

ИСПОЛЬЗОВАНИЕ ГИБРИДНОЙ РОЛИ —
«MIXED QA» — РУЧНОГО
ТЕСТИРОВЩИКА И АВТОМАТИЗАТОРА



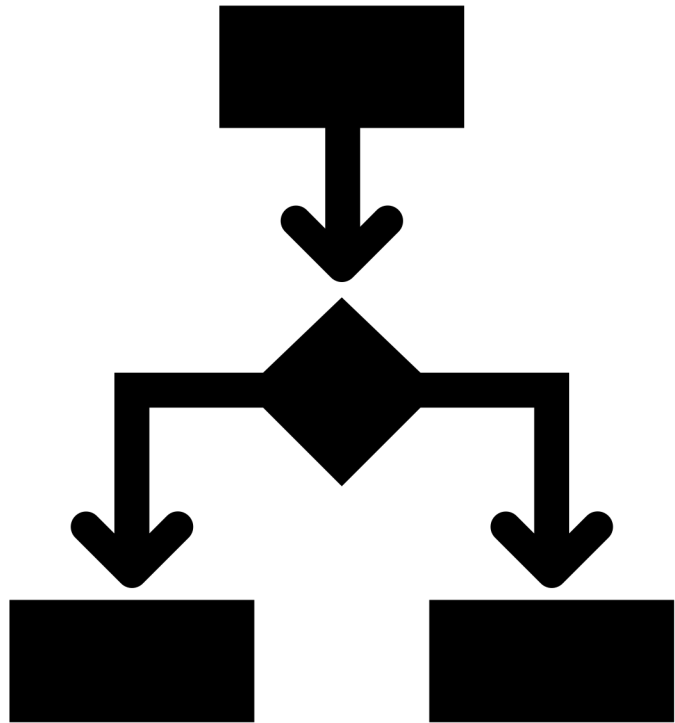
INTRODUCTION

СТАНИСЛАВ БАДОВ

- 10 лет в IT, 6 лет в Quality Assurance;
- Testing Team Leader;
- Управление тестированием и построение процессов в телеком домене на больших проектах в Швейцарии и Бельгии;
- Ресурсный менеджер юнита Mixed QA инженеров в EPAM Systems.

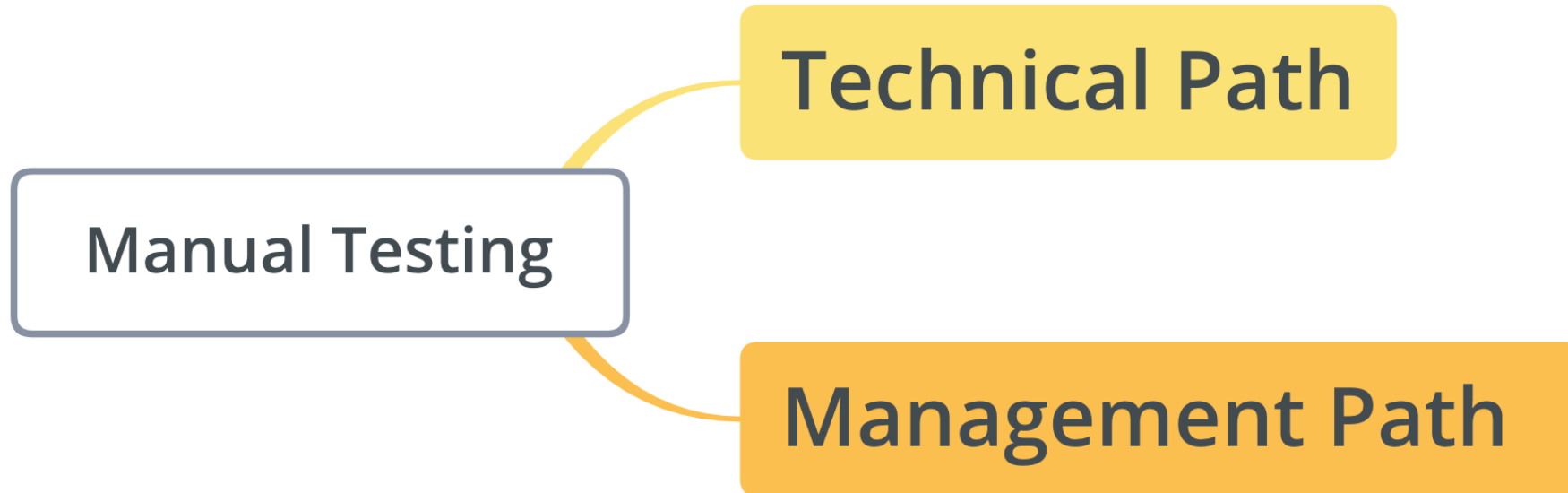
AGENDA

- Career Development Paths for Manual QA's
- Path to Automation
- Hybrid Role Manual + Automation Testing
- Tools and Frameworks with Quick Accession and Required Technical Skills
- How it Works

