tesena
SMART TESTING

Welcome !

Don’t miss the train!

Test automation is finally growing up.

Trends in Test Automation

Test automation is finally growing up!

About me

Marcel Veselka

tesena
SMART TESTING

IT in change . . .

IT in change . . .

Investments still grow

Spending on SW grows over HW

Competition via quality & speed

Impact on testing?

General Trends in Testing

4 moves

1. Expectations

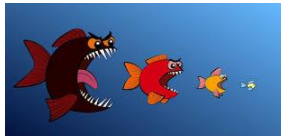
- Responsibilities
- Skills & competencies

2. Organization changes

- Test transformation surges
- Testing centers of excellence
- Adoption of Agile / DevOps

3. Market moves

- Merges & acquisitions (e2e offerings)
- Open source vs vendor tools



4. Disruption & disruptors

- AI / ML
- Blockchain testing
- RPA

The key drivers . . . still same?

Quicker

Cheaper

Better quality

. . . . **Test Automation** might be helpful, see next slides

Current status of Test Automation

Elite DevOps teams perform only

10% of Testing Manually

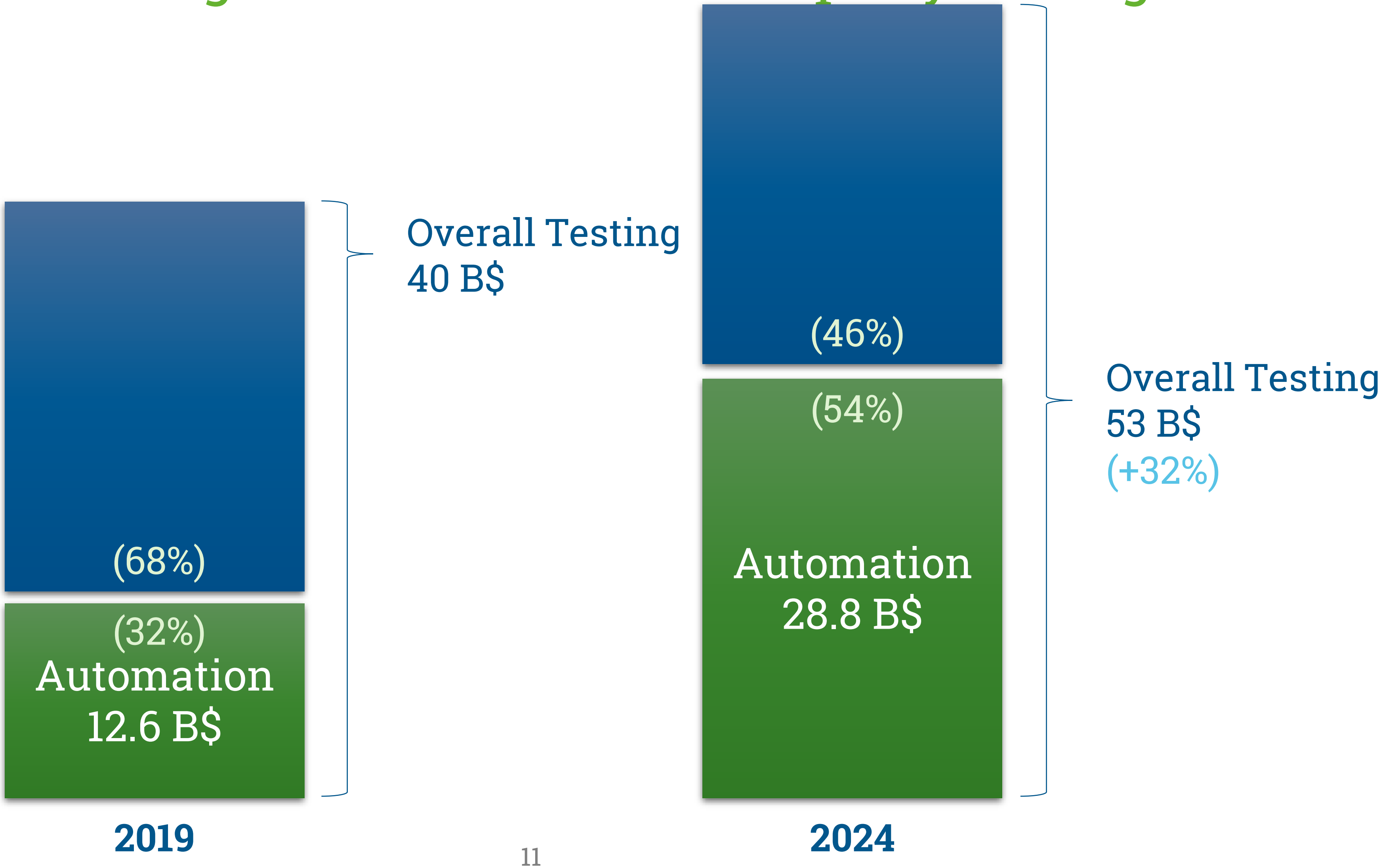
. . . not in future

NOW ! ! !

