

Code or Low- Code

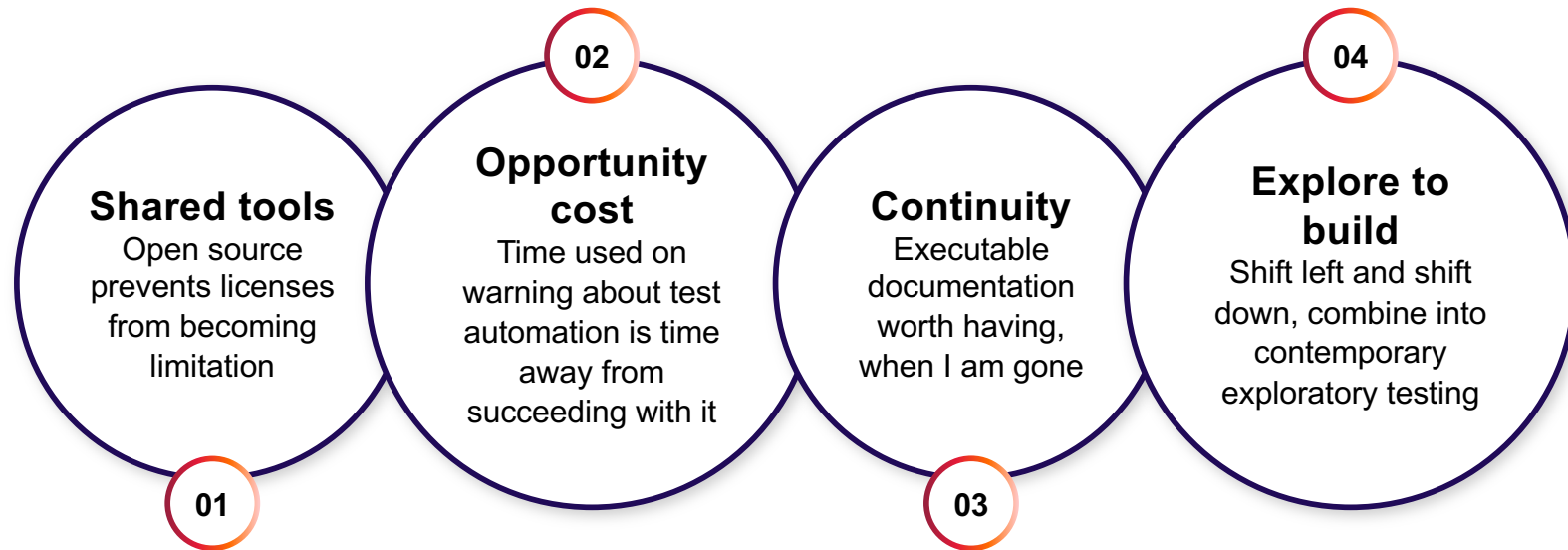
Navigating the test
automation options

Maaret Pyhäjärvi
September 2024

CGI



Hyperautomation – automating everything that can be automated



Collecting **key points** from years with projects of various states of test automation.

```
obfuscated.py x
obfuscated.py > ...
1 x,y=[6 ,11)\
2 ,[ 48,67 ];s\
3 =[ x for x \
4 in vars ( )\
5 ;b= vars( )][s[ x[
6 0]]];k= list( (b.
7 dict_ .keys (
8 ) ));g= getattr(b
9 ,k[x[1]*2]);m=g(
10 b,k[y[1]+1]);l=g\
11 (b ,k
12 [x [1
13 ]* 3] );i=\
14 g( b, k[ x[
15 1] *6]);p =g\
16 (b ,k[y[0 ]-
17 x[0]]);a=g(g(b,k
18 [x[0]] )("".join(
19 m( g(b,k[ x[
20 0] //2+x[ 1
21 ]), (m( g(
22 b, k[y[ 0]
23 -1 ]), g(b,k[x [1]+
24 y[ 1] +1]))(*[ g(b,
25 k[ y[0]-1] (y)] *3
26 ], [0,x[0], 0])) ))
27 )) ,"". join(m(g(b
28 ,k [x[0] //2+x[1]
29 ), (m (g
30 (b ,k [y
31 [0 ]- 1
32 ), g(
33 k[x[1] +y[1] ]+1]
34 )(*[g (b,k [y[0
35 ]- 1])( y) ]*4],[x[0]*- 3,-1
36 ,x [0]* -2 ,x[0]//2])) ));
37 p(i((1/5**.5)*((1+5**.5)/2)**(i(a[1]+1))-((1
38 -5**.5)/2)**(i(a[1]+1)))) if l(a)>1 else p(1)
```