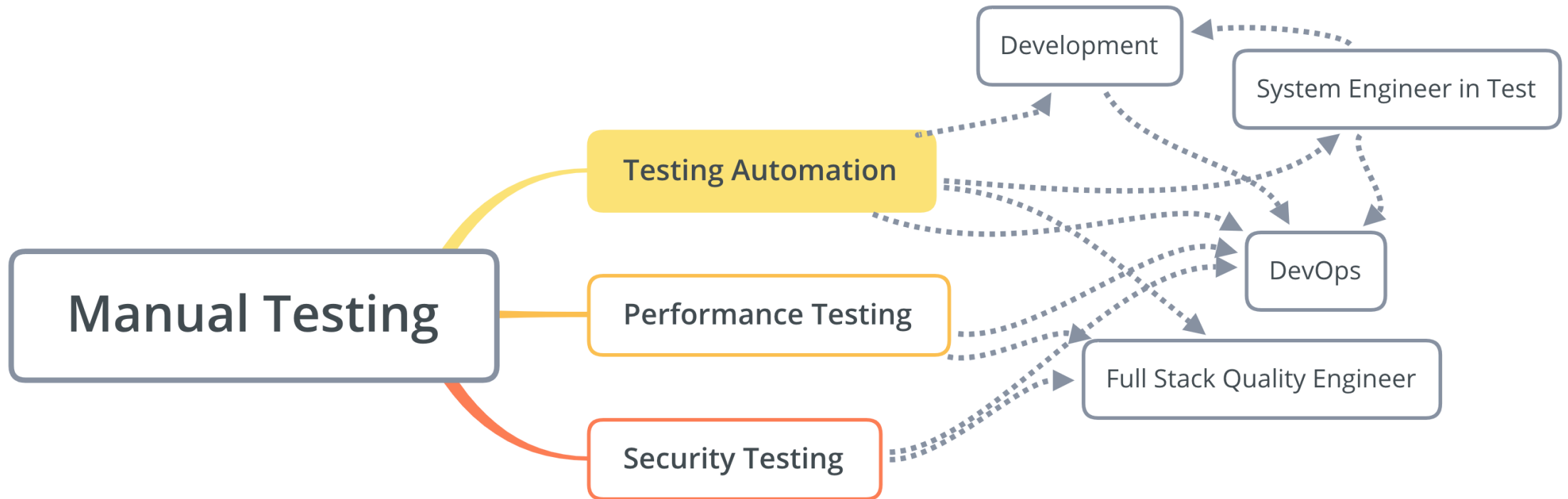


CAREER DEVELOPMENT

CAREER DEVELOPMENT PATHS FOR MANUAL QA



TECHNICAL CAREER PATHS FOR MANUAL QA



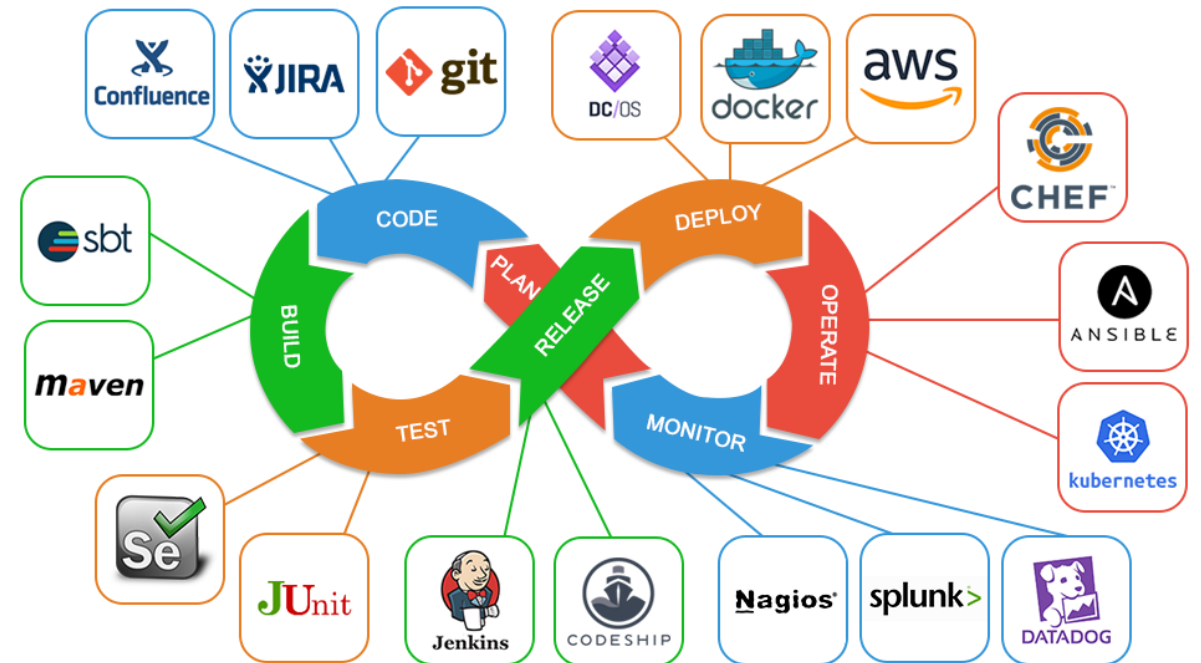
DEMANDS AND GROW

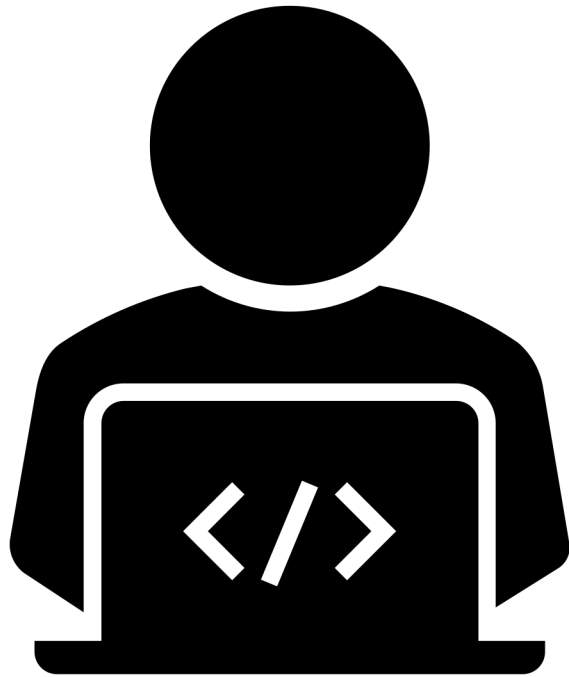
Professional grow

- Grow competence and competitive
- Salary increase

Projects demands

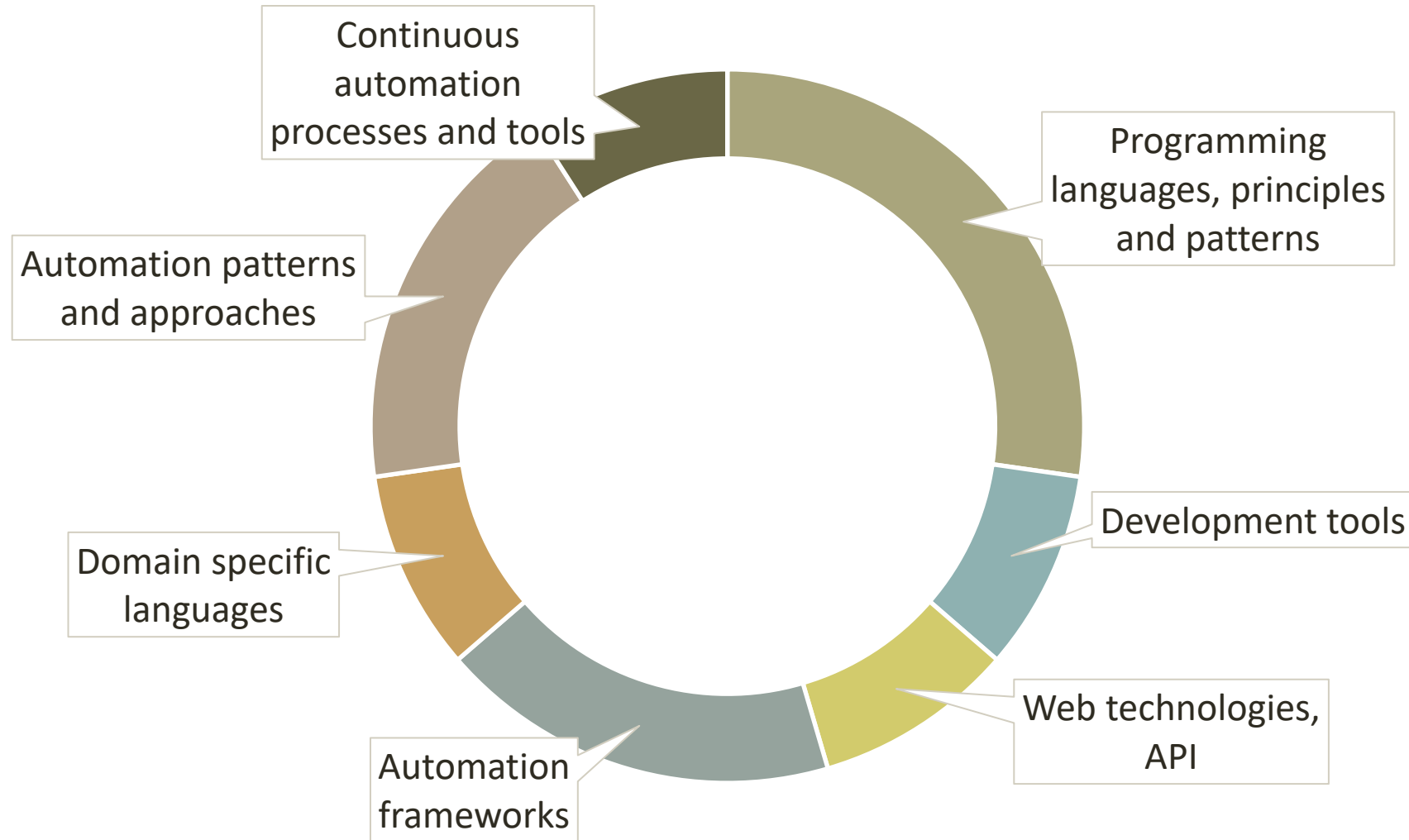
- Agile development, reduce time to production
- Continuous processes and automation
- More sophisticated tools
- Hybrid roles



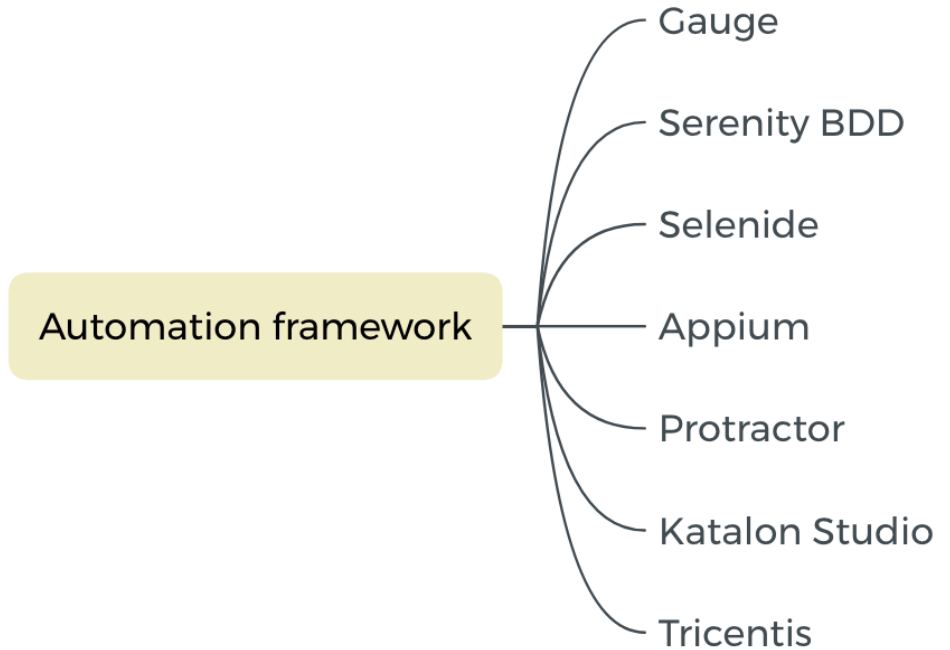
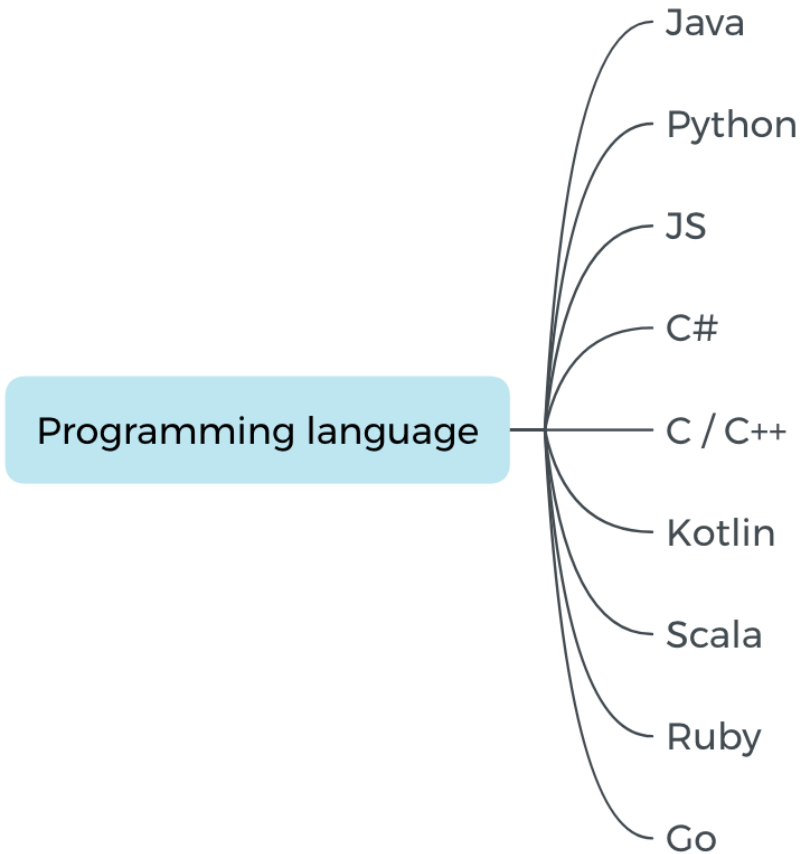


