



R5.07 : Automatisation de la chaîne de production





R5.07 : Objectifs

Concrétiser la chaîne de production d'une application afin de développer les aptitudes nécessaires pour travailler efficacement dans une équipe informatique.





R5.07 : Objectifs



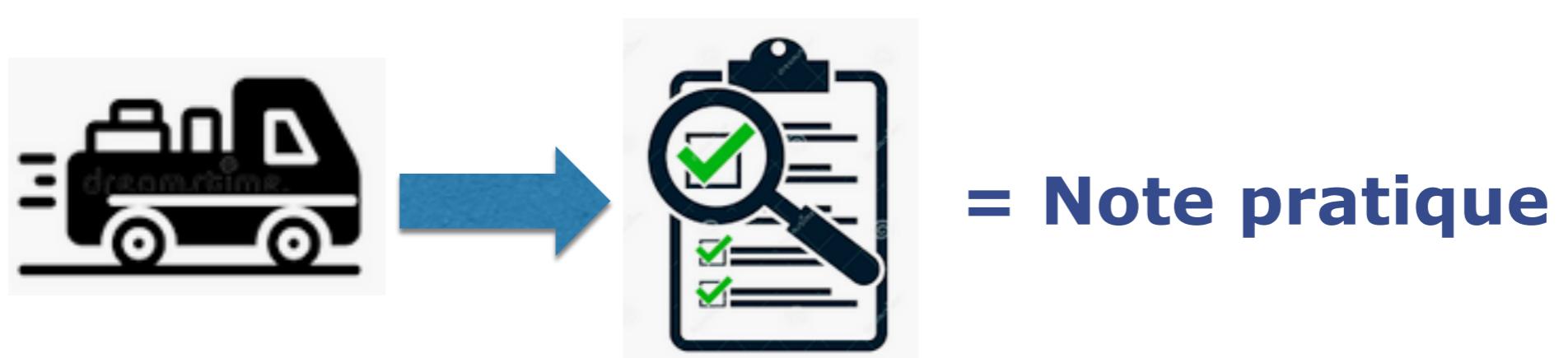
- **Partie 1 : Automatisation des tests (12hTDm)**
 - **Cours 1 : Robot-framework (6h TDm)**
 - **Cours 2 : RPA (6h TDm)**
- Partie 2 : Intégration continue (7hTDm) (Monsieur Himdi)
- Partie 3 : Utilisation de conteneurs (??TDm) (???)
 - Suite de la virtualisation