Programming language = system of notation for writing computer programs

Low code / No code = using *intuitive* drag-and-drop tools that reduce need for traditional developers

Agenda for learning

1 **Differences in learning**
Code and Low code

2 **Minimal knowledge for useful contributions**
Learning in layers

3 **Writing and reading**
Reading on failures

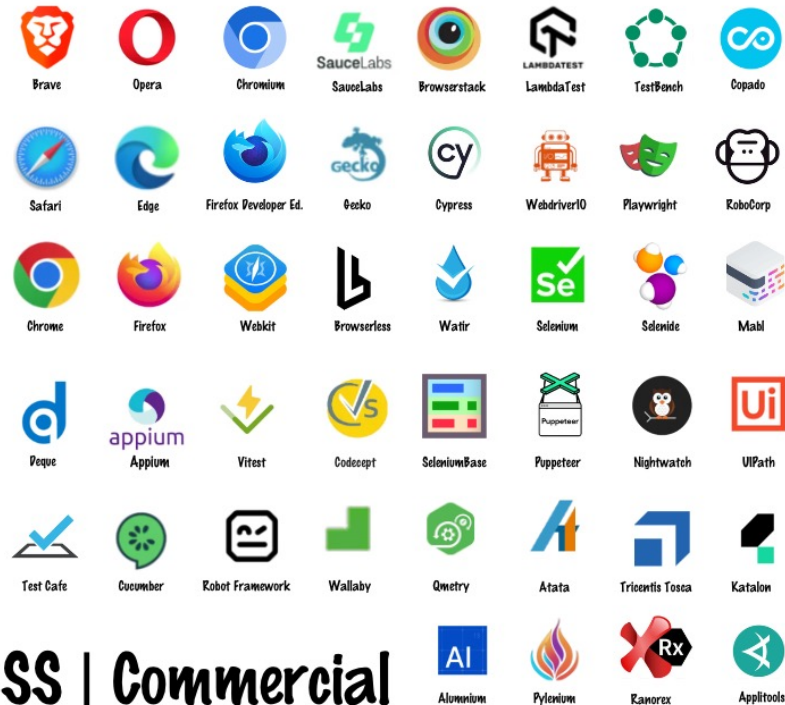


Differences in learning

Code and low code

Polyglot and polytool is a lot of work

Hard to choose



FOSS | Commercial

Different scopes

Can't you
just tell us
which
one to
use?

hello world

02

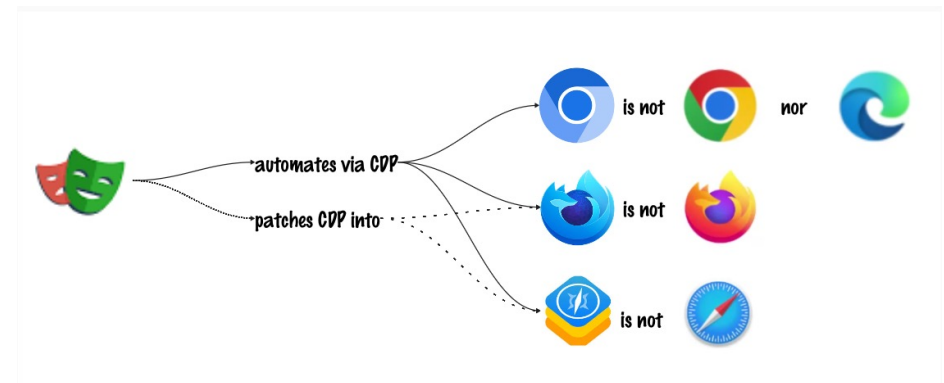
Anti-toolist worldview

Realize that features in tools can be copied. Looking for the **one best tool** makes **little sense**. We need to protect our time to a partner of choice.