PATH TO AUTOMATION

AUTOMATION ENGINEER SKILLS SET

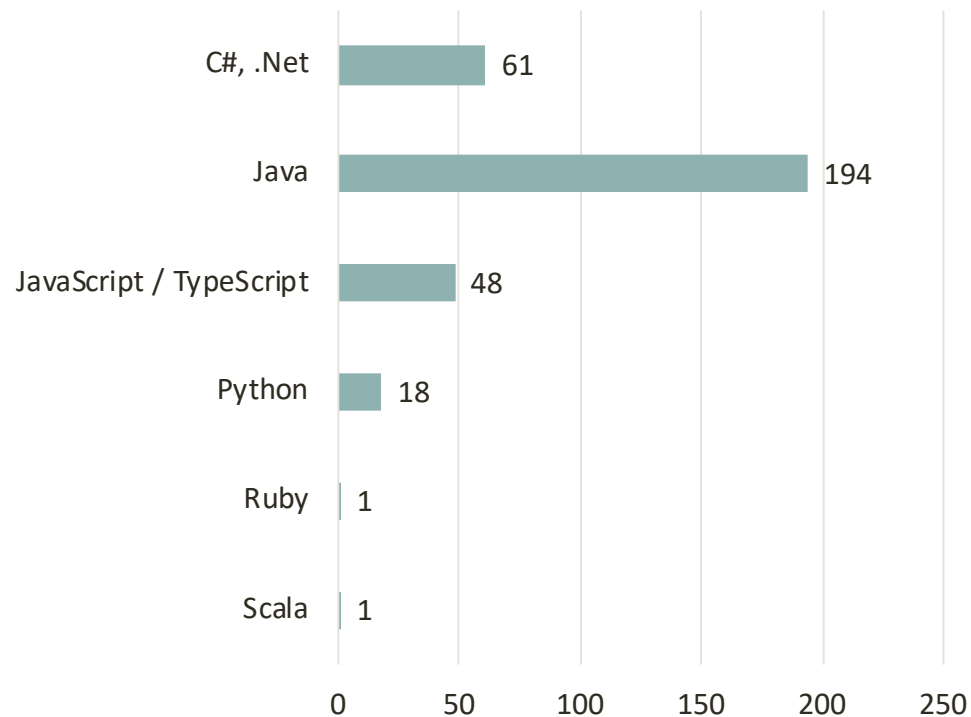


CREATE A ROAD MAP

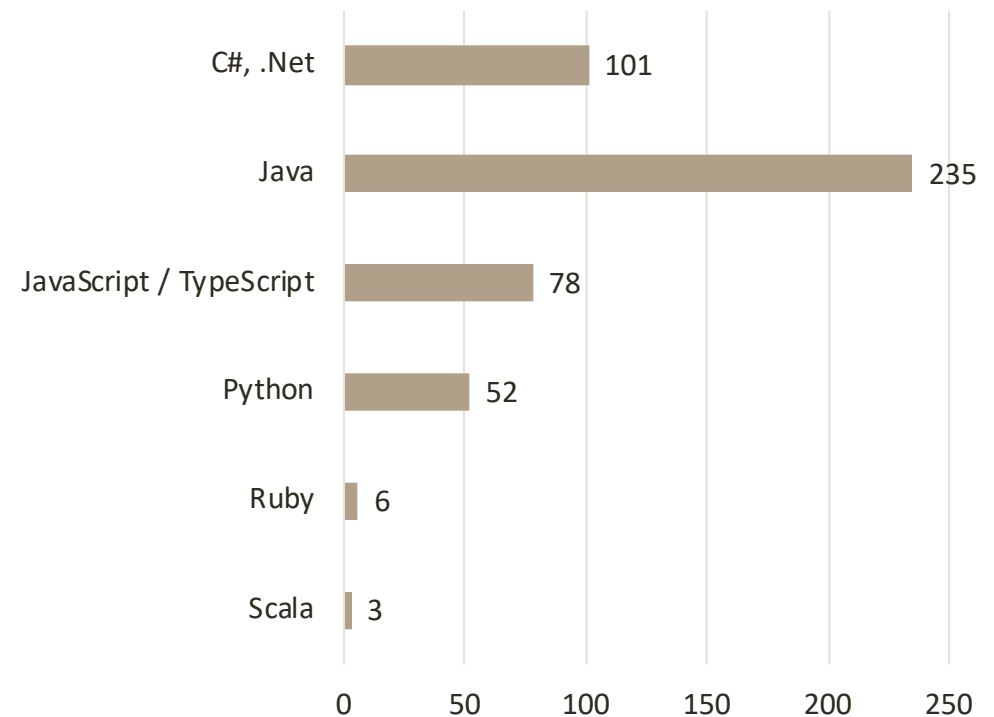


ENGINEERS MOVED FROM MANUAL TO AUTOMATION TESTING

Use on the project



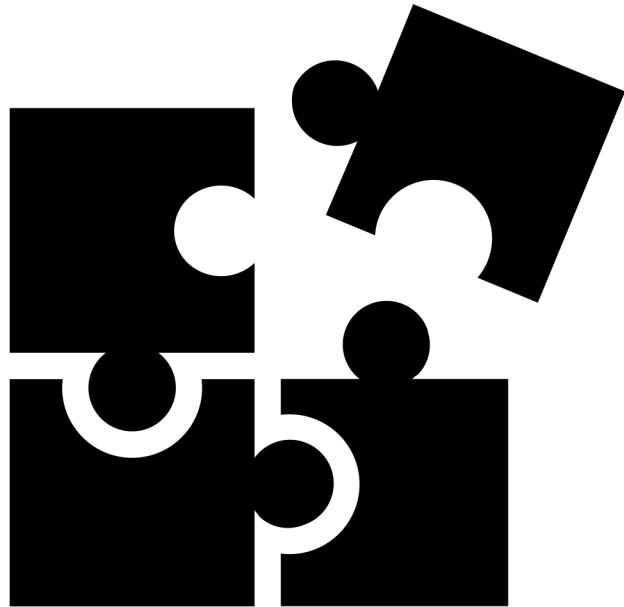
Experience in



SOURCES OF EDUCATION

Internet resources:	YouTube, Software-Testing.ru
Online courses:	Udemy, Coursera, Lynda, Codecademy
Mentoring programs:	EPAM E1toE2, E2toE3, TtoE
Courses in IT companies:	EPAM learn.by
Internal learning resources	
Books	
Colleagues	

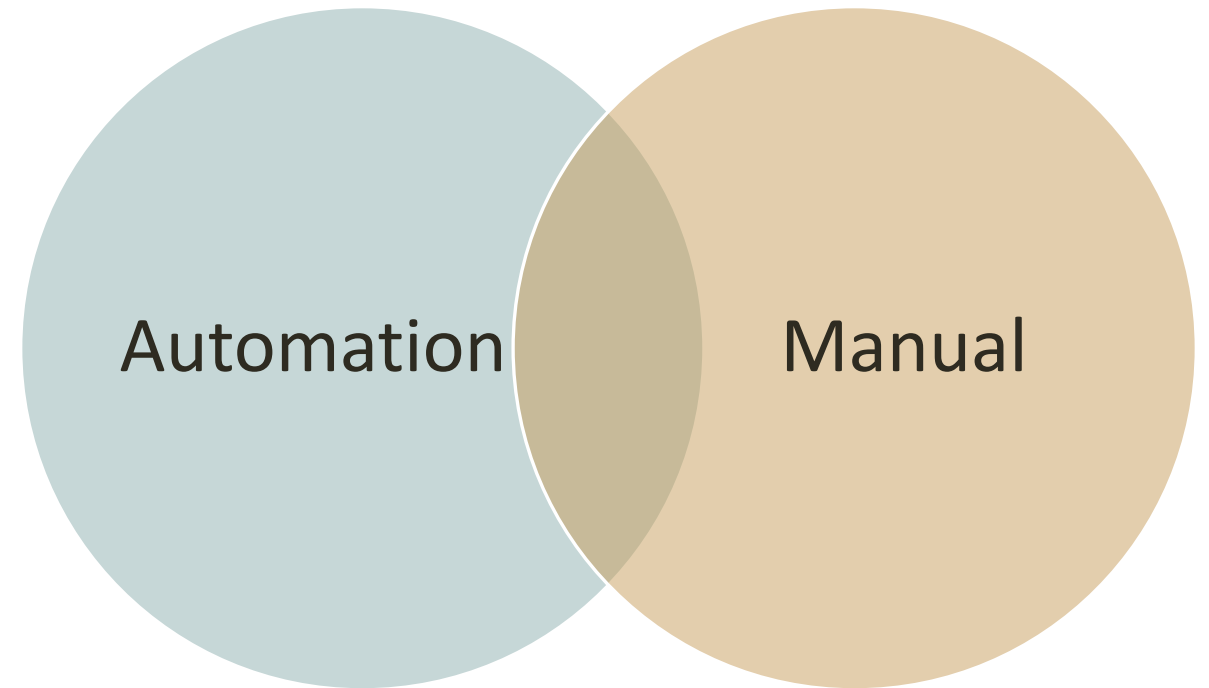
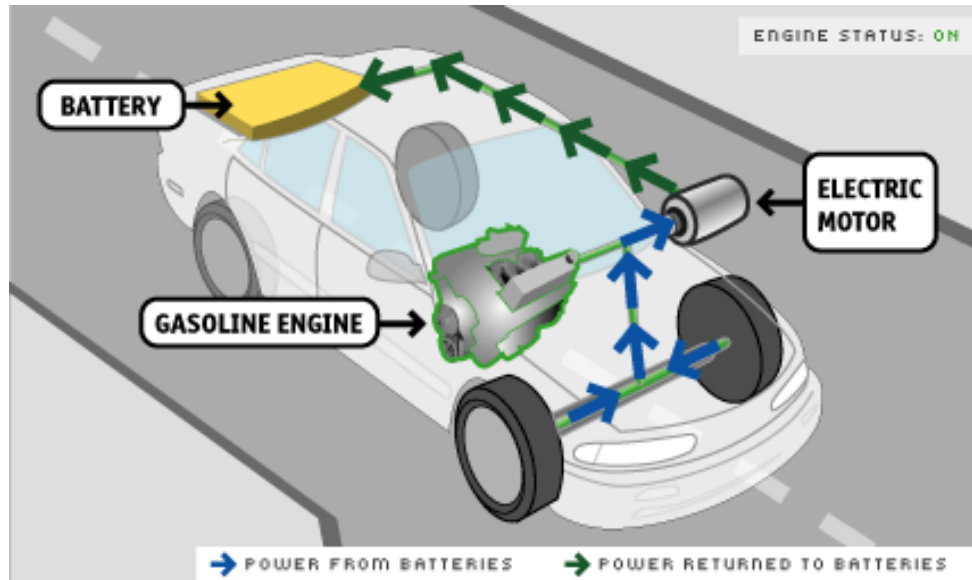




