ArtComputer

General Presentation

Automated Test with the Robot

sopra steria

Philippe Goffin
Automation Testing Tool

About the Tool

Automated testing tool for the **business** is not in the culture of the European Commission (UIPath… but not used)

We detected the need to test for **non-regression** but also to automate the **creation of records** (contracts, calls…) ready for further (manual) tests.

On complex project, the acceptance of the stakeholders is difficult due to the complexity of the application but also due to the length of the workflow.

An automation testing tool could improve the process!

History of the project

The first version was based on an Excel sheet for the user interface and Protractor / Selenium for the engine.

Main objective: build a tool that can be used by testers without any technical skill.

This version was used during 4 years on 2 big projects at the Commission (DG INTPA).

In 2024, a new version with a **web interface** was born. With very high ambitions… this new version is a real game changer:

- The code is optimised (performance increased to 35%).
- A more intuitive interface with dynamics parameters for the functions.
- Possibility to share scenarios between the projects (of the same workspace / customer)
- Export data (scenarios, dataset…) into a .json file, ready to be formatted in Excel for a discussion with the business.
- Open to a team of testers (login/password)
- ….

sopra steria

Automated Testing Tool

Roles

Designer

Designer is the main actor.
He will design the scenarios, prepare the stories for the Testers.

The user interface is optimized for productivity.

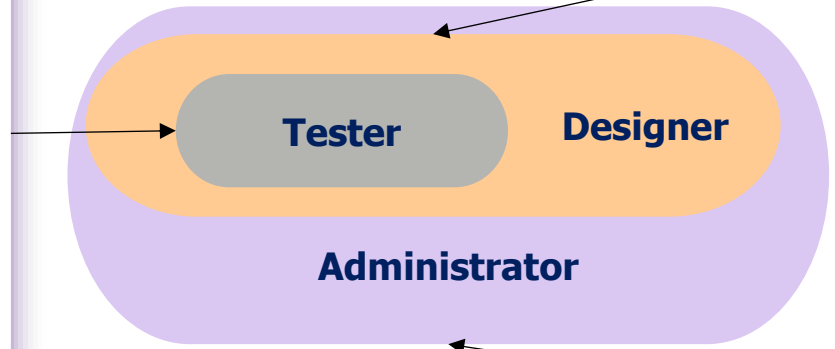
Tester

Tester can be a **Manual Tester** or a **Business Analyst**.

The user interface must be very easy to use to reduce the learning curve.

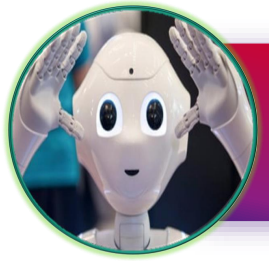
A tester can only perform 3 tasks:

- Execute a story (including steps)
- Manage the data
- Manage the dummy users (users used during the tests)



Administrator

Administrator will manage the workspace (customer space), the projects/sub-project users, the role of the users and will train the robot to allow the automatic detection of the elements on a page.



Automated Testing Tool

Technical corner: Archirecture

MVC

The MVC is designed with **Webdriver**, Selenium and Node.js/JavaScript languages.

User Interface

The web user interface is developed in **Vue 3**.

Vue 3 combines the html, the script and the CSS on the same file source. Script is based on node.js/JavaScript

Rest API

Database

The database is **MySQL** database

Automated Testing Tool

Exchange of information

References

References are used to exchange data between the scenarios.
A scenario can read or write a reference.