Photo by [Clay Banks](#) on [Unsplash](#)

With code

```
≡ requirements.txt
1  #Popular test runner in python
2  pytest
3  #Popular browser approximation library
4  playwright
5  pytest-playwright
6  #Popular real browser library
7  selenium
8  #Popular framework that wraps many libraries
9  robotframework
10 robotframework-browser
```



```
● maaretp@FI-HQ6Q36CQ7C hello_world % python3 -m venv .venv
● maaretp@FI-HQ6Q36CQ7C hello_world % source .venv/bin/activate
○ (.venv) maaretp@FI-HQ6Q36CQ7C hello_world % pip install -r requirements.txt
○ (.venv) maaretp@FI-HQ6Q36CQ7C hello_world % playwright install
○ (.venv) maaretp@FI-HQ6Q36CQ7C hello_world % rfbrowser init
```

test_eprime.py > ...

```
1 from selenium import webdriver
2 from selenium.webdriver.common.by import By
3
4 def test_simplest():
5     driver = webdriver.Chrome()
6     driver.implicitly_wait(10)
7     driver.get("https://maaretp.com/app/")
8     input_field = driver.find_element(By.ID, "inputtext")
9     input_field.send_keys("hello world")
10    driver.find_element(By.ID, "CheckForEPrimeButton").click()
11    assert driver.find_element(By.ID, "wordCount").text == "2"
12    assert driver.find_element(By.ID, "discouragedWordCount").text == "0"
13    assert driver.find_element(By.ID, "possibleViolationCount").text == "0"
14    driver.quit()
```


test_eprime.py > ...

```
1  from playwright.sync_api import Page, expect
2
3  def test_eprime(page: Page):
4      page.goto("https://www.exploratorytestingacademy.com/app/")
5      page.fill("#inputtext", "hello world")
6      page.click("#CheckForEPrimeButton")
7      expect(page.locator("#wordCount")).to_have_text("2")
8      expect(page.locator("#discouragedWordCount")).to_have_text("0")
9      expect(page.locator("#possibleViolationCount")).to_have_text("0")
```