HYBRID ROLE MANUAL + AUTOMATION

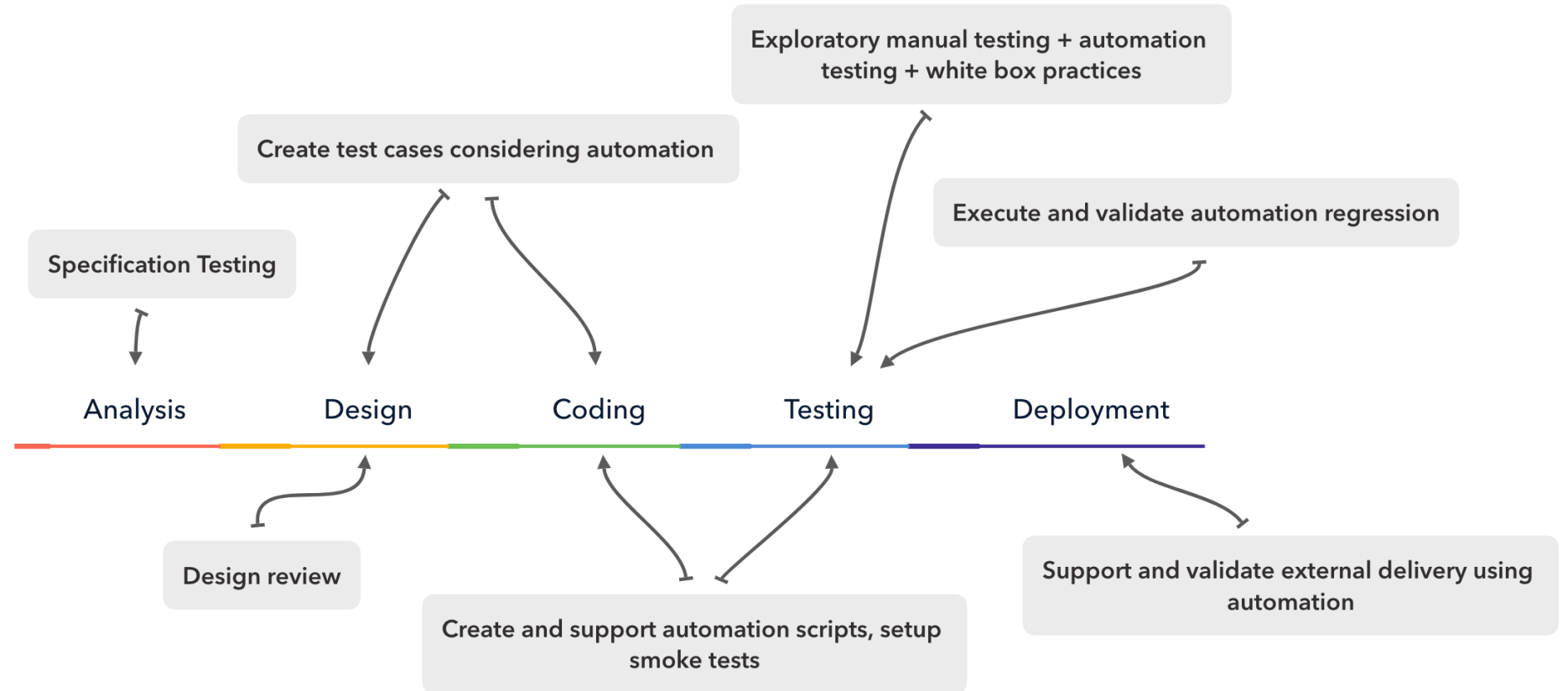
AUTOMATION + MANUAL TESTING

Advantages from the joint technologies as in a hybrid car engine

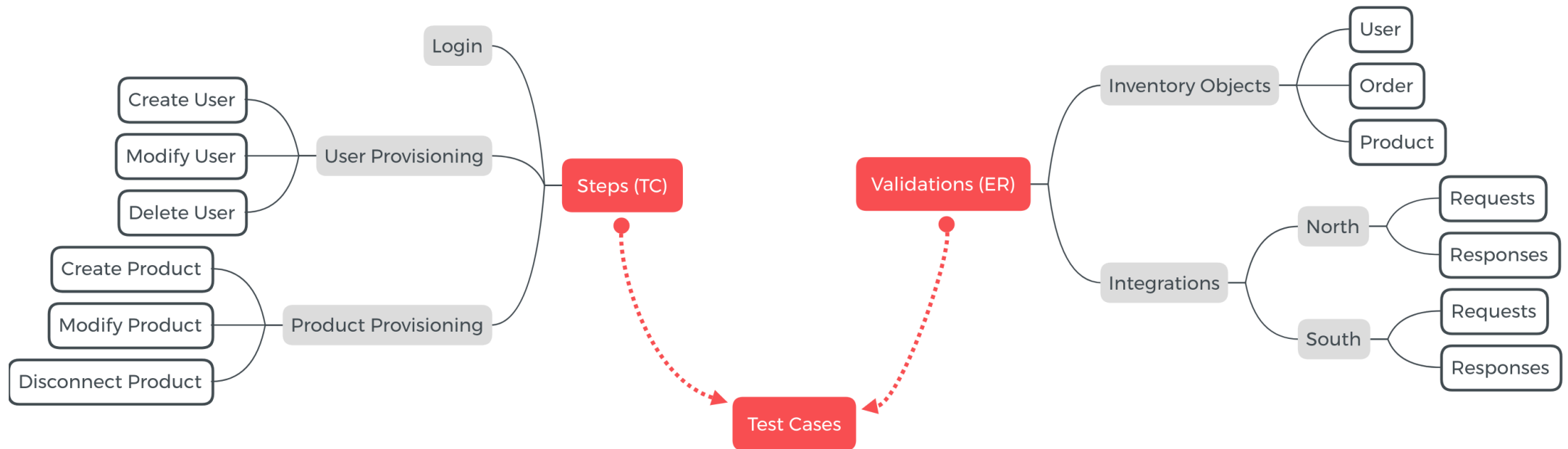


PRACTICAL USE

Development Process



TEST CASES LAYOUT



 Confluence+  Excel /  Jira

WHAT CAN WE ACHIEVE

PROS

- View on the testing processes from an user and programming sides
- Introduce test automation practices into manual testing processes and vice versa
- Create efficient test coverage
- Improved estimations and risks assessment

CONS

- High work load
- More responsibility
- Blurry expertise



FRAMEWORKS REQUIRING HIGH TECHNICAL SKILLS

GAUGE

The open source Test Automation Framework from ThoughtWorks

Features

- Write specifications in markdown
- Script tests in the language/s of their choice (Javascript, C#, Python, Java, Ruby)
- IDE plugins (Visual Studio, VS Code, IntelliJ)
- Run tests in parallel
- Generate custom reports
- Develop bespoke plugins for custom requirements

The screenshot displays the VS Code interface with the Gauge test framework. The Explorer view on the left shows the project structure, including a 'tests' directory with various JavaScript files. The main editor area is split into three panes. The left pane shows the 'pageActions.spec' file with a table of test steps. The middle pane shows the corresponding JavaScript code for the 'Write, Press, Click' scenario, with a red arrow pointing to the 'Press Enter' step. The right pane shows the JavaScript code for the 'Hover' scenario. The status bar at the bottom indicates the current file is 'Ln 15, Col 1' and the language is JavaScript.

```
12 | |Type|Selector |
13 | |---|-----|
14 | |link|Calculator|
15 |
16 |
Run Scenario | Debug Scenario
## Write, Press, Click
17 |
18 |
19 | * Navigate to "http://todomvc.com/examples/react/#/"
20 | * Write "flow"
21 | * Press "Enter"
22 | * Click link "Active"
23 | * Click checkBox with attribute "{\`class\`: \"toggle\"}" near
24 |
25 | |Type|Selector|
26 | |---|-----|
27 | |text|flow |
28 | * Click link "Completed"
29 | * Click button "Clear completed"
30 |
31 | Run Scenario | Debug Scenario
32 | ## Right Click
33 |
34 | * Navigate to file with relative Path "/specs/rightClick.html"
35 | * Right click
36 |
37 | |Type|Selector |
38 | |---|-----|
39 | |text|Someelement |
40 | * Click "Share On Facebook"
41 |
42 | Run Scenario | Debug Scenario
43 | ## Hover
44 |
45 | * Navigate to "http://react-compare-app.surge.sh/"
46 | * Hover on element
47 |
48 | |Type|Selector |
49 | |---|-----|
50 | |$ | ".image_overlay"|
51 | * Click "Compare"
52 | * Hover on element
53 |
54 | |Type|Selector |
55 | |---|-----|
56 | |$ | ".image_overlay"|
57 | * Click "Remove"
58 |
59 | Run Scenario | Debug Scenario
60 | ## Drag and drop
61 |
62 | * Navigate to "https://marcojakob.github.io/dart-dnd/basic/"
```