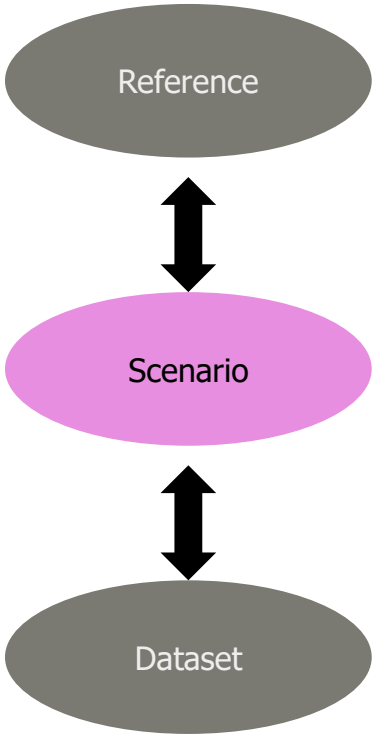
References are shared between all the sub-projects.
- ► You can create a sub-project to create a call for tenders and store the id
- ► You can create another sub-project to prepare an offer based on the call previously created

Dataset

The Dataset is used to store data to be used during the test.
A data can be static (a constant) or a specific keyword (<TODAY>, <TODAY+5>, <SEQUENCE>….)

It's a good practice to create a data driven test (the process is guided by the value of the data)

A scenario can read or write a data inside a dataset.



Automated Testing Tool

Readability

Readability

It's very important to keep the scenarios readable and not only by the **Designers**.
The **Business** must also be able to understand the flow of the scenario (for instance for **validation**).

During the definition of the steps, you need to specify a description (English text).
You can indicate if the instruction has a business added value (versus technical aspect).
In the user interface, you can filter on this indicator.

Describe and It

Although it is optional (but highly recommended), we have two instructions to structure a scenario.
Describe is used to define a chapter.
It is used to define a section.

These 2 instructions make the scenario more readable and facilitate the maintenance by another person.

1 > Describe: Display the webpage

2 > It: possible to setup the data for the test

3 > Step: Get the Call number from the reference (getReference)

4 > It: possible to display the PROSPECT webpage

5 > Step: Get the environment for the test (getReference)

Automated Testing Tool

Hiding the technical aspects

Hiding the complexity

We started from the assumption that the **Tester** was not a developer. We also want to provide an easy interface for the **Designer** without the need to compile the sources.

To achieve this goal, we designed a library with **smart functions**

Library of smart functions

Using functions to execute a test is the guarantee to have a code free of bugs (more we use the tool more we secure the code)
We designed the functions to be smart. It means, the function must be able to bring the best solution for a specific request.
Example: the function getValue() is used to get a value of a field. The function will return the best solution for the specified element.
The value can be a text, a value, an innerHTML, outerHTML…

Automated Testing Tool

Scenario like a language

Scenario is a real language

A scenario is not just a set of actions to mimic the behavior of a real user.
A scenario must be able to **make decision**, to **call a subroutine** or even to **call another scenario** depending on some conditions.

At the end, the scenario is a small language…

Condition & Loop

It's very easy to skip the execution of a step inside a scenario with a **condition**.

You can also create **loop** and **sub loop** inside your scenario.

Variables

You can store a value into a **variable** to reuse the value later in the scenario.

The value can be a constant or any valid JavaScript expression.

Automated Testing Tool

Make decision

Rules Engine

A **rule** is a set of instructions based on a **condition** and an **action.**

All the functions can be used in the rules.

Use case

The rules engine has two uses:

► Create a reusable routine (E.g.: the login process)

► Make decision:

- Create a business rule to check the validity of a value or compute a value based on some criteria
- Identify the document to upload, based on some criteria (E.g.: status: accepted, rejected…)
- ….

Automated Testing Tool

AI in the Robot

AI to detect elements

The most complex part for the **Designer** is the creation of the **Xpath** (or CSS).
Depending on the quality of the application, the construction of the Xpath can be very complex (for instance, in the case there is no ID or no Name).

► The **detectGUI**() function can resolve this issue!

Training of the Robot

Before using the function, the **Administrator** must train the Robot to recognize an element (a field, a button, a checkbox, a menu…). The type of element is dynamic and depends on your project.
For instance, on the CRIS project, we introduced the type HIDDEN to search for the signature of the hidden fields.