Source: Accelerate: State of DevOps 2018

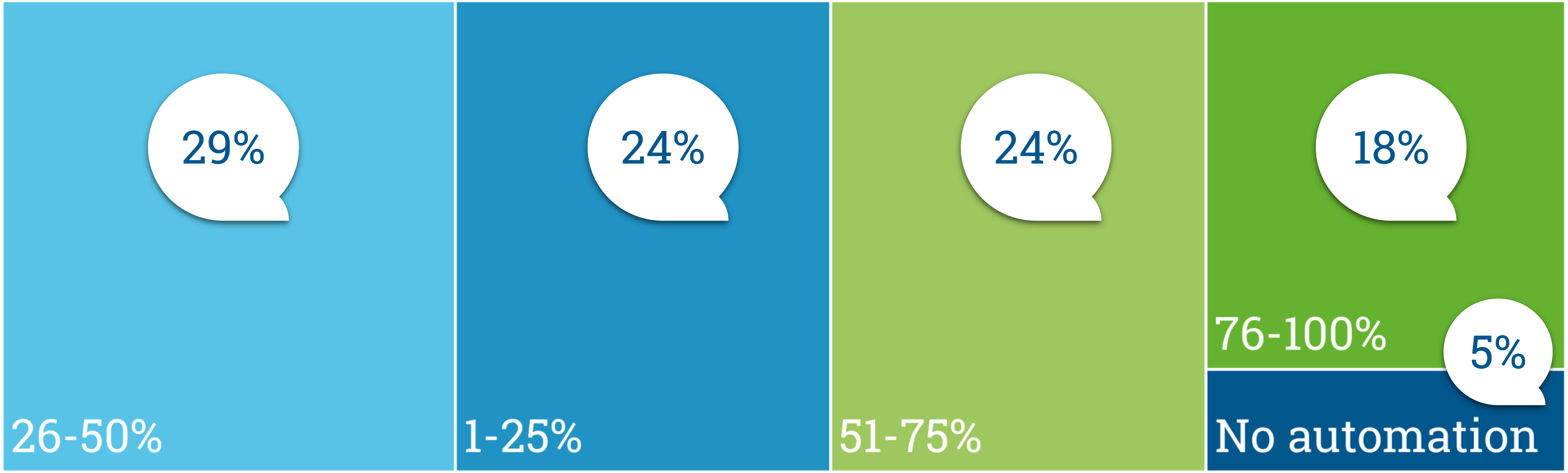
Automation overtakes testing?

Automation Testing will eat 50%+ of company's budgets.



Current Level of Test Automation

Above 50%: in 34% organizations
No automation: (only) in 5% organizations



Source: Survey KPMG & NASSCOM

. . . so what are the

Trends in Test Automation?

Trends in Test Automation

1. Old ideas are coming back

- More tools (support API & microservices, simulation)
- Record & play is back!
- Selenium era is over?

2. Fixing the pain of scale

- Intelligent test execution
- Intelligent maintenance (self-healing)

3. AI & ML

- Top AI/ML opportunities in Automation
- Autonomous Test Automation

4. Other opportunities

- Continuous testing (shift left & right)
- Chaos engineering
- Robotic Process Automation

1. Old ideas are coming back

- More API & microservices; more simulation
- Record & play is back!
- Selenium era is over?

