline: Valid Login
    Given I open the application "<URL>" in "<BROWSER>" browser
    When I login to the application with username as "<USER_NAME>" and password as "<PASSWORD>"
    Then login should be successfull

    Examples:
      |URL              | BROWSER | USER_NAME | PASSWORD |
      |http://myapp.com | Chrome  | amitrawat | pass123  |
```

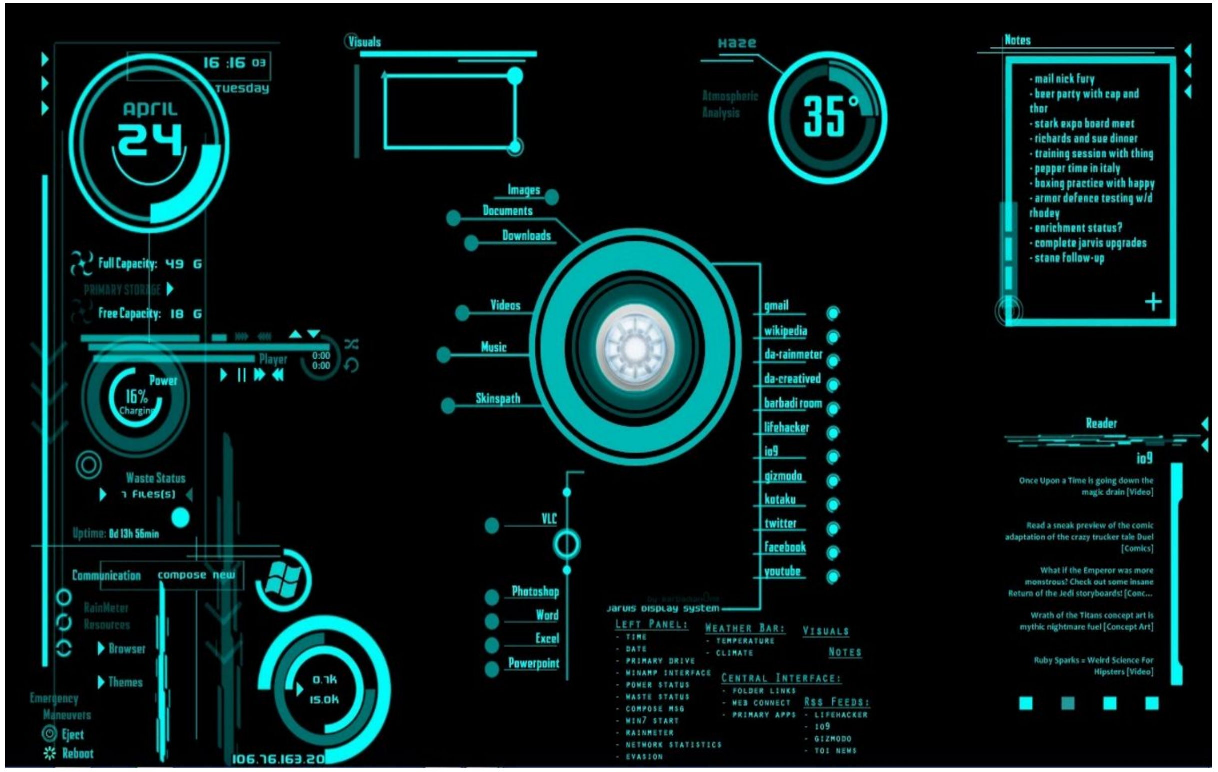
Cucumber Keywords

Background	Doc Strings	Macro/Snippets
Entry	Scenario Outline/Example	Given/When/Then/And/But
Exit	Tags	DataTable

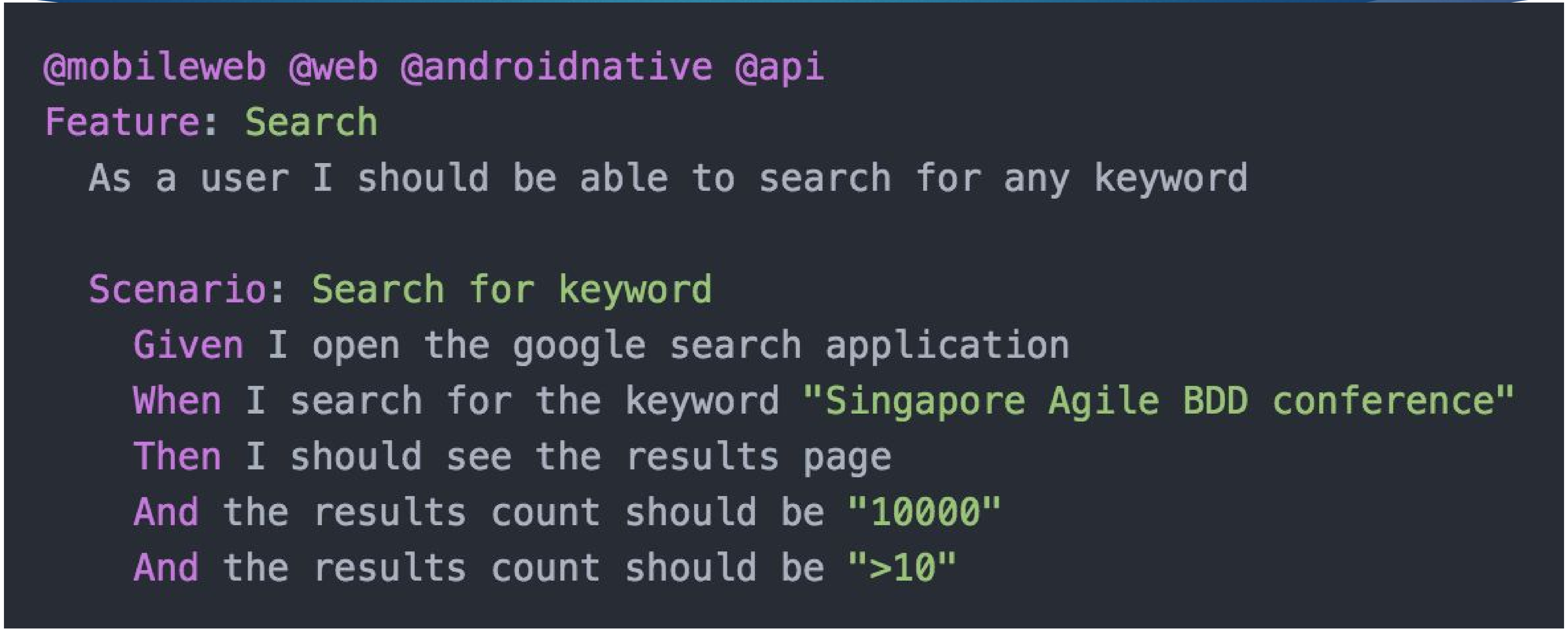