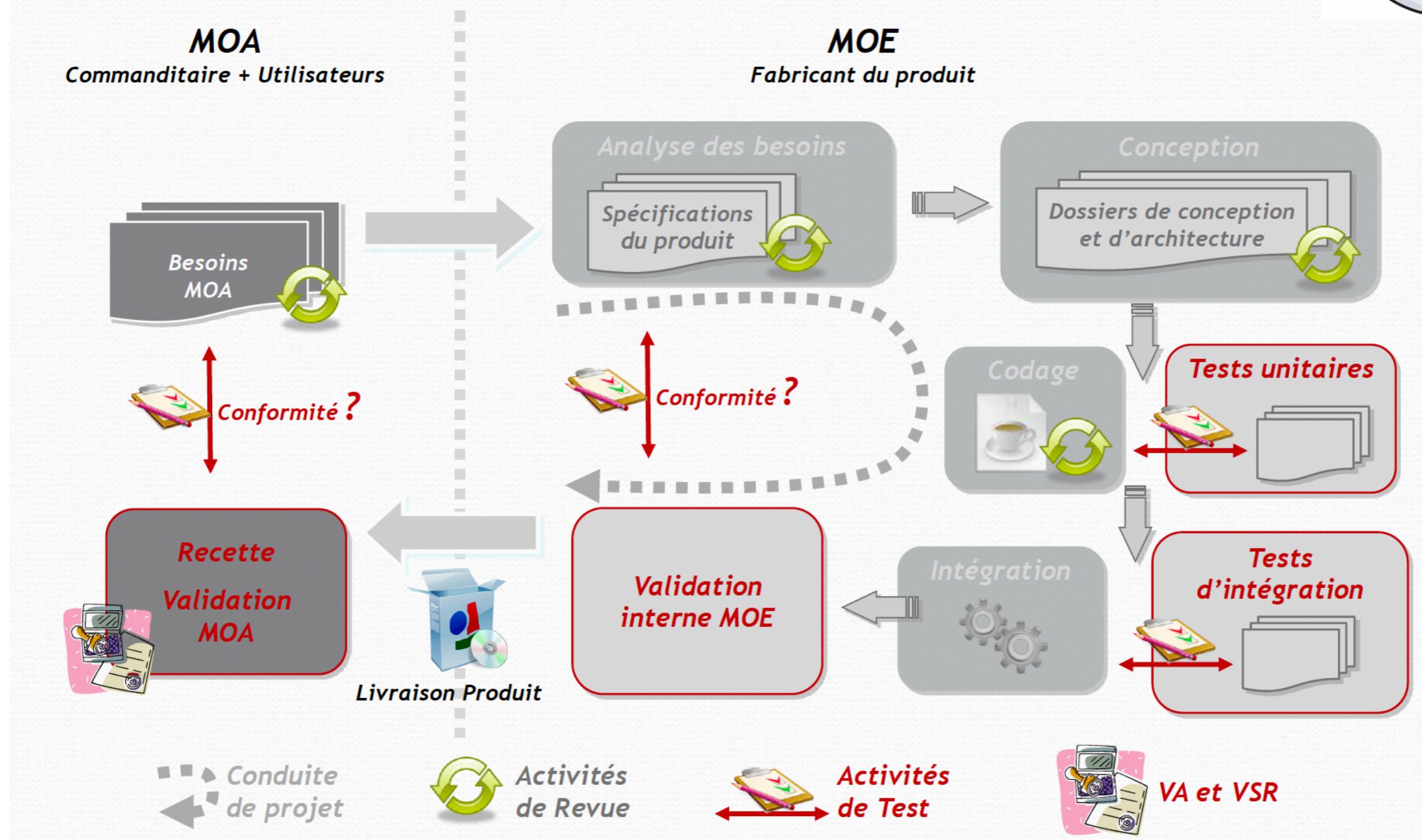
Évaluation/Livrables

- Scriptes Projets Test (TP1 & TP2)
- UMTICE





Automatisation d'une chaîne de production (Rappel)



Cours LPTQL M. Himdi (2018)