Containers, Simulators, Cloud

- Microservices & APIs

GraphQL

- Containers & virtualization

docker kubernetes

- Cloud services

aws Microsoft Azure Google Cloud

- Web & mobile configuration simulators

SAUCELABS

BrowserStack

SMARTBEAR CrossBrowserTesting

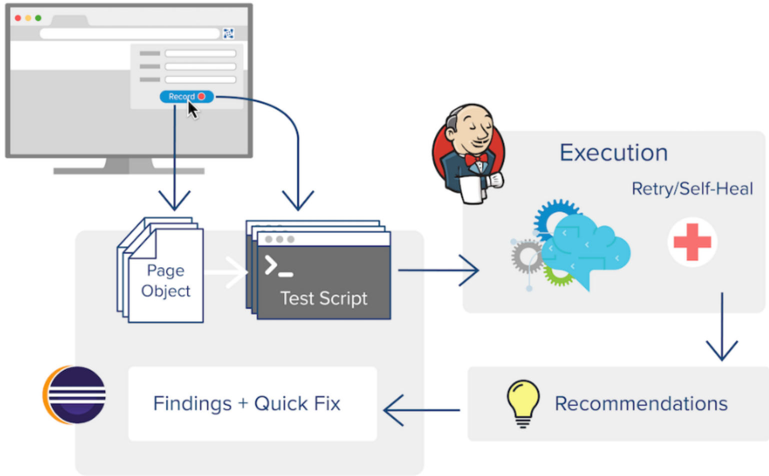
Record & play come back

Example 1: New Selenium IDE

- Better & more complex implementations
- Better stability
- Better Execution
- Extensible with Plugins

Example 2: Parasoft Selenic

- Self-healing implementation
- Generate (both into existing or newly build projects)
  - Page objects and
  - Better code structure



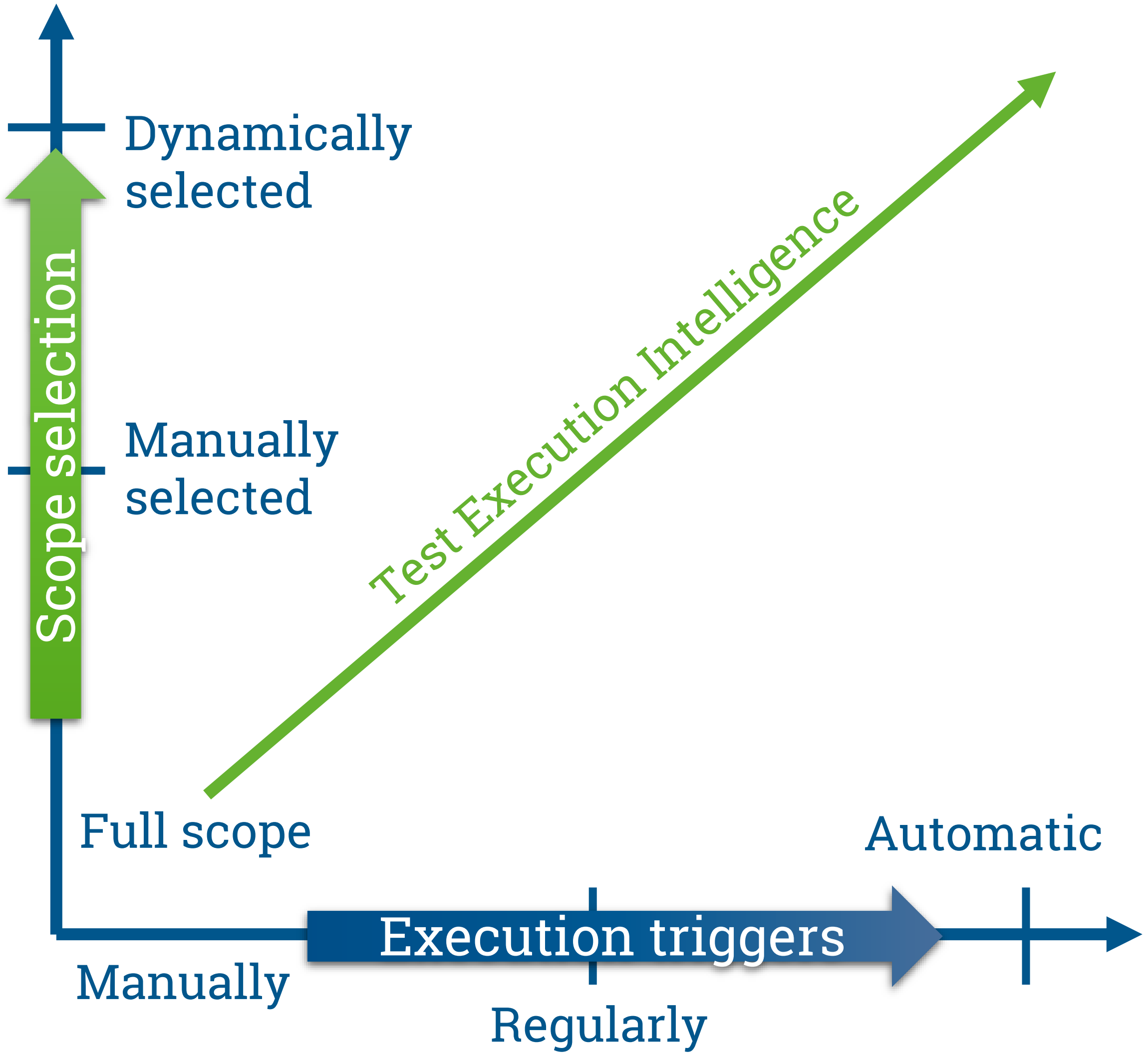
Selenium era is over?