Automating the UI or Automating the FUnctionality ??

UI/UX

Functionality/Workflow



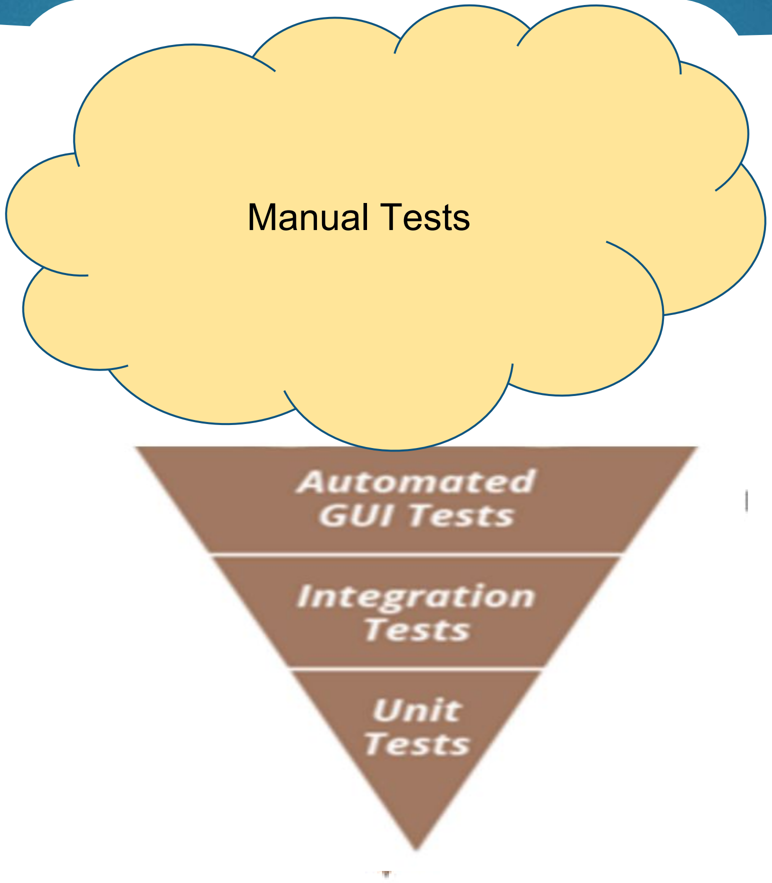
Same Test Across Platforms



```
@mobileweb @web @androidnative @api
Feature: Search
  As a user I should be able to search for any keyword

  Scenario: Search for keyword
    Given I open the google search application
    When I search for the keyword "Singapore Agile BDD conference"
    Then I should see the results page
    And the results count should be "10000"
    And the results count should be ">10"
```