Run Suite | Debug Suite | Load in Interactive Console

1 ∨ ***** Settings *****

2 **Library** **Browser**

3

4 ∨ ***** Test Cases *****

Run | Debug | Run in Interactive Console

5 ∨ **Eprime**

6 **New Browser** chromium headless=\${FALSE}

7 **New page** <https://www.exploratorytestingacademy.com/app/>

8 **Fill Text** css=#inputtext hello world

9 **Click** css=#CheckForEPrimeButton

10 **\${wordCount} = Get Text** css=#wordCount

11 **\${discouragedWordCount} = Get Text** css=#discouragedWordCount

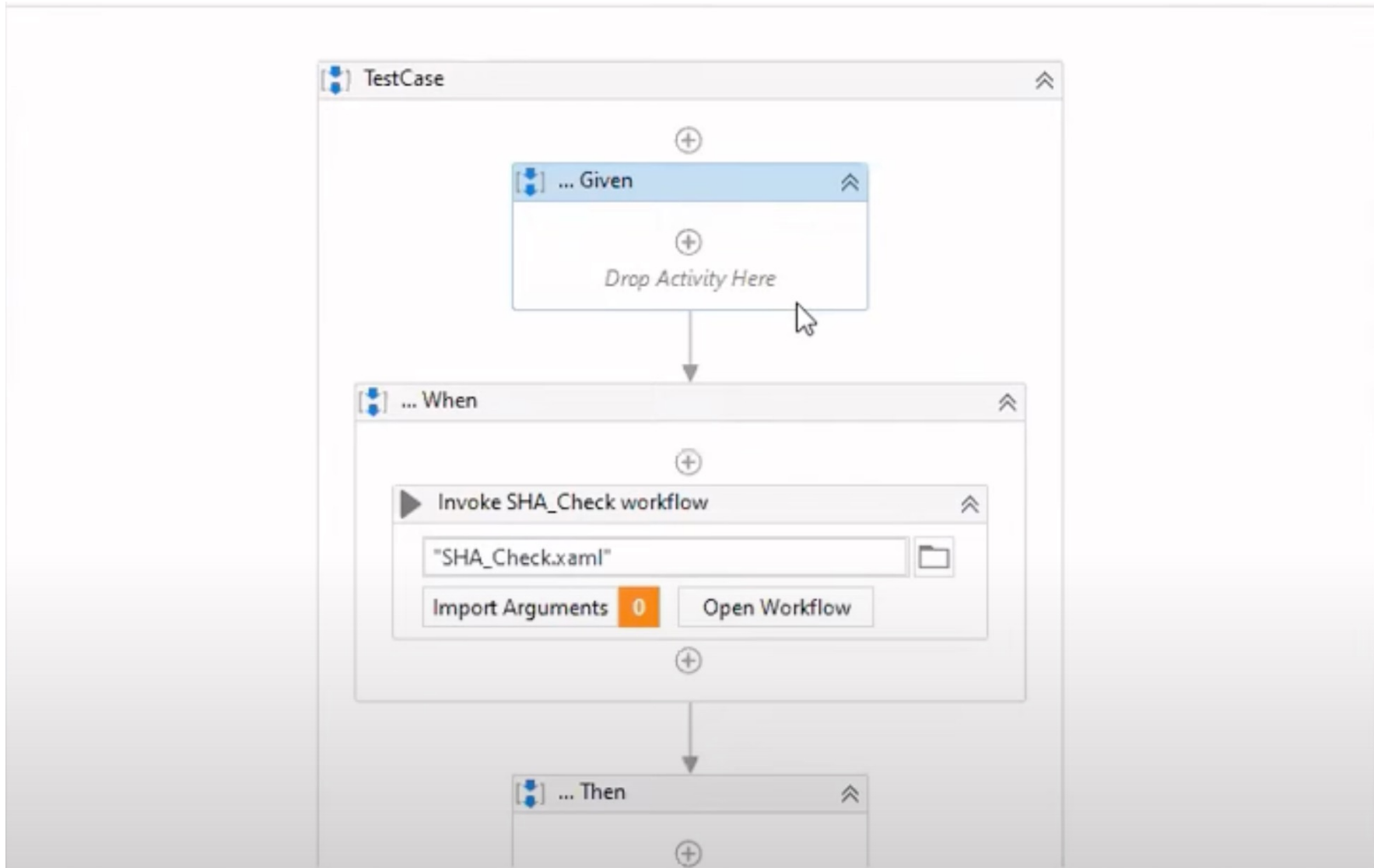
12 **\${possibleViolationCount} = Get Text** css=#possibleViolationCount