Cypress.io https://cypress.io	cypress.io
Playwrite https://playwright.dev/	Playwright
Nightwatch.js http://nightwatchjs.org	Nightwatch.js End-to-end testing, the easy way.
Puppeteer https://github.com/puppeteer/	Puppeteer
TestCafe https://testcafe-studio.devexpress.com	TestCafé®

2. Pain of scale

- Intelligent test execution
- Intelligent maintenance (self-healing)

Intelligent Test Execution



Self-healing tests

= ability to automatically update scripts / behavior during test execution

Features of self-healing tests

1. Fix locators

Dynamic

Intelligent (powered by AI/ML)

2. Assessments

Improved maintenance

- Advanced reporting
- Dynamic assertions

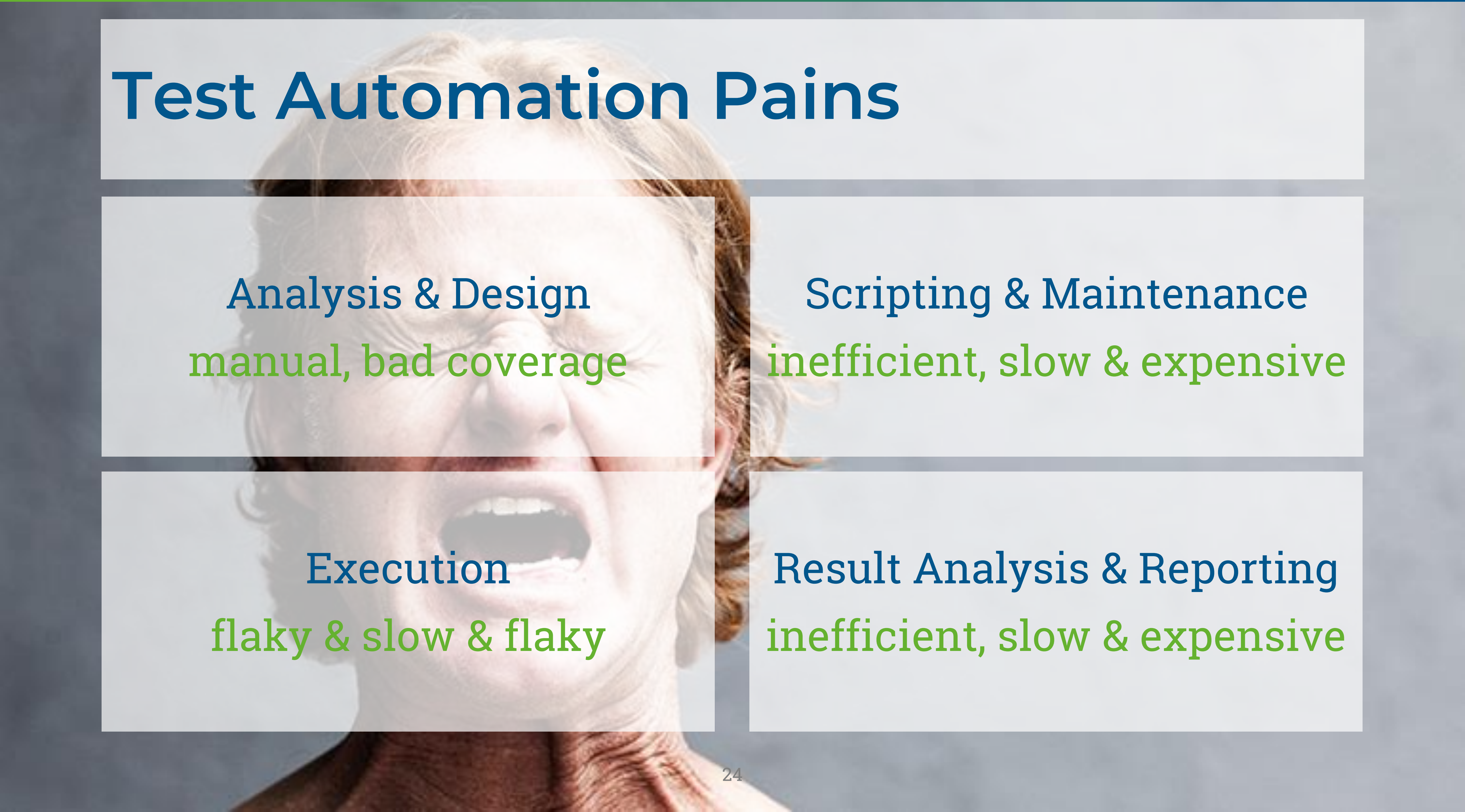
3. Code update