GAUGE

Steps example

The screenshot displays the VS Code interface with the Explorer sidebar on the left, the Explorer view in the center, and the Editor view on the right. The Explorer sidebar shows a project structure for 'JS-TAIKO-MASTER' with a 'tests' folder highlighted. The Explorer view shows the 'click.js' file selected. The Editor view shows the code for 'click.js', with a red box highlighting a specific step function. A red arrow points from this step to a context menu for 'proximitySelectors.spec', which shows the execution output for the 'Click link "Insights" below' step.

```
1 'use strict';
2 const assert = require('assert');
3 var _selectors = require('./selectors')
4
5 const {
6   link, click,below,image,above,toRightOf,toLeftOf,button,rightClick
7 } = require('!taiko');
8
9 1 reference(s)
10 step("Click link <userlink> below <table>", async function (userlink,table) {
11   | await click(link(userlink,below(_selectors.getElement(table))));
12 });
13
14 4 reference(s)
15 step("Click an element that contains <text>", async function (text) {
16   | await click(text);
17 });
18
19 4 reference(s)
20 step("Click link <userli
21   | await click(link(use
22 });
23
24 0 reference(s)
25 step("Click <selector> w
26   | await click(link(sel
27 });
28
29 12 reference(s)
30 step("Click <selector>",
31   | await click(selector
32 });
33
34 1 reference(s)
35 step("Click image above
36   | var element = _selectors.getElement(table);
37   | await click(image(above(element)));
38 });
39
40 1 reference(s)
41 step("Click link to right of <table>", async function(table) {
42   | await click(link(toRightOf(_selectors.getElement(table))));
43 });
44
```

proximitySelectors.spec ~/projects_js/js-taiko-master/specs

```
Run Spec | Debug Spec
# Proximity Selectors
Run Scenario | Debug Scenario
## Below
* Intercept Google Analytics
* Navigate to "https://gauge.org"
* Click link "Insights" below
|Type|Selector      |
|----|-----|
|link|Gauge Commands|
```


GAUGE

Built in report example

The screenshot displays a Gauge HTML report for the project 'js-taiko-master'. The report includes a summary table, a specifications sidebar, and a detailed view of a failed test case.

		Failed	Passed	Skipped
TOTAL SPECS	10	5	5	0
Total Scenario	46	5	41	0

Environment : default
Success Rate : 50%
Total Time : 00:03:20
Generated On : Mar 26, 2019 at 2:15pm

Specifications

Name	Execution time
API	00:00:29
API accessing IFrame Elements	00:00:29
Alert	00:00:04
Proximity Selectors	00:00:24
Selectors	00:00:16

API 00:00:29

Radio Button without for 00:00:11
Tags: knowlissue
Execution Time : 00:00:01

Navigate to relative path "/specs/data/HTMLElements.html"

Execution Time : 00:00:10

Radio Button "Y"

```
AssertionError [ERR_ASSERTION]: false == true
AssertionError [ERR_ASSERTION]: false == true
```

Combo Box 00:00:03
Execution Time : 00:00:02

SERENITY BDD

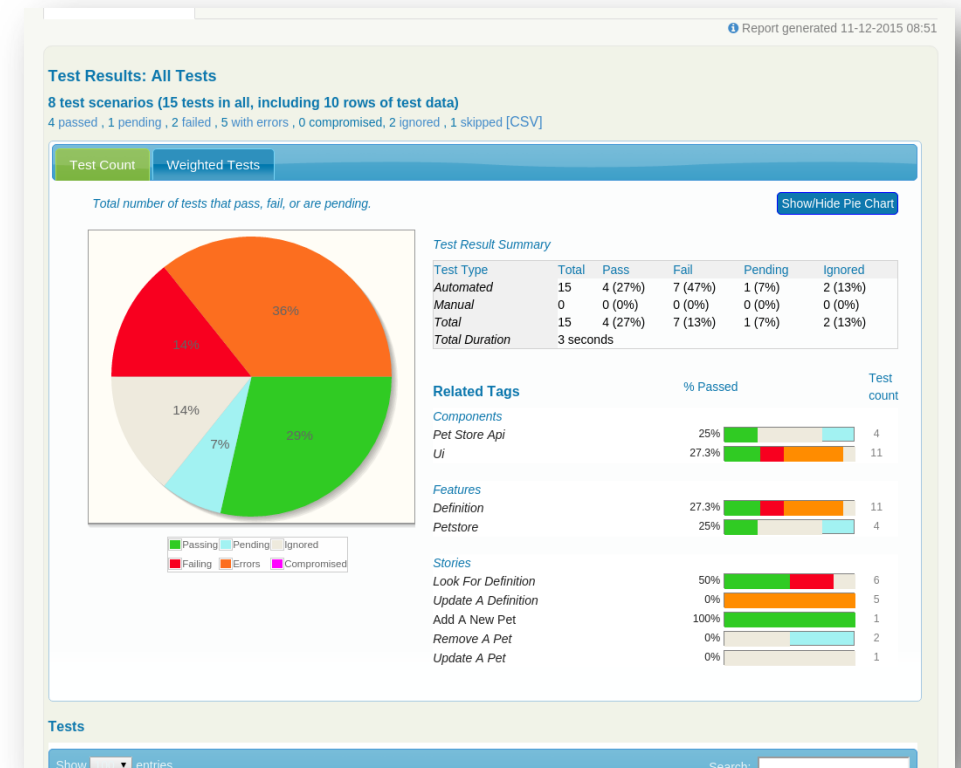
Serenity BDD is a test automation framework and open source library.

Behaviour Driven Development is a core concept underlying many of Serenity's features:

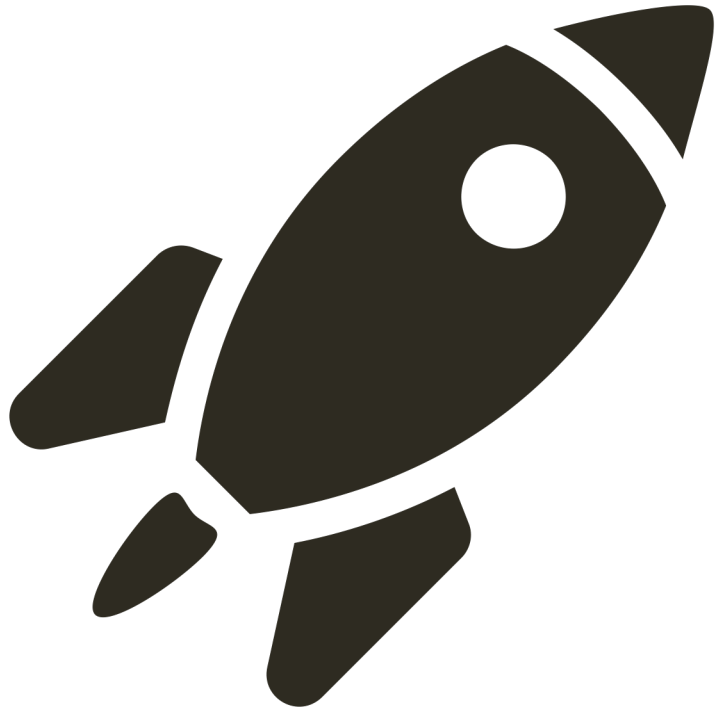
- Automated acceptance tests using BDD
- Powerful reporting and living documentation features
- Web testing with Selenium
- API testing using REST Assured

Serenity BDD supports several design patterns:

- Classic Page Objects
- Lean Page Objects/Action Classes approach
- Screenplay pattern



```
1 @Before
2 public void setTheStage() {
3     OnStage.setTheStage(newOnlineCast());
4 }
5
6 @Given("^(.*) is on the DuckDuckGo home page")
7 public void on_the_DuckDuckGo_home_page(String actor) {
8     theActorCalled(actor).attemptsTo(
9         NavigateTo.theDuckDuckGoHomePage()
10    );
11 }
12
13 @When("she/he searches for {string}")
14 public void search_for(String term) {
15     theActorInTheSpotlight().attemptsTo(
16         SearchFor.term(term)
17    );
18 }
19
20 @Then("all the result titles should contain the word {string}")
21 public void all_the_result_titles_should_contain_the_word(String term) {
22     theActorInTheSpotlight().should(
23         seeThat("search result titles",
24             SearchResult.titles(),
25             hasSize(greaterThan(0))),
26         seeThat("search result titles",
```