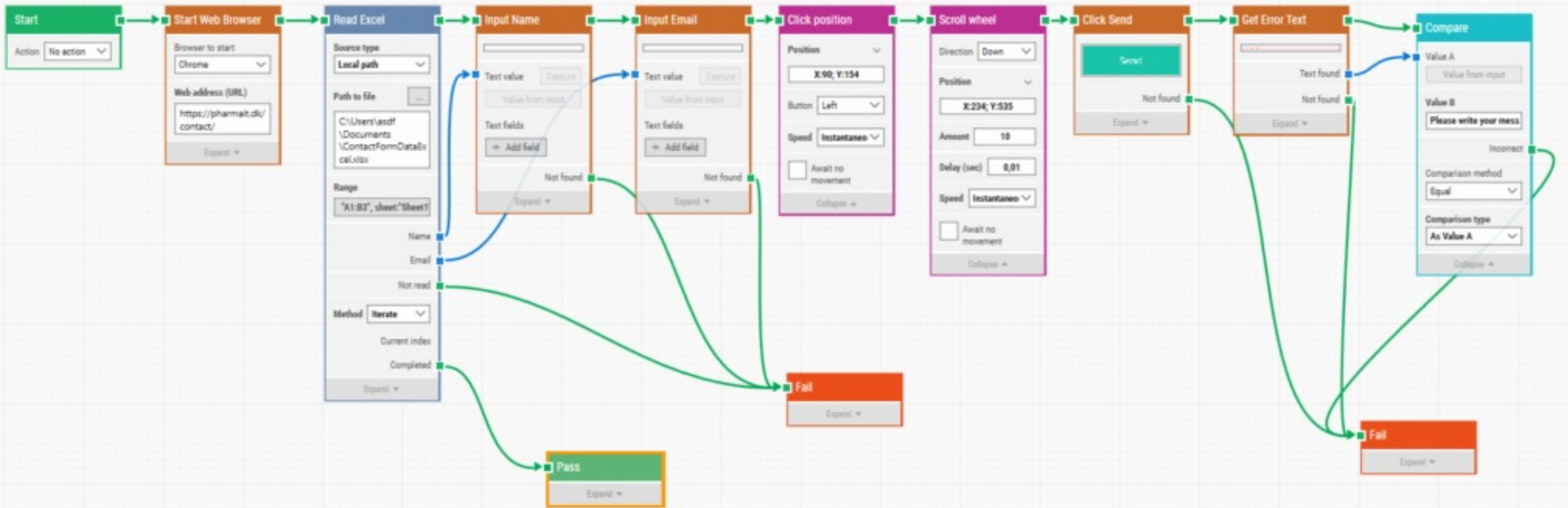
13 **Should Be Equal As Numbers** \${wordCount} 2

14 **Should Be Equal As Numbers** \${discouragedWordCount} 0

15 **Should Be Equal As Numbers** \${possibleViolationCount} 0

16 **Close Browser**





```
(.venv) maaretp@FI-HQ6Q36CQ7C Hercules-demo % testzeus-hercules --project-base MINI --llm-model gpt-4o --llm-model-api-key sk-proj-6hxuda
```

```
MINI > input > ≡ test.feature
```

```
1 Feature: Eprime text analysis
```

```
2 As a user,
```

```
3 I want to verify my text for violations of eprime,
```

```
4 So I learn to write proper English
```

```
5
```

```
6 Scenario: Eprime analysis
```

```
7 Given the eprime page (https://maaretp.com/app/) is displayed
```

```
8 When user analyses sentence hello world
```

```
9 Then user learns sentence has 0 be-verbs, 0 possible be-verbs and total 2 words
```

Test Report : test.feature_result.xml

Eprime text analysis
• [Eprime analysis](#)