Automatic script updates

3. AI & ML

- Top AI/ML opportunities in Automation
- Autonomous Test Automation

Test Automation Pains



Analysis & Design

manual, bad coverage

Scripting & Maintenance

inefficient, slow & expensive

Execution

flaky & slow & flaky

Result Analysis & Reporting

inefficient, slow & expensive

My top AI / ML
Opportunities in Automation

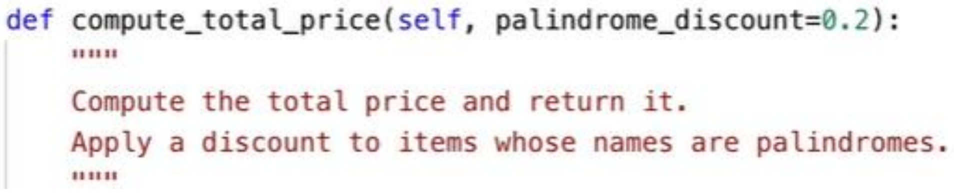
1. Scripting & Maintenance

Generate tests

user story > test script

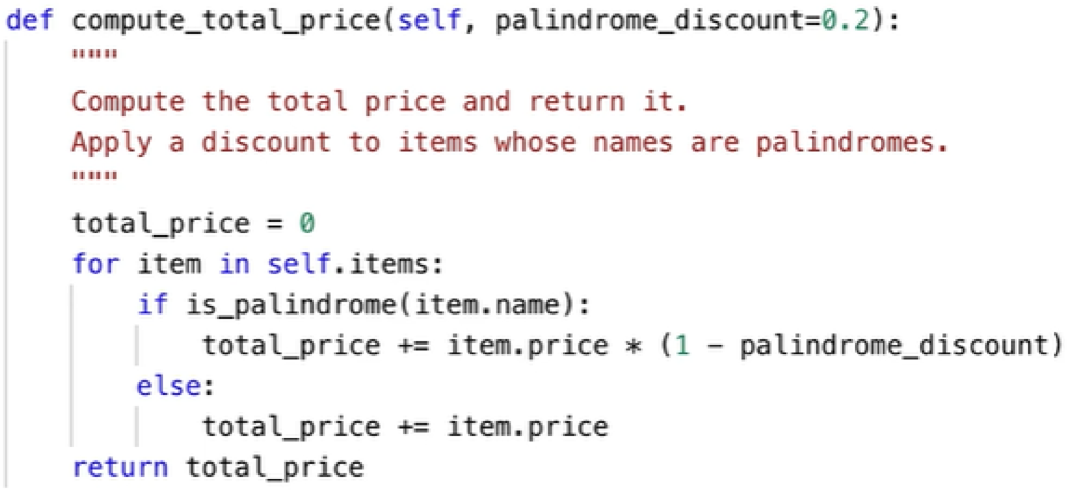
Example with python code

Input



```python
def compute_total_price(self, palindrome_discount=0.2):
    """
    Compute the total price and return it.
    Apply a discount to items whose names are palindromes.
    """
```