During the training, the Administrator plays the role of the Expert. He must be able to determine the best signature for the element by inspecting the web page.
The Robot will analyze different patterns and provide the top 5 best solutions. Expert selects the best solution

Statistics

Thanks to the statistics, the Robot is able to improve the quality of the 'signature'.
E.g.: during the first training, the Robot detects that a button requires the class "abc".
The next time you train the Robot with a button, it will be able to 'understand' that the class is not relevant.

Note: there are different strategies that you can put in place: a list of possible values or removing the attribute.

► Example of use of the function: detectGUI ("button", "Save").
The result (Xpath) will be stored in the variable $GUI

Automated Testing Tool

Natural language in the Robot

Natural language to write the scenario

Although the process is still in the prototyping phase, it is very promising.

The idea is the possibility to define a technical request in a natural language (NLP)

- "Wait a little bit" will be transposed with the function pause(5)
- "Wait for 2 seconds" will be transposed with the function pause(2)
- "Detect the button 'Save'" will be transposed with the function detectGUI ('button', 'Save')

This process could reduce drastically the learning curve of the tool!

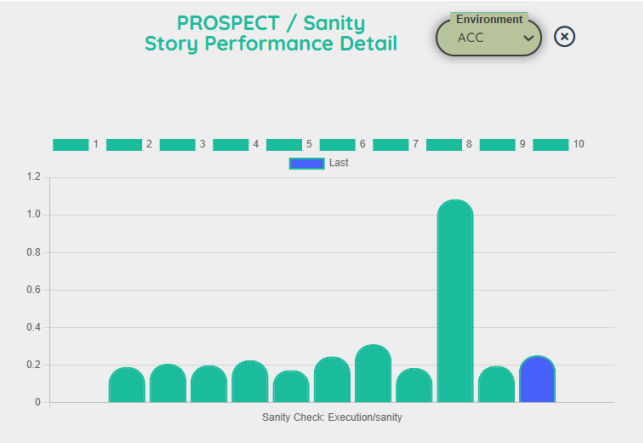
Automated Testing Tool

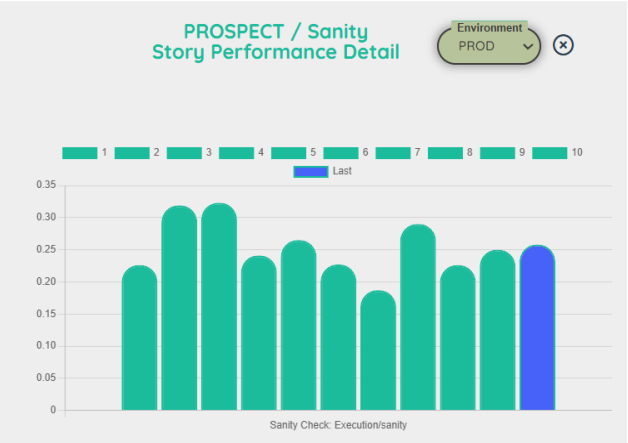
Measure the performances

Performance

Although this type of tool is not appropriate to measure the performance, it can be useful in the following case:

- To measure the impact of a new deployment on the performance.
- To measure the stability of the response time of the server.
- To evaluate the required time to perform a set of actions.





Documentation

Documentation of the tool is very important, so it is organized by profile:

Tester:

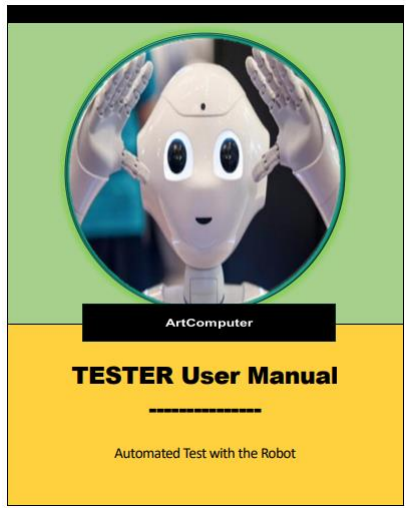
- Tester Manual

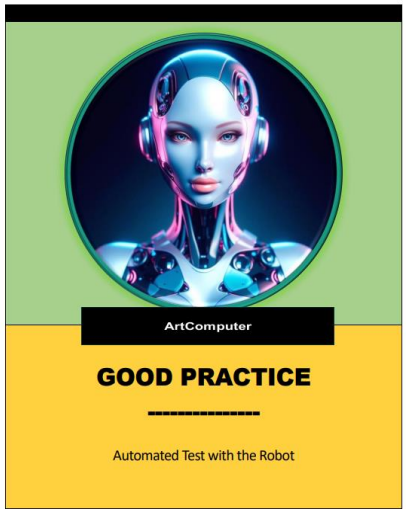
Designer:

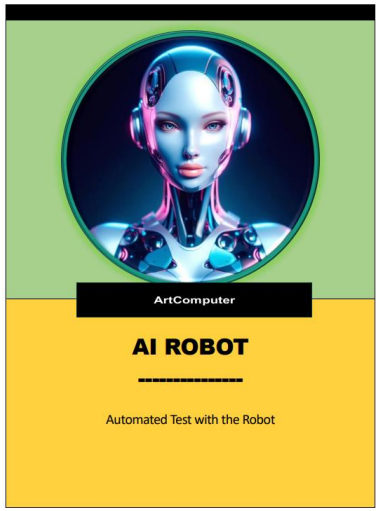
- Designer Manual
- Function Manual
- Good Practices
- Natural Language

Administrator

- Admin Manual
- AI Robot Manual







Automated Testing Tool

Long term review

Long term review after 5 years

The use of an automated testing tool must be shared by the **structure** because it's a **long-term investment**.

The success of the tool is closely related to the **quality of the application**.
If the application is not stable, poorly written (no Id, no Name, no specific class…), you cannot bring a correct added-value!

With the same idea, if your application is not **correctly documented** (use cases, business analysis…), you will fail!

It's important to create an **eco-system** around the application to improve the quality of the scenario.

It's important to **understand the business** behind the application (you cannot hire 3-5 people and ask them to handle all the projects in your organization).

The type of testing in the private sector is different from a public administration (non-regression versus data preparation)

The **overall time** to execute a big scenario with an automated testing tool is not very different from a manual test.
(E.g.: Create a call + Create Offer + Signature → Manual Test: 1 per day → Robot: 3-4 per day)
The big difference is the robot's lack of empathy which allows it to repeat the same test over and over again…

The detectGUI() function, to automatically detects an element, has a success rate of 80 to 90%.

Automated Testing Tool

Test Case 1

Test Case 1

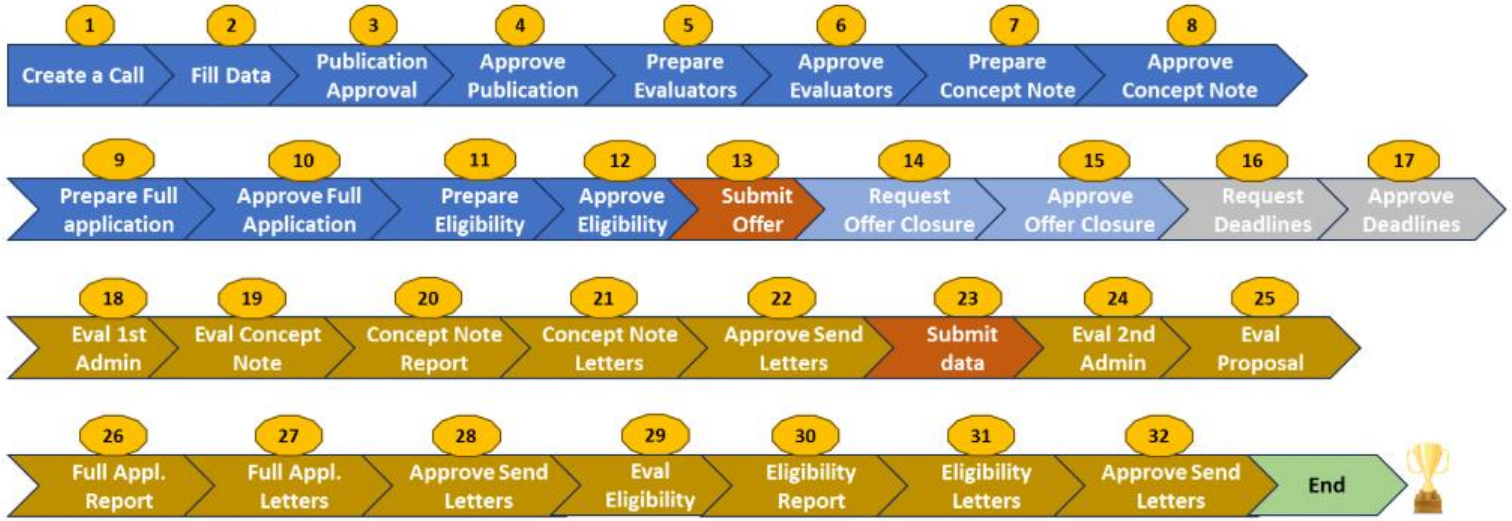
Prepare contracts: from the creation to the approval of the contractors.