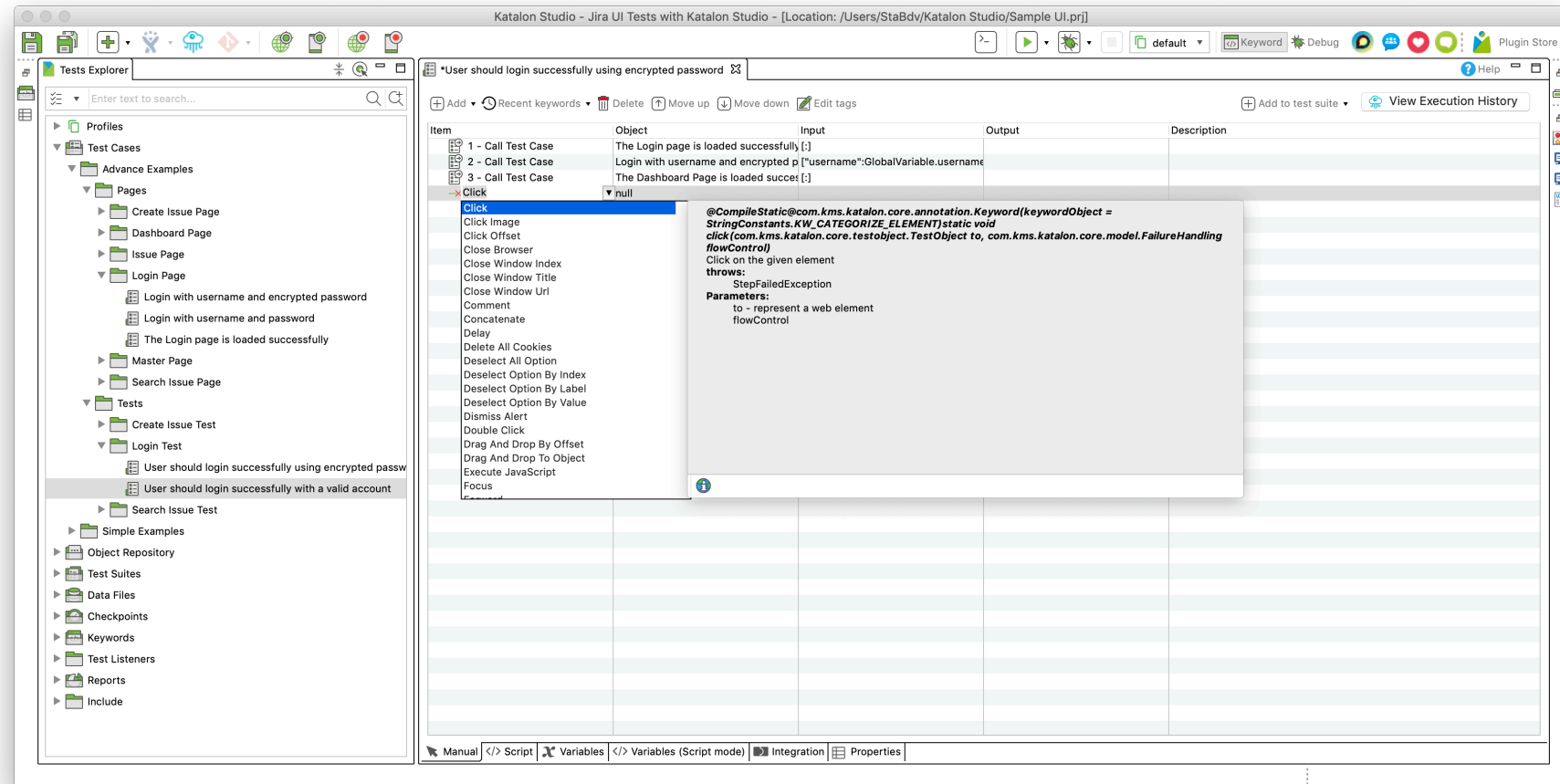


FRAMEWORKS WITH QUICK ACCESSION

KATALON STUDIO

Features

- Test Case Recording
- Keyword-Driven Testing
- Cross-browser Testing
- Parallel Execution
- API Testing
- Data-driven Testing
- TDD/BDD Testing
- Extension & Customization
- SDLC Integration
- Advanced Reports
- Debugging
- CI Integration

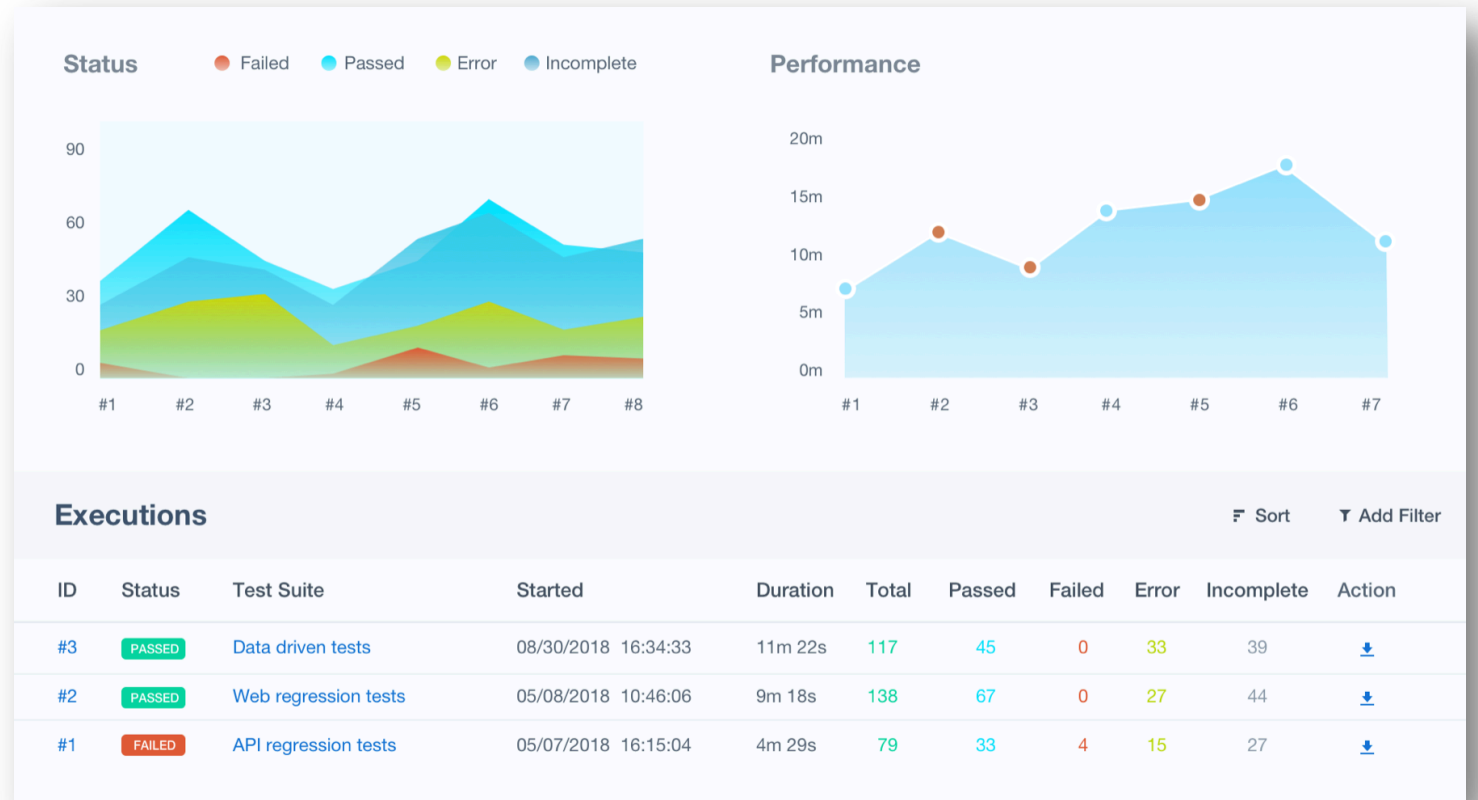


KATALON ANALYTICS

The web-based application provides reports through visualization including charts and graphs.

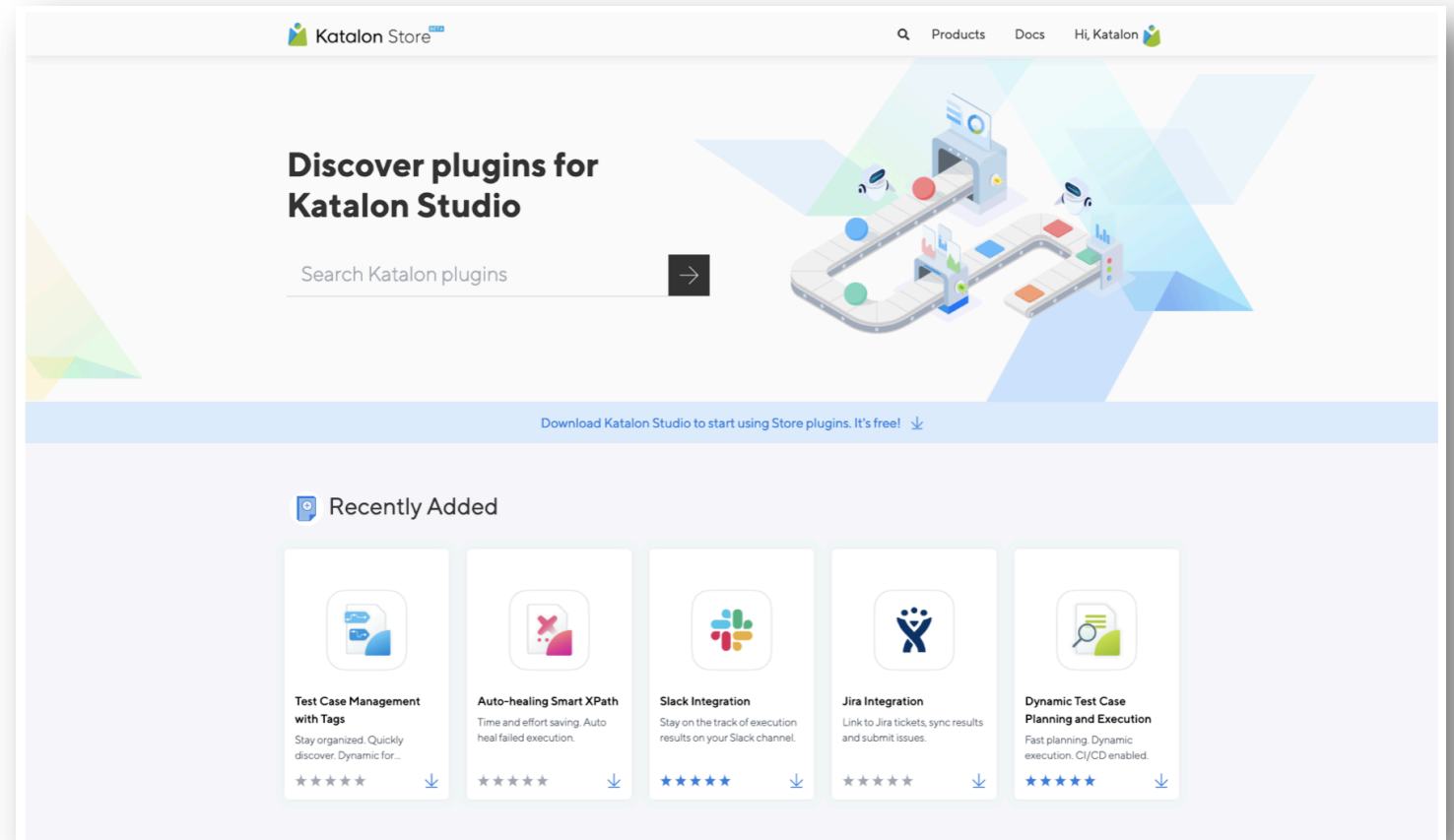
Supports

- Status Report
- Performance Report
- Executions Report
- Execution Detail Report



KATALON STORE

Katalon Store serves as a platform for testers and developers to install add-on products, also known as 'plugins,' to help our users add more features and optimize test automation strategies in Katalon Studio.