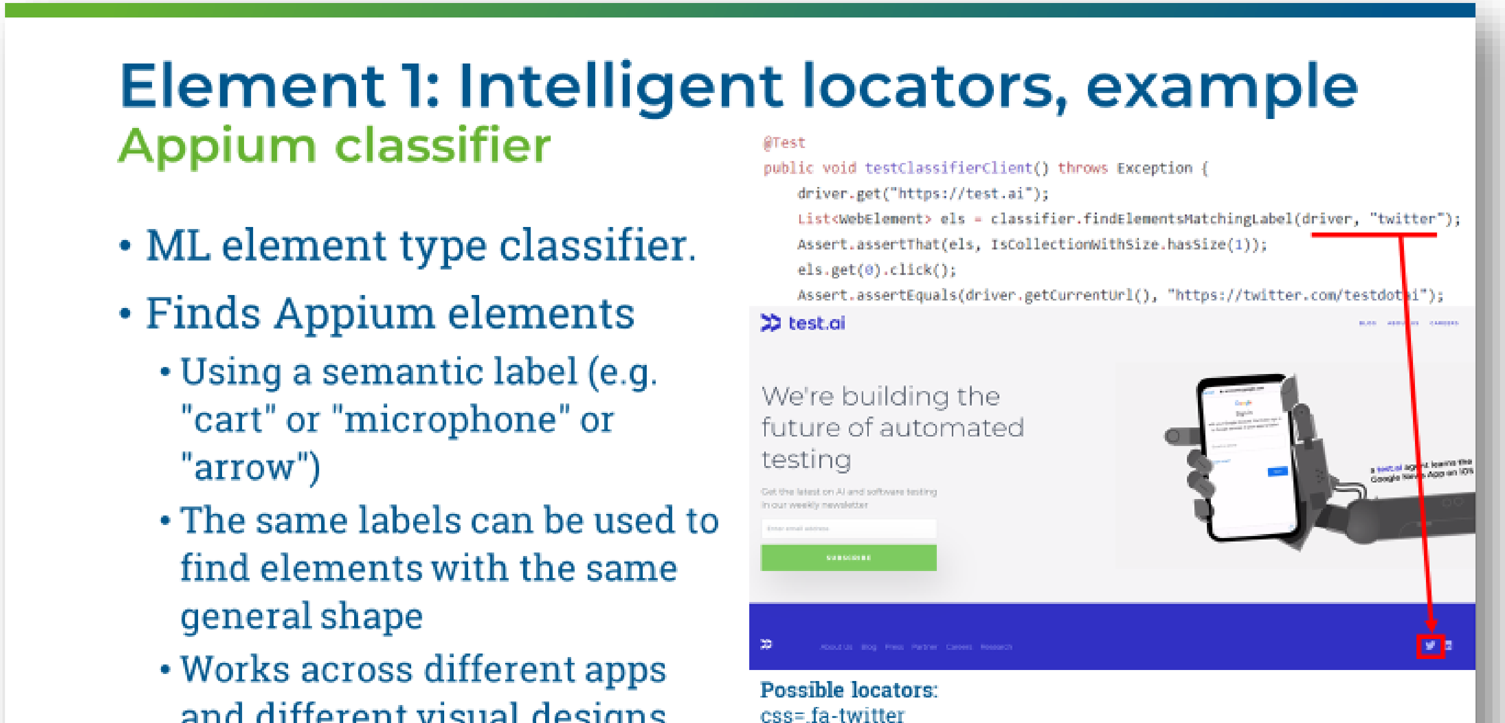
Output



```python
def compute_total_price(self, palindrome_discount=0.2):
    """
    Compute the total price and return it.
    Apply a discount to items whose names are palindromes.
    """
    total_price = 0
    for item in self.items:
        if is_palindrome(item.name):
            total_price += item.price * (1 - palindrome_discount)
        else:
            total_price += item.price
    return total_price
```

2. Execution & Maintenance

- Easier navigation through apps
- Intelligent (UI) interaction, dynamic locators
- Re-execution with self-healing



Element 1: Intelligent locators, example

Appium classifier

- ML element type classifier.
- Finds Appium elements
  - Using a semantic label (e.g. "cart" or "microphone" or "arrow")
  - The same labels can be used to find elements with the same general shape
  - Works across different apps and different visual designs