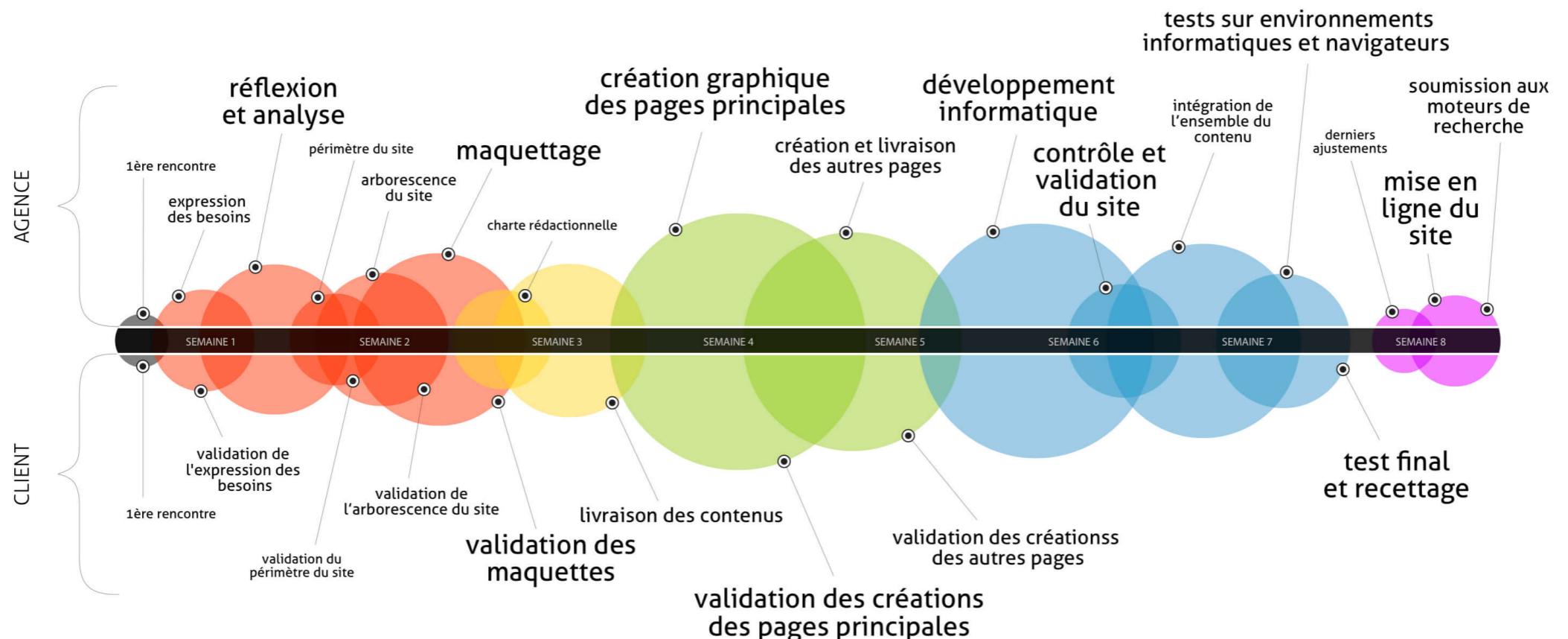


Automatisation d'une chaîne de production (Exemple)



La Chaîne De Production d'un Site

JALONS, IMPLICATION, IMPORTANCE ET DÉLAI



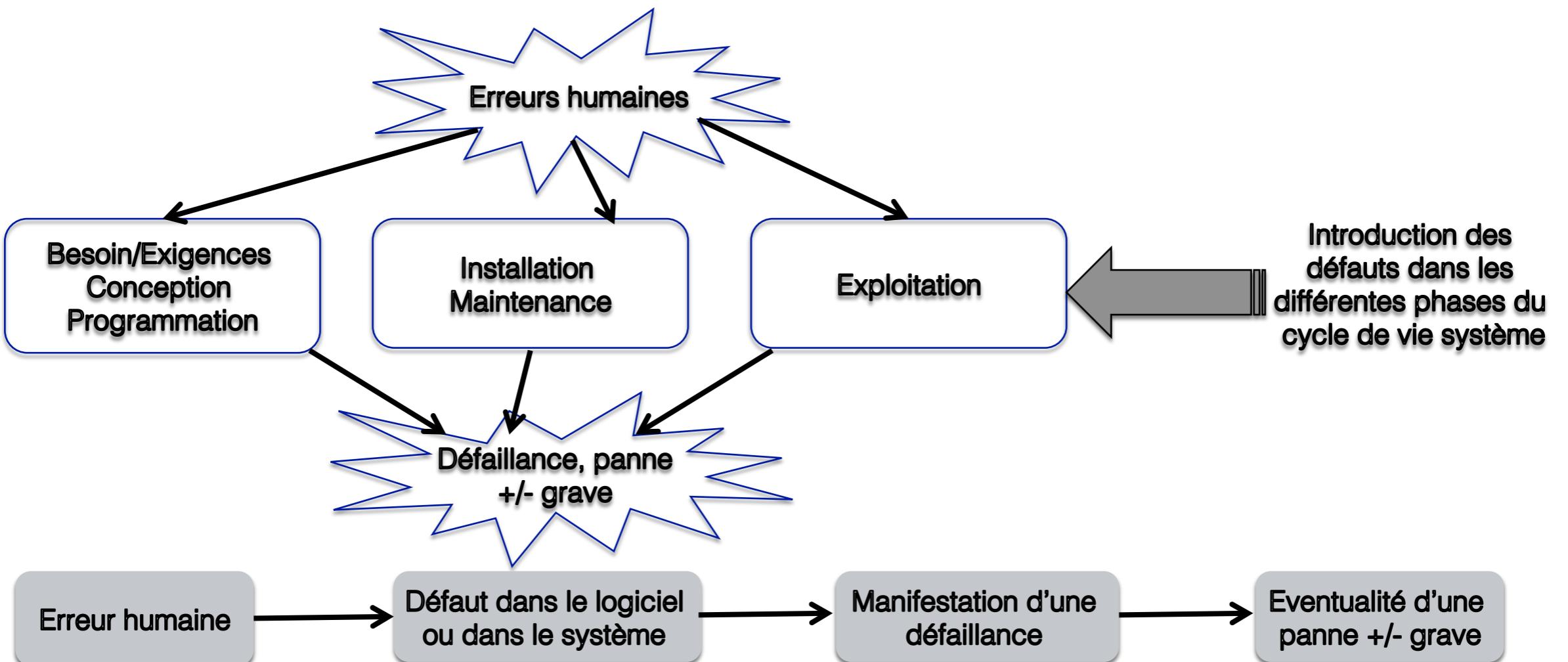
La création graphique "La Chaîne De Production d'un Site" a été réalisée par John Furness de Simple Square. © 2011 - Tous droits réservés.
Traduit en français par Jean-Philippe Wozniak@www.advisia-conseil.com