TRICENTIS TOSCA

Continuous end-to-end testing platform

Tricentis Tosca automates tests across all layers of modern enterprise architectures.

Features:

- SAP Testing
- Packaged App Testing
- Functional Test Automation
- Model-Based Test Automation
- Risk-Based Testing
- BI/Data Warehouse Testing
- Load Testing



TRICENTIS TOSCA

The screenshot shows the Tricentis Tosca interface with a test case configuration table. The table has columns for Name, Value, ActionMode, DataType, and WorkState. The test case is named 'FirstTest' and is in a 'PLANNED' state. It contains several steps, including 'Click on Login Link' and 'Enter Credentials'.

Name	Value	ActionMode	DataType	WorkState
FirstTest				PLANNED
Click on Login Link				
Register			String	
Log in	{Click}	Input	String	
Shopping cart(0)			String	
Books			String	
Computers			String	
Electronics			String	
Enter Credentials				
Email:	test@testauto...	Input	String	
Password:	*****	Input	Password	
Log in	{Click}	Input	String	
OpenUrl				

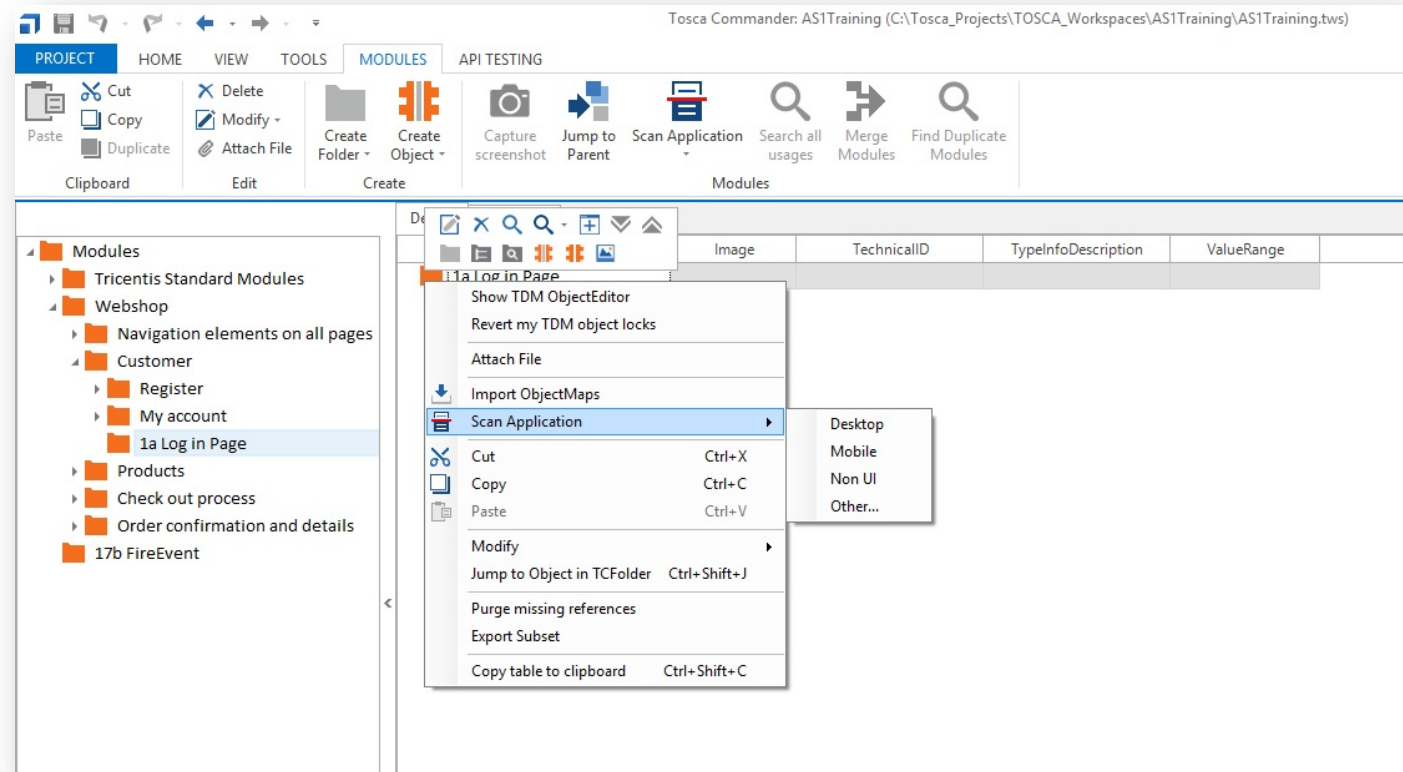
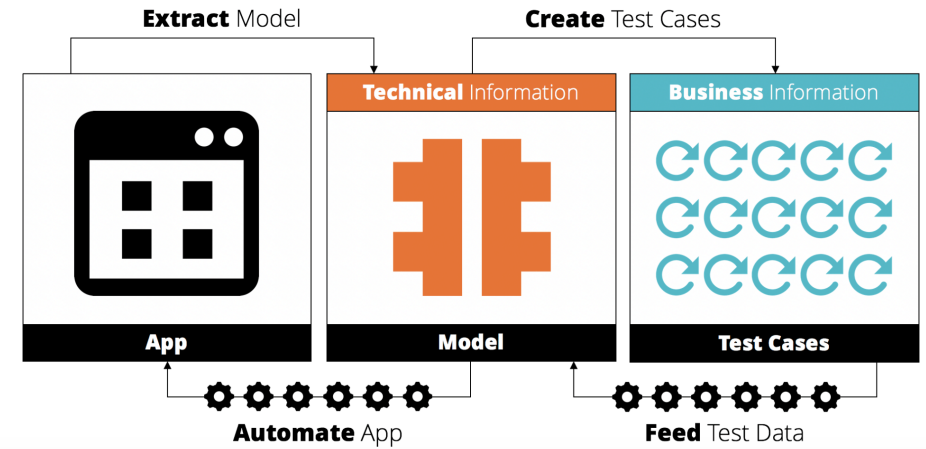
The screenshot shows the Tricentis Tosca interface with a test case design and execution summary. The test case is named 'Web shop' and is in a 'PLANNED' state. It contains several steps, including 'Navigation elements on all pages', 'Customer', 'Products', 'Check out process', and 'Order confirmation and detail'. The execution summary table shows the results of the test case execution, including the number of successful and failed test cases.

Name	Loginfo
ExecutionLists	115 35 7
Automated	45 7
Create Quote	
ActualLog	45 7
Open Sample Application	
Create Automobile Quote	14 2
Create Truck Quote	13 4
Create Camper Quote	17 1
Manual	33 13 7
Create Automobile Quote [Manual]	1
Create Automobile Quote	
Create Quote	
ActualLog	33 13 6
Open Sample Application	
Create Automobile Quote	2 8 6
Create Truck Quote	13 4
Create Camper Quote	17 1
Regression	37 15

TRICENTIS TOSCA

Model-Based Test Automation

The approach is architected to enable anyone from developers to business experts to contribute to test automation as well as eliminate the maintenance burden that erodes most test automation initiatives.



HOW TO MAKE A CHOICE?

Frameworks requiring high technical skills

PROS

- flexible, more integrations
- changes allowed
- tests quality and maintenance

CONS

- required programming skills

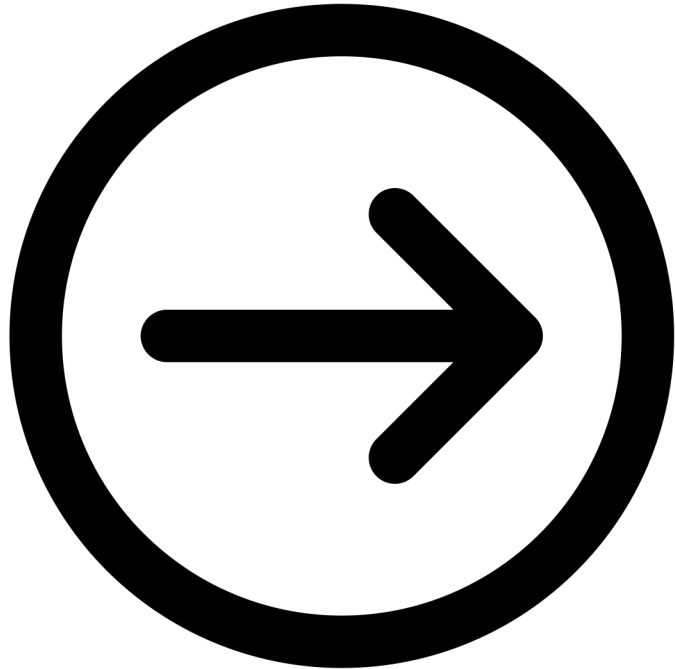
Frameworks with quick accession

PROS

- faster and easier to start
- involve more participants into tests creation
- can be automated more tests