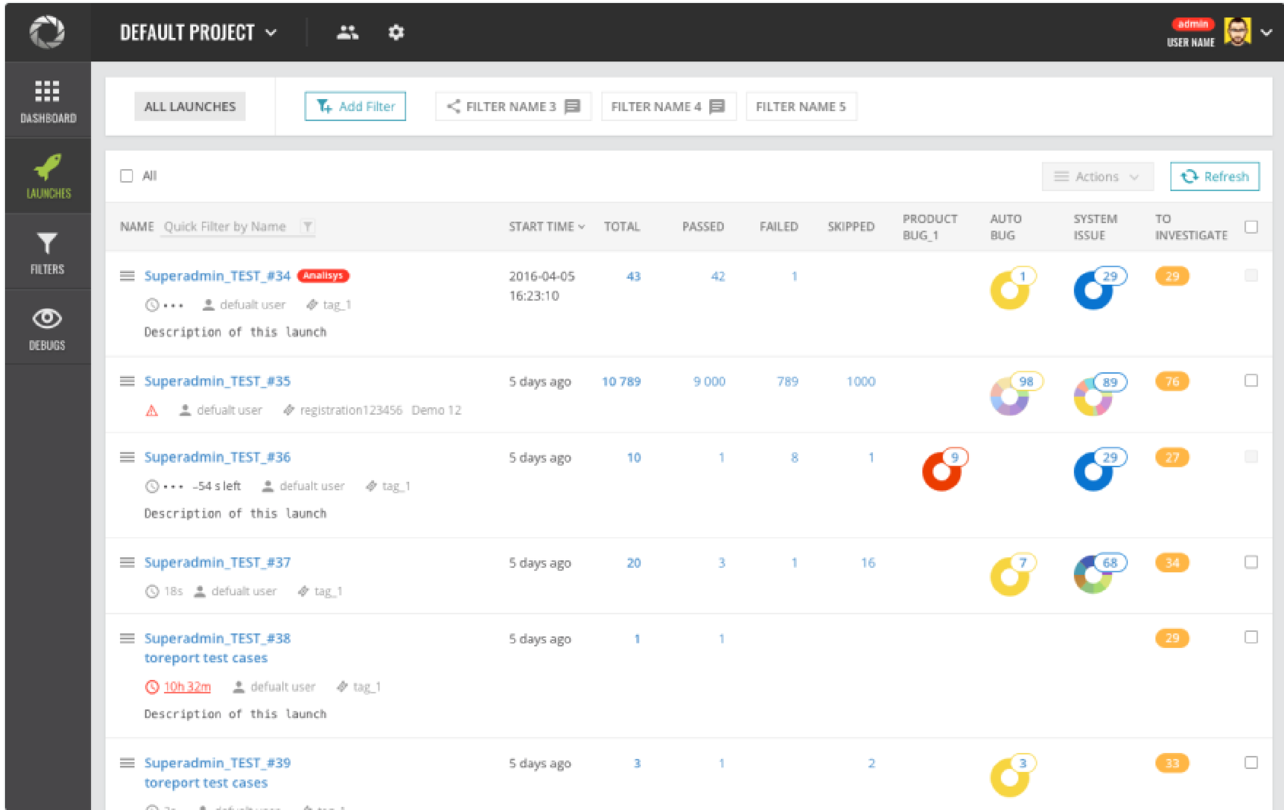
```
@Test
public void testClassifierClient() throws Exception {
    driver.get("https://test.ai");
    List<WebElement> els = classifier.findElementsMatchingLabel(driver, "twitter");
    Assert.assertThat(els, IsCollectionWithSize.hasSize(1));
    els.get(0).click();
    Assert.assertEquals(driver.getCurrentUrl(), "https://twitter.com/testdotai");
```

test.ai
We're building the future of automated testing
Get the latest on AI and software testing in our weekly newsletter
SUBSCRIBE
About Us
Blog
Press
Partner
Careers
Research

Possible locators:
css=.fa-twitter

3. Result Analysis & Reporting

- Identify / classify results from reports
- Predict quality / result based on code changes



NAME	START TIME	TOTAL	PASSED	FAILED	SKIPPED	PRODUCT BUG_1	AUTO BUG	SYSTEM ISSUE	TO INVESTIGATE	□
Superadmin_TEST_#34 Analisys ••• defualt user tag_1 Description of this launch	2016-04-05 16:23:10	43	42	1			1	29	29	
Superadmin_TEST_#35 ⚠ defualt user registration123456 Demo 12	5 days ago	10 789	9 000	789	1000		98	89	76	□
Superadmin_TEST_#36 ••• -54 s left defualt user tag_1 Description of this launch	5 days ago	10	1	8	1	9		29	27	
Superadmin_TEST_#37 18s defualt user tag_1	5 days ago	20	3	1	16		7	68	34	□
Superadmin_TEST_#38 toreport test cases 10h 32m defualt user tag_1 Description of this launch	5 days ago	1	1						29	□
Superadmin_TEST_#39 toreport test cases	5 days ago	3	1		2		3		33	□