Test Automation Pyramid - Current State



Test Automation Pyramid : Ideal State

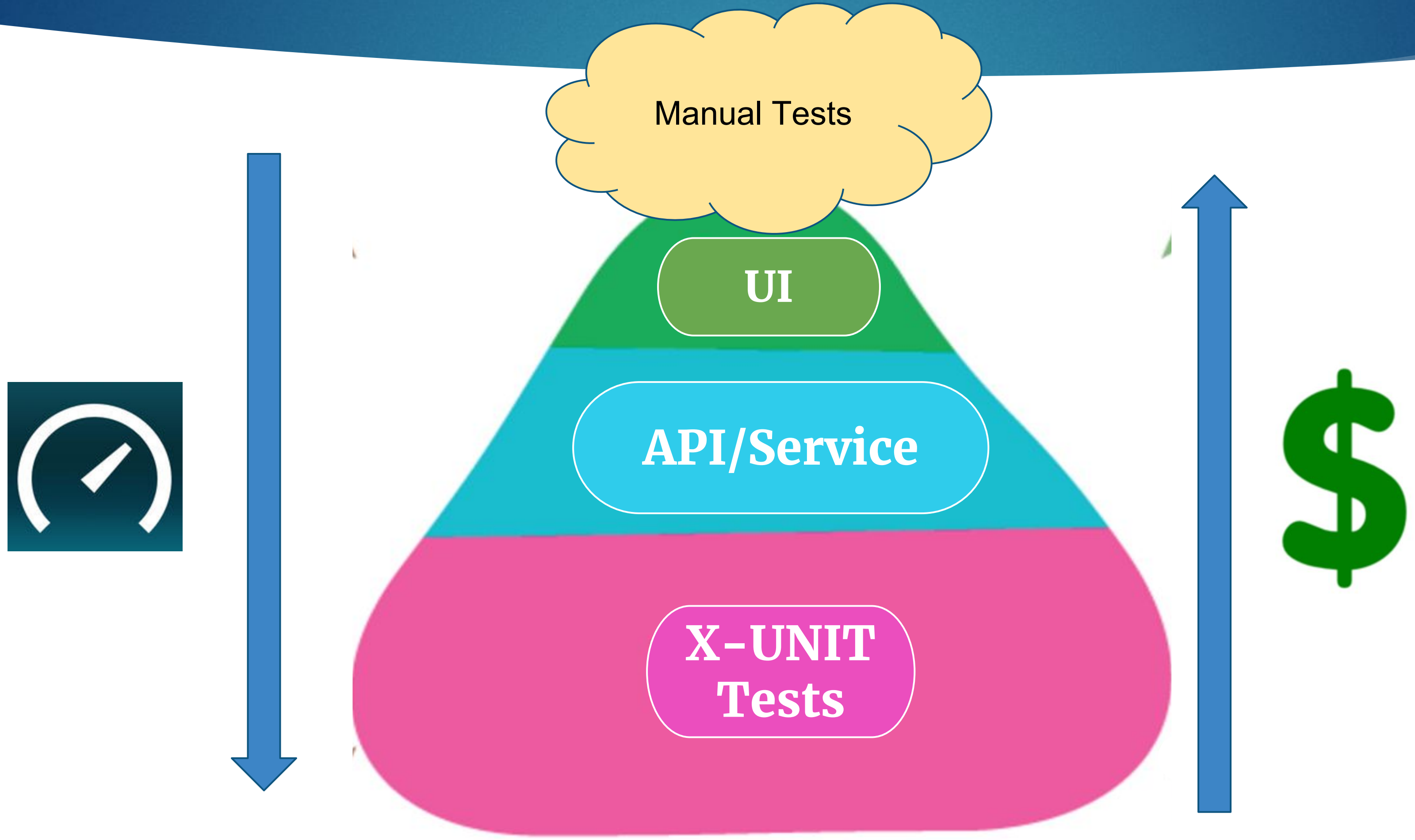
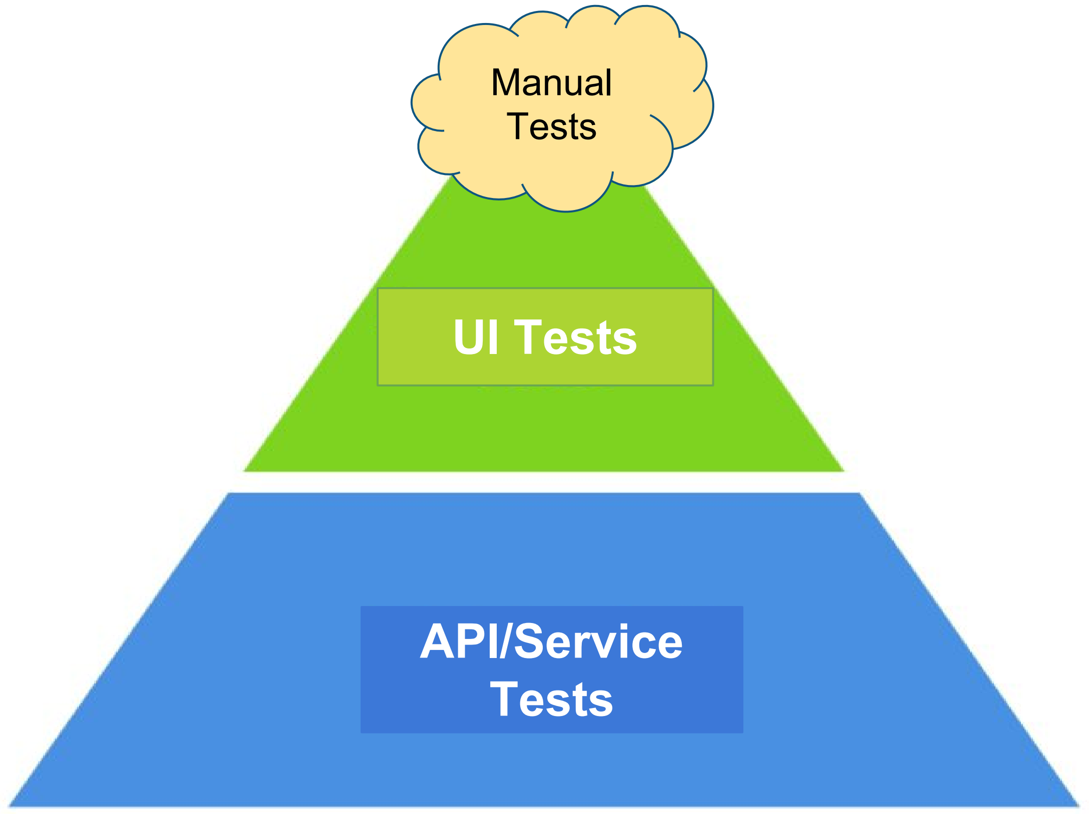
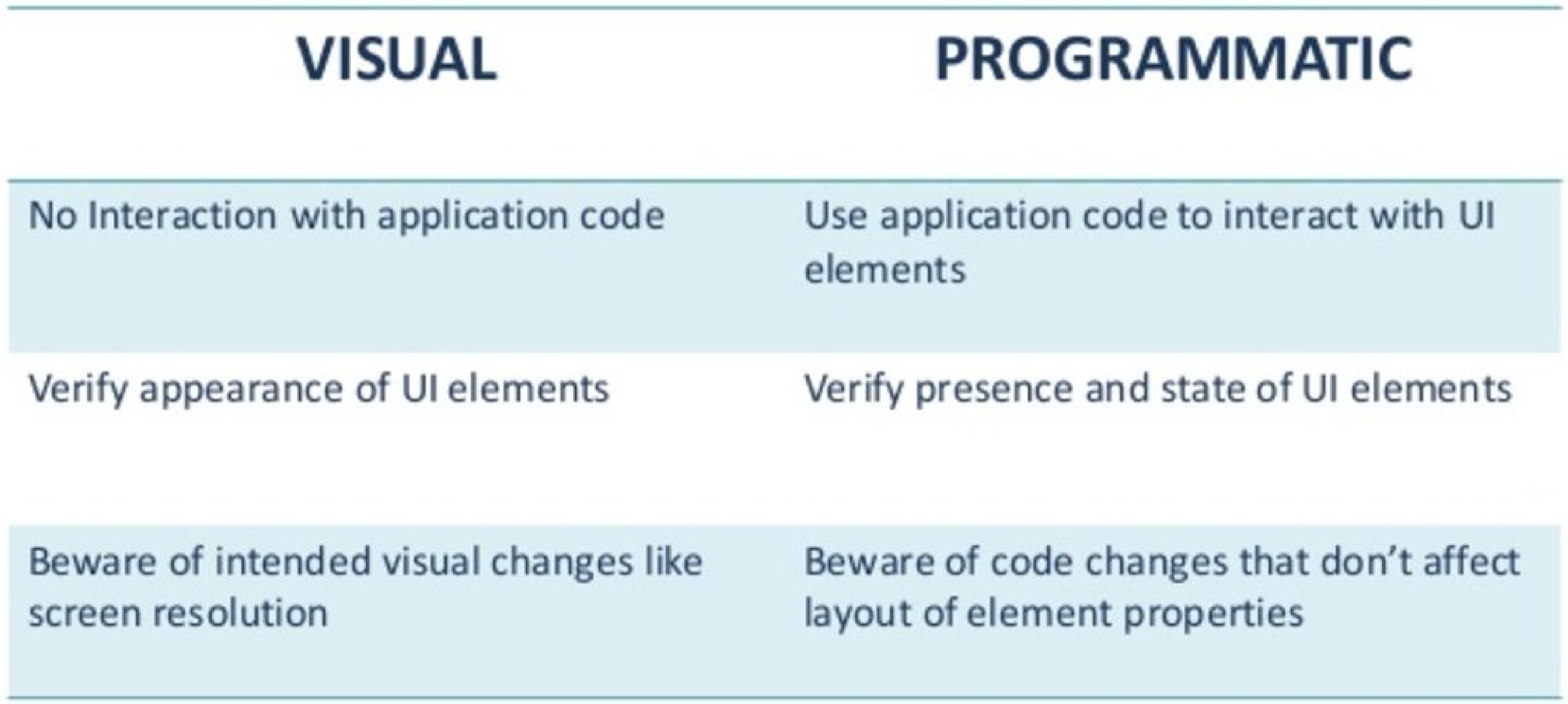


Image Credit:
http://martinfowler.com/bliki/TestPyramid.html

Test Automation Pyramid : Achievable State

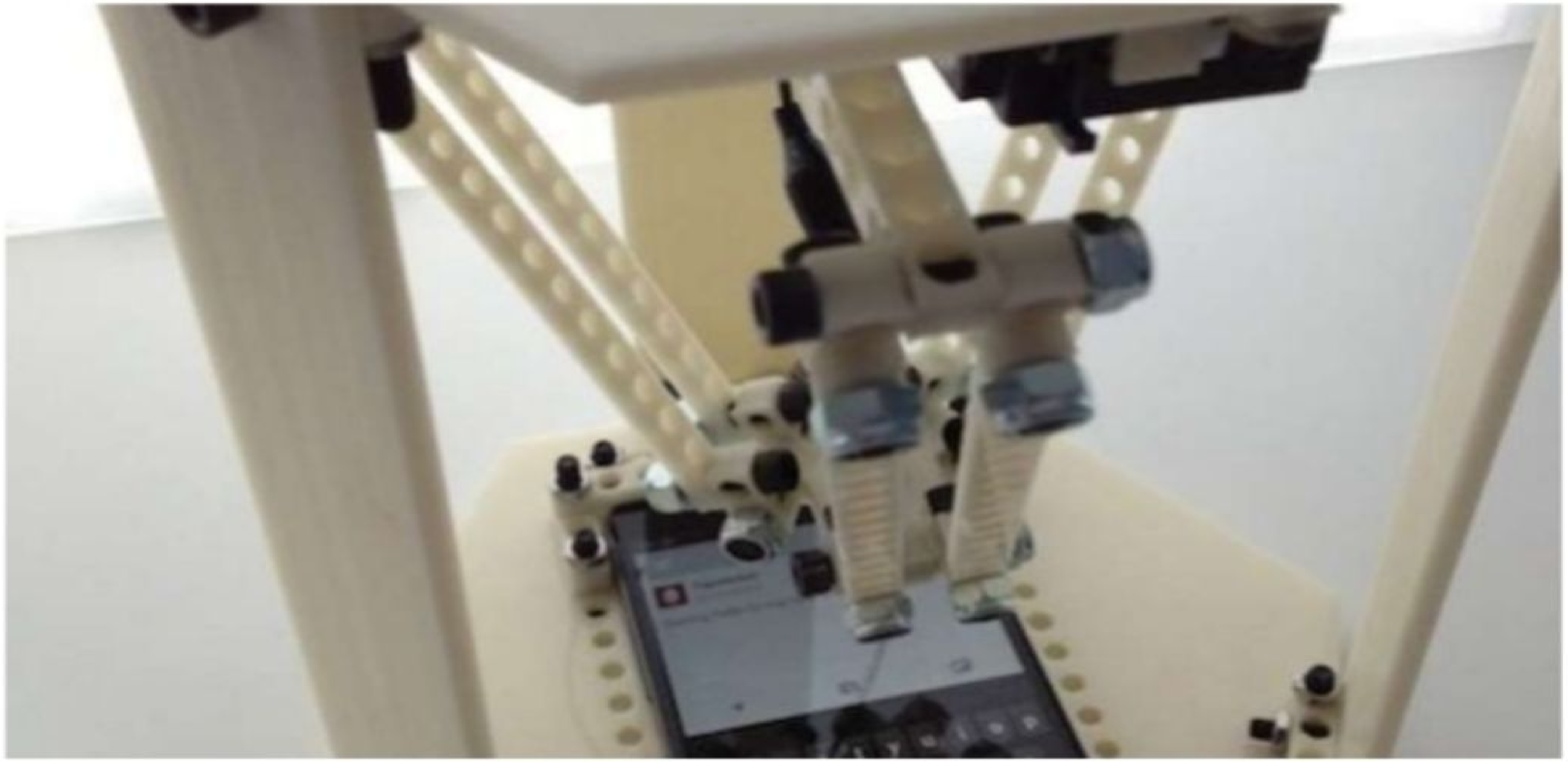


Automation Paradigm: UI Interaction



VISUAL	PROGRAMMATIC
No Interaction with application code	Use application code to interact with UI elements
Verify appearance of UI elements	Verify presence and state of UI elements