Minimal knowledge for useful contribution

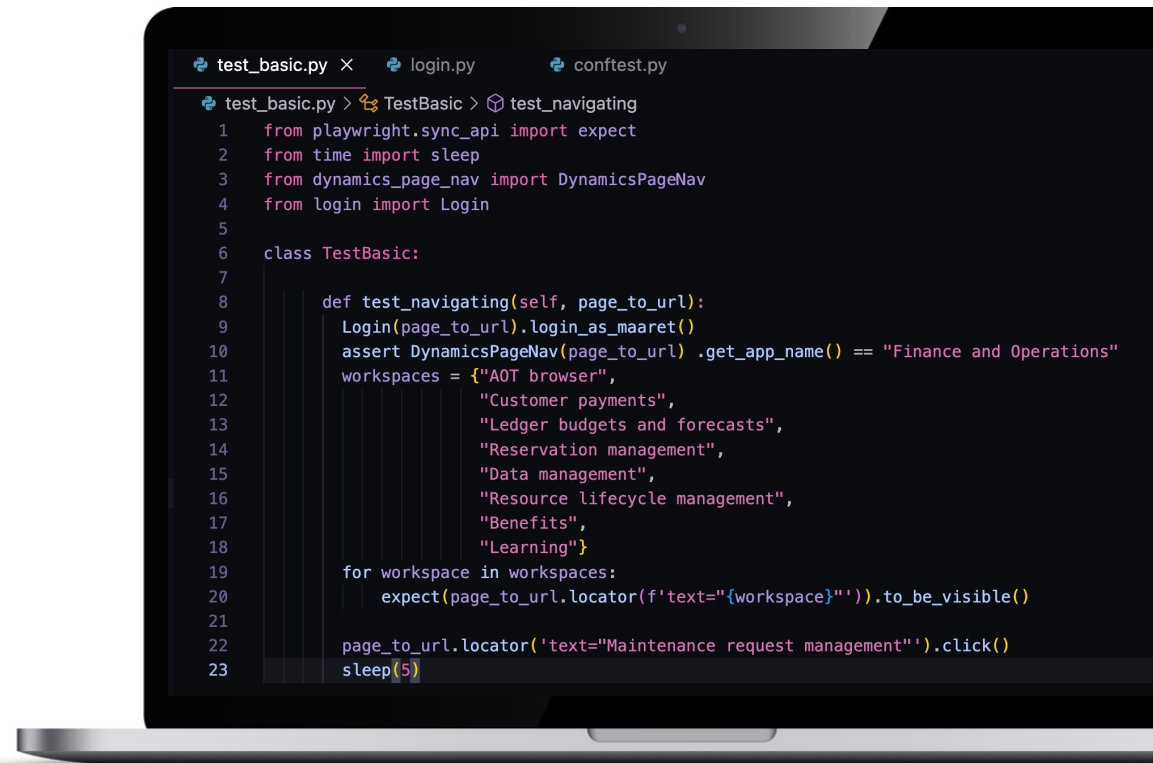
Learning in layers

For test automation, you choose a language

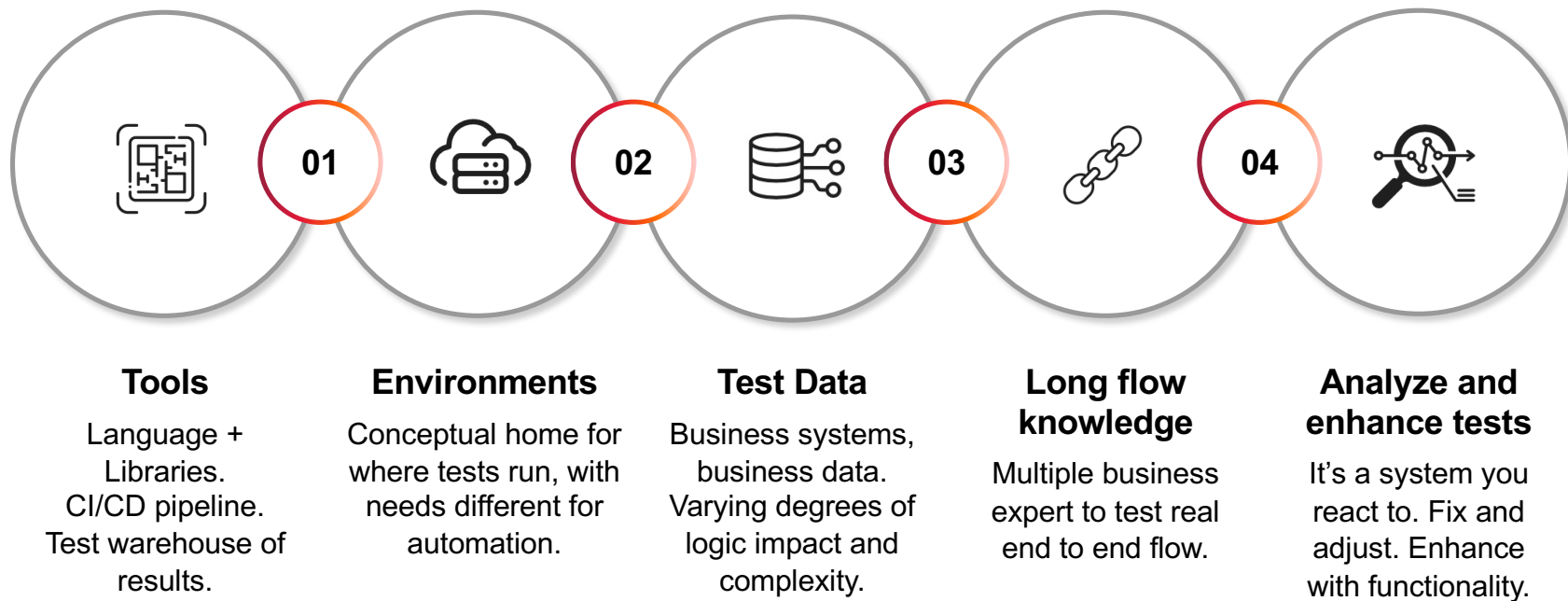
Tradeoffs

- Visual / Robot Framework / Gherkin / TypeScript / Python / Java / C# / ...