It's possible to progress step by step or all in a row.
A batch mode is also available ready for a scheduler.

Direct Calls

STORY

WORKFLOW



Automated Testing Tool

Test Case 2

Test Case 2

Automatically feed a ‘warehouse’ on Confluence. Update the status on Confluence, based on the actual status in the application.

The purpose of the warehouse is to provide a service to Testers and Business to make records ready for further testing.

DIRECT

Responsible	Date	Env.	Nature	Procedure	Award	Mgt Mode	Lot	Workflow	Reference	Status	Comment
goffipl	04/12/2024	*ACC	ACT	Restricted	(FR2018) Restricted procedure	Direct Management - Headquarter	Without lots	23	152928	Under eval. (CN)	DIRECT Call
goffipl	02/12/2024	*ACC	ACT	Restricted	(FR2018) Restricted procedure	Direct Management - Headquarter	Without lots	23	152929	Under eval. (CN)	DIRECT Call 10 offer
goffipl	04/12/2024	*TEST	ACT	Restricted	(FR2018) Restricted procedure	Direct Management - Headquarter	Without lots	23	151546	Under eval. (CN)	DIRECT Call 10 offer
goffipl	02/12/2024	*TEST	ACT	Restricted	(FR2018) Restricted procedure	Direct Management - Headquarter	Without lots	23	151547	Under eval. (CN)	DIRECT Call 10 offer
goffipl	06/01/2025	TEST	ACT	Restricted	(FR2018) Restricted procedure	Direct Management - Headquarter	Without lots	20	151566	Under eval. (CN)	DIRECT Call for Testing
goffipl	10/01/2025	TEST	ACT	Restricted	(FR2018) Restricted procedure	Direct Management - Headquarter	Without lots	30	151567	Under eval. (CN)	DIRECT Call for Testing
goffipl	08/01/2025	TEST	ACT	Restricted	(FR2018) Restricted procedure	Direct Management - Headquarter	Without lots	21 Out of deadline	151568	Under eval. (CN)	DIRECT Call for Testing
goffipl	13/01/2025	TEST	ACT	Open	(FR2018) Restricted procedure	Direct Management - Headquarter	Without lots	20	151579	Under eval. (CN)	DIRECT Call for Testing
goffipl	14/01/2025	TEST	ACT	Open	(FR2018) Restricted procedure	Direct Management - Headquarter	Without lots	25	151585	Under eval. (CN)	DIRECT Call for Testing
Filip	16/01/2025	DEV	ACT	Restricted	(FR2018) Restricted procedure	Direct Management - Headquarter	Without lots	31	151010	Under eval. (EL)	DIRECT Call for Testing (1 offer)

Automated Testing Tool

Test Case 3

Test Case 3

Sanity check scheduled at 6:00 AM.
Check the availability of the different environments with a notification send to the group of people:

- Only for specific environment(s)
- Only in case of errors
- Only in case of a new version
- ….