CONS

- tests quality and maintenance
- closed for changes



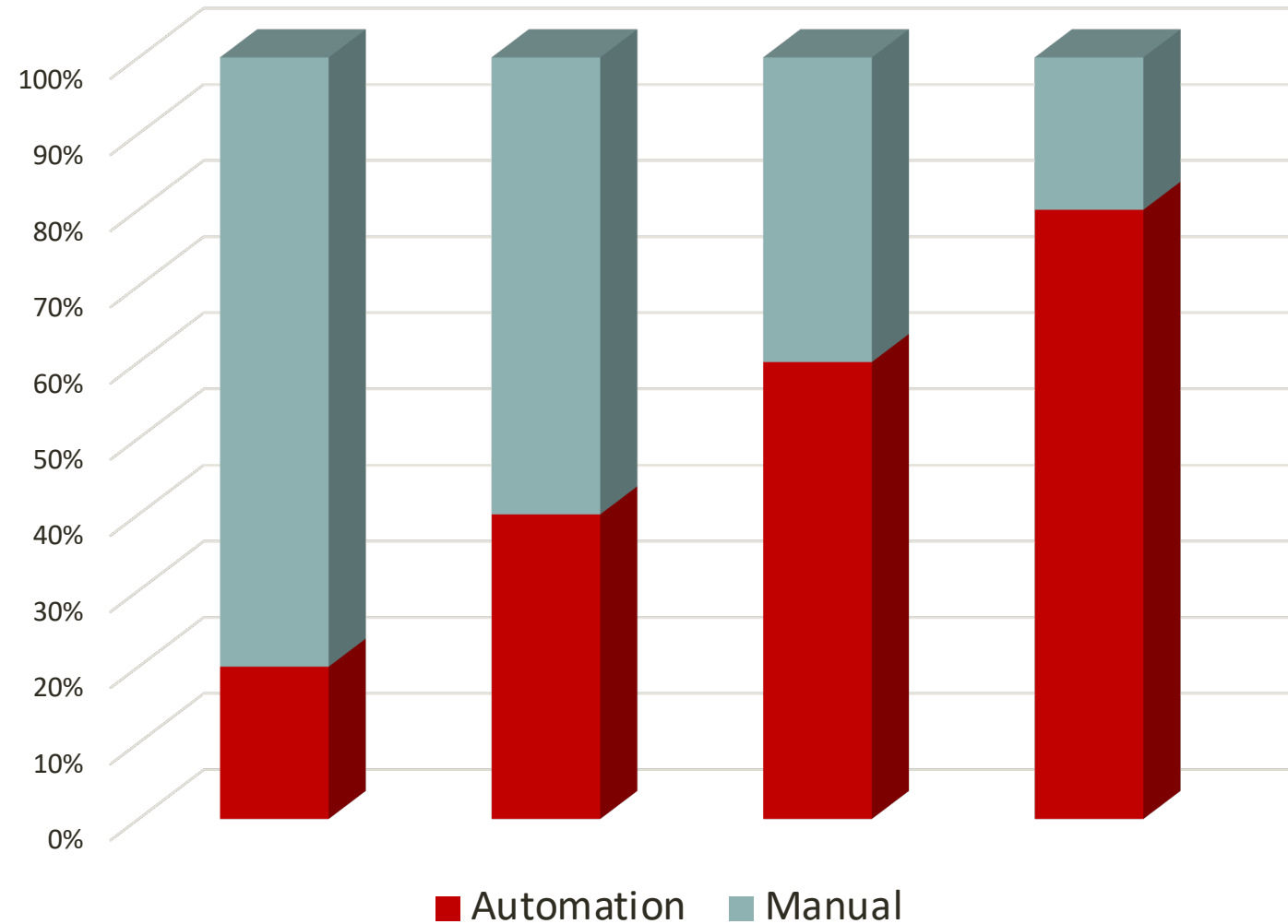
HOW IT WORKS

MIXED QA UNIT

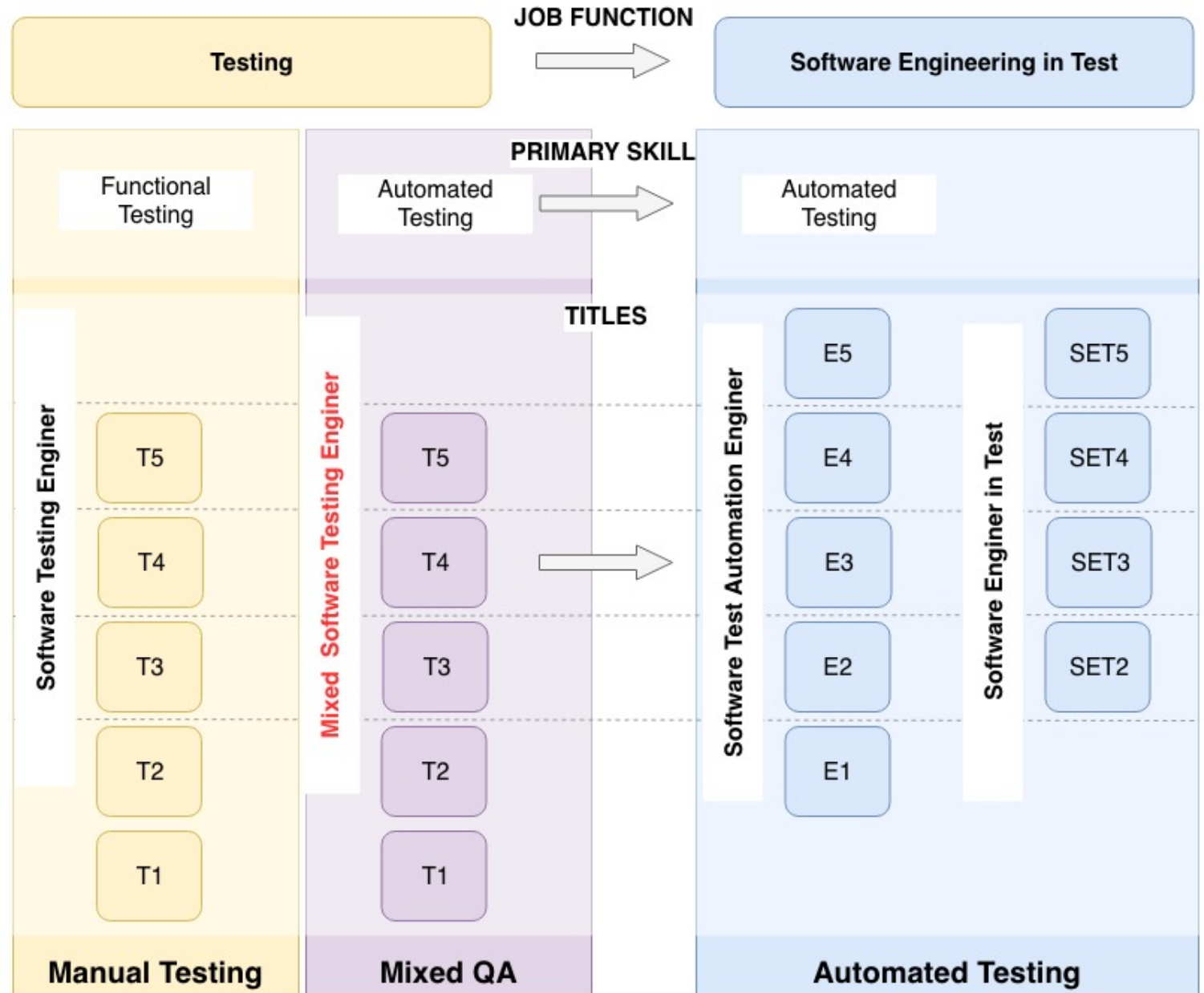
Entry criteria

- Practice experience in manual testing
- Knowledge in programming languages and principles
- Completed courses and/or some practical experience in testing automation

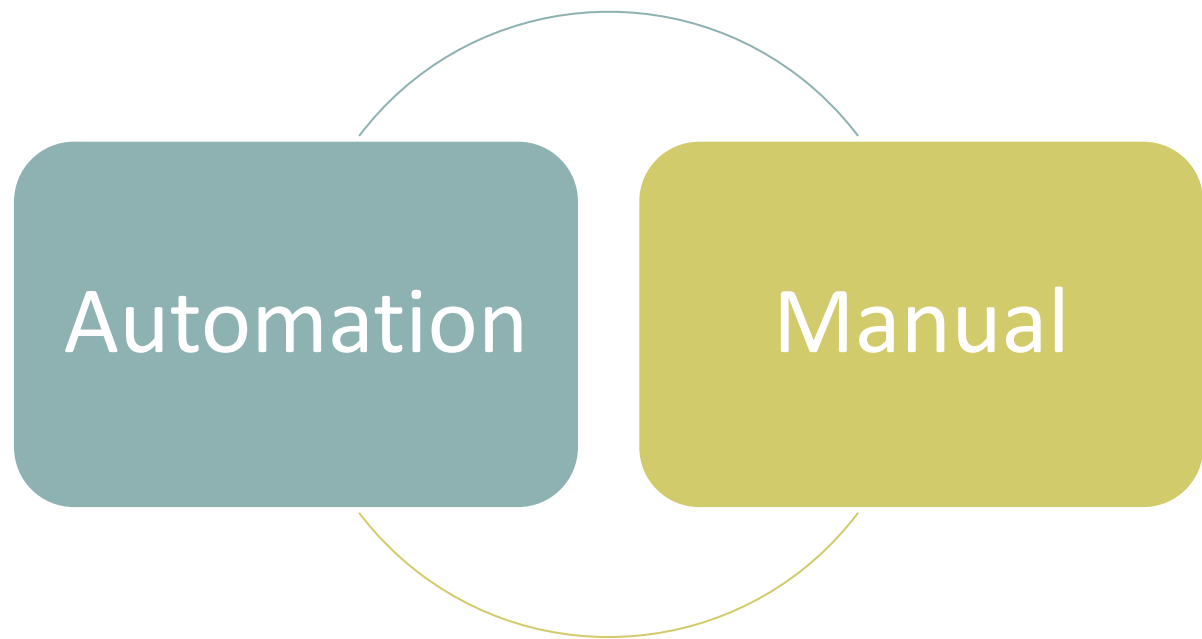
Automation and Manual Tasks Proportions



MIXED QA'S CAREER PATHS



CONCLUSION



QUESTIONS