- Enable ownership over time when people change positions
- Granularity: feedback to where things got broken
- Reuse and integrations

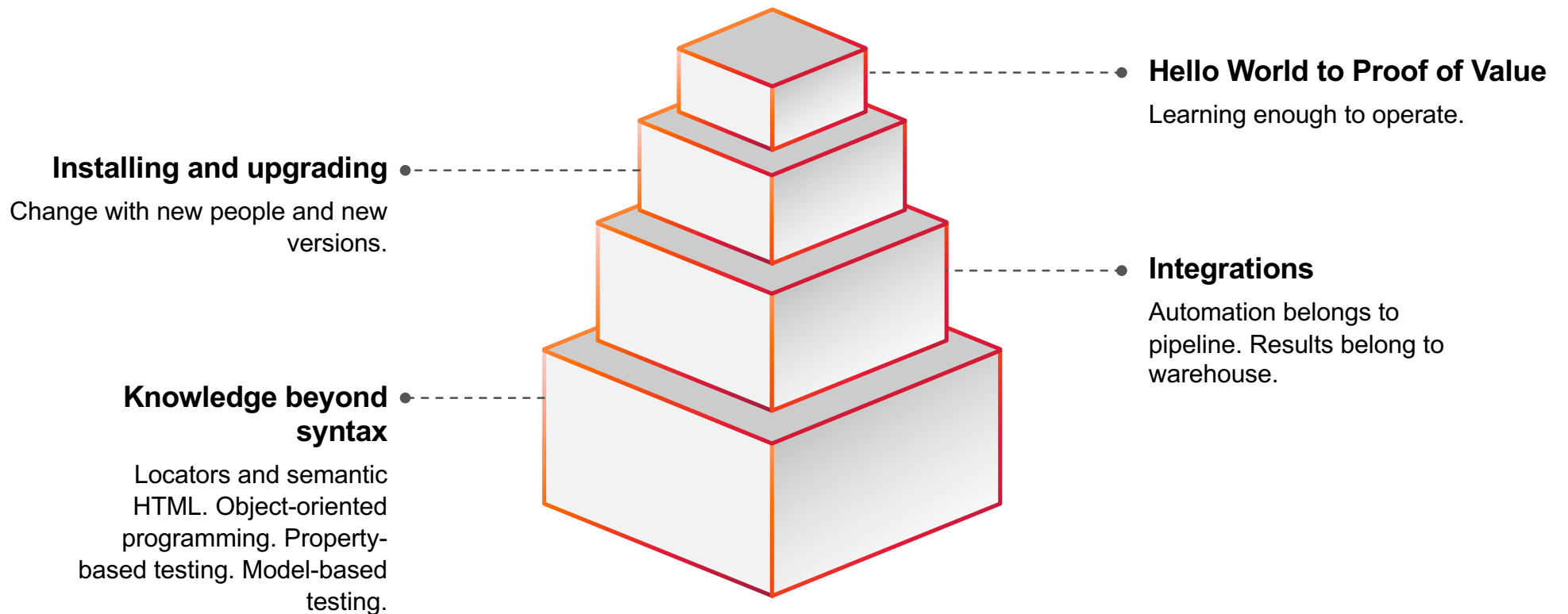
Usual choice: automation in the language of the implementation.