Test Autonomy Level Definitions

0. Manual Testing (“Unassisted”)	• Manual exploratory and regression testing physically carried out by people.
1. Scripted Automation (“Hands On”)	• Handcrafted test automation scripts that can repeatedly execute a test case, sometimes with basic self-healing attributes if the application structure changes.
2. Exploratory Bots (“Hands Off”)	• Automated semi-intelligent exploration of apps and some measurement of performance/stability without human intervention.
3. Human-Directed Regression (“Eyes Off”)	• Humans describe the high-level intent of a test case e.g. “Add two items to shopping cart, delete one, and make sure there isn’t a crash.” Automated machines autonomously decide out how to execute the intent of the regression test case on one or more applications.
4. Generative (“Mind Off”)	• Machines generate and execute most test coverage across exploration, and regression testing efforts.
5. Fully Autonomous (“Human Tester Optional”)	• Without human assistance, machines are able to evaluate an application, decide what, when, where and how testing should be performed, and summarize the results for humans (or machines) to make a release/no-release decision based on test results.

4. Other opportunities

- Continuous testing (shift left & right)
- Chaos engineering
- RPA

Automate **entire Testing** NOT just **Test Execution!**

Even better: automate whole development process.

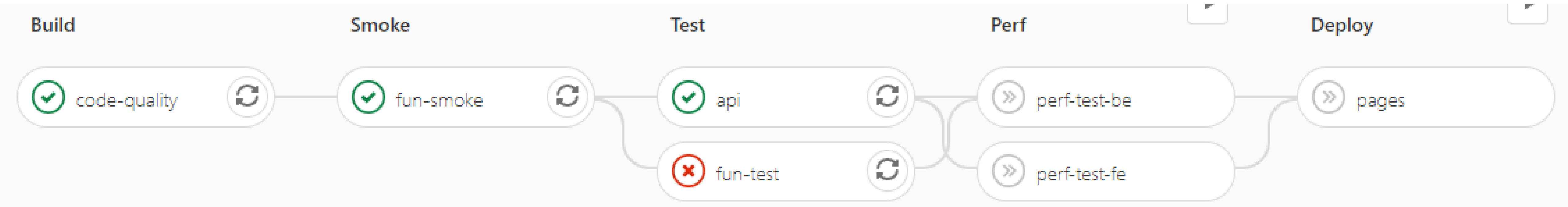
// continuous testing: . . . integrate testing into your DevOps pipelines

Continuous testing

(shift left & right approach)

- Automate all testing stages
  - Static analysis & scanners
  - Dynamic Testing
    - Test automation
    - Exploratory testing
  - Monitoring & phased (canary) deployment

- Continuous testing Workshop(s)
  - follow us @ tesena.com



Chaos engineering

- Experimenting on a software system in production
- Goal: to build confidence in the system's capability to withstand turbulent and unexpected conditions
- Effective (testing) method for modern microservice architecture
- **Chaos Monkey**: tool invented in 2011 by Netflix to test the resilience of its IT infrastructure

Robotic Process Automation (RPA)

- Automating processes in production
- Tools we use for test automation could be used in RPA
- Tools: UIPath, Robot framework, Blueprism or RPA Studio

UiPath™ blueprism® TRICENTIS RPA Studio

Trends in Test Automation

1. Old ideas are coming back

- More tools (support API & microservices, simulation)
- Record & play is back!
- Selenium era is over?

2. Fixing the pain of scale

- Intelligent test execution
- Intelligent maintenance (self-healing)

3. AI & ML

- Top AI/ML opportunities in Automation
- Autonomous Test Automation

4. Other opportunities

- Continuous testing (shift left & right)
- Chaos engineering
- Robotic Process Automation

World is changing

Don't miss the train!

Prague
Headquarter
Budějovická 1550/15a,
Prague 4, 140 00
Czech republic

Brno
Delivery center
Cyrilská 7
Brno, 602 00
Czech republic

Bratislava
Delivery center
Karadžičova 2
Bratislava, 811 09
Slovakia

London
Sales office
27 Old Gloucester Street
London, WC1N 3AX
Great Britain

Vienna
Sales office
Linke Wienzeile 4,
Wien, 1060
Austria

tesena
SMART TESTING

Thank You!

www.tesena.com

linkedin.com/in/marcelveselka/

Prague
Headquarter
Budějovická 1550/15a,
Prague 4, 140 00
Czech republic

Brno
Delivery center
Cyrilská 7
Brno, 602 00
Czech republic

Bratislava
Delivery center
Karadžičova 2
Bratislava, 811 09
Slovakia

London
Sales office
27 Old Gloucester Street
London, WC1N 3AX
Great Britain

Vienna
Sales office
Linke Wienzeile 4,
Wien, 1060
Austria