Beware of intended visual changes like screen resolution	Beware of code changes that don’t affect layout of element properties

New Paradigm: Automation Robots (Tapster)



Things to consider before designing your Framework

- Progressive Test Automation / Agile Test Automation
- Types of testing to be supported: Functional and Non-Functional
- Parallel execution
- Application Interfaces to be supported: Mobile, Web, Desktop, APIs.
- Operating System
- Reporting : Screenshots, Videos, Data, Logs
- Framework Interface: Web, Excel, Feature Files
- Tool / Language Independence
- Run via CI
- Design Patterns
- Automation Environment

Parallel Execution & Automation Environment

- Headless Execution
- Third party Automation Cloud (SauceLabs, BrowserStack, AWS)
- Virtualization (VMs, Vagrant, Docker)
- Creating Test Environment on the fly (Docker, Kubernetes)

Tool Agnostic Framework

- Abstraction on the tool specific commands
- Leveraging multiple tools beneath the framework layer for different types of testing
    - Web UI Testing - Selenium,AutoIT
    - API Testing - RestAssured, SoapUI, Postman
    - Data Reconciliation - google-diff-match,PDFBox,BeyondCompare
    - Mobile App Testing - Appium, Calabash
    - Network Testing - BMP
    - Responsive Design Testing - Galen Framework
    - Analytics Testing - Fiddler, CharlesProxy
    - Security Testing - Burp
    - Mainframe Testing - Jagacy, IBM PCOM
- Expose Domain Specific Language(DSL)

Locator Strategy

- Multiple Locators
- Statistical Technique
- Artificial Intelligence

Test Automation Design Patterns

- Page Object Model
- Page Factory Pattern
- Loadable component
- Builder
- Singleton