It's not just the language / tools



Learning in Layers




Writing and reading

Reading on failures

Failing automation
is invitation to
explore

WHAT
DO YOU
MEAN
?

Photo by [Jon Tyson](#) on [Unsplash](#)

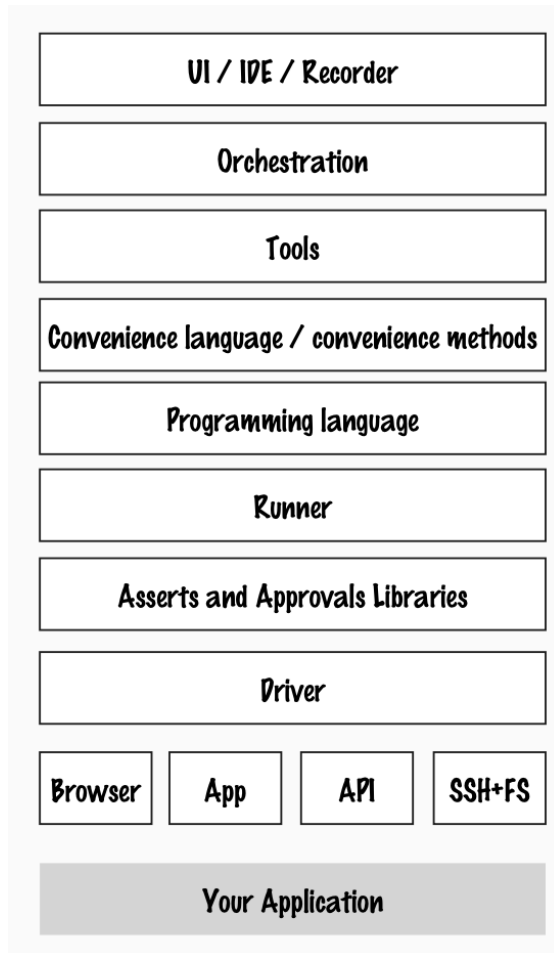


After AI-enabled automation of 80% of the work, the 20% that remains is the portion that requires skills

Photo by [Austin Distel](#) on [Unsplash](#)

Looking inside the box

...or building up the box so that we know what the box contains?



Key insights



Whole team ownership

Sharing tests with developers is essential.



“The 10 years of success turned into failure within a month after the creator left”



Incremental strategy

One reliable test already running is better than plans.



“The plan for what to automate was obsolete after automating the first scenario”



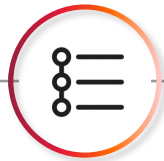
Design for automation

Decomposing feedback for automation.



“We have 5000 tests and half of them are failing. Can you recommend AI that fixes them?”

Test automation gives us more than regression



Specification

Examples shared **before implementation** act as specification.



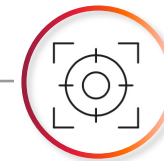
Feedback

We know **when it works** as we're implementing the progress.



Regression

We **safeguard our past intent** over lifetime of system ownership.



Granularity

We **know when and what** changed without extensive analysis.



Documenting

Capture examples whenever we discuss them.



Extending reach

Time. Environments. Data.



Alerting to attend

Changes, both intended and unintended.



Guiding to detail

Drives attention to details humans do repair for.

Insights you can act on

Founded in 1976, CGI is among the largest IT and business consulting services firms in the world.

We are insights-driven and outcomes-based to help accelerate returns on your investments. Across hundreds of locations worldwide, we provide comprehensive, scalable and sustainable IT and business consulting services that are informed globally and delivered locally.

cgi.com



CGI