Chaine de production

La chaîne d'erreur



Source : *Pratique des tests logiciels - 4e éd. - Améliorer la qualité par les tests. Gérer une campagne de tests.*
Jean-François Pradat-Peyre , Jacques Printz





Tests logiciels : introduction

Pourquoi Tester/Vérifier & Valider ?

Pour éviter les bugs

- Ariane V-vol 501, 1996 : problème de conversion de nombres
→ Coût : 370 millions de dollars
- Année 2000
- Bug d'Amazon casse les prix sur le Net, 18 juin 2009
- « Au 1^o janvier 2010, nombre de logiciels, aussi bien grand que professionnels, sont tombés en panne. En Allemagne, ce sont des millions de cartes bancaires qui ne fonctionnent pas »
→ O1 INFORMATIQUE du 21/02/2010
- Un bug informatique double les aides agricoles (Un pactole surprise de 4,2 milliards d'euros...)
→ Ouest France, jeudi 05 décembre 2013





Tests logiciels : introduction



- Motivations
 - Coût des bugs
 - Coûts économiques : des milliards \$/an
 - Coûts humains, environnementaux, etc.
 - Nécessité d'assurer la qualité des logiciels
 - domaines critiques : atteindre une très haute qualité imposée par les lois/normes/assurances/...
 - autres domaines : atteindre le rapport qualité/prix jugé optimal (attentes du client)
 - Méthodes de Validation et Vérification (V & V)
 - Vérification : est-ce que le logiciel fonctionne correctement ?
 - Validation : est-ce que le logiciel fait ce que le client veut ?





Tests logiciels : introduction

Tester c'est...

*« Le test est **l'exécution** ou **l'évaluation** d'un système ou d'un composant, par des moyens **automatiques** ou **manuels**, pour vérifier qu'il **répond à ses spécifications** ou **identifier les différences** entre les résultats attendus et les résultats obtenus. » Norme IEEE*





Automatisation : introduction

L'automatisation des tests logiciels

- Processus qui permet de créer et d'exécuter des programmes informatiques (scripts)
➔ valider automatiquement un logiciel.
- Permet de s'assurer que le logiciel répond aux *normes de qualité*
➔ Style de code, Fonctionnalités & Exigences client.
- Basée sur des outils spécifiques qui simulent les interactions humaines avec SUT (System under test)
➔ effectue des vérifications et génèrent des rapports de résultats.
- Accroître *l'efficacité* et la *fiabilité* des tests logiciels
➔ réduire les efforts et les coûts associés aux tests manuels.





Automatisation : Objectifs

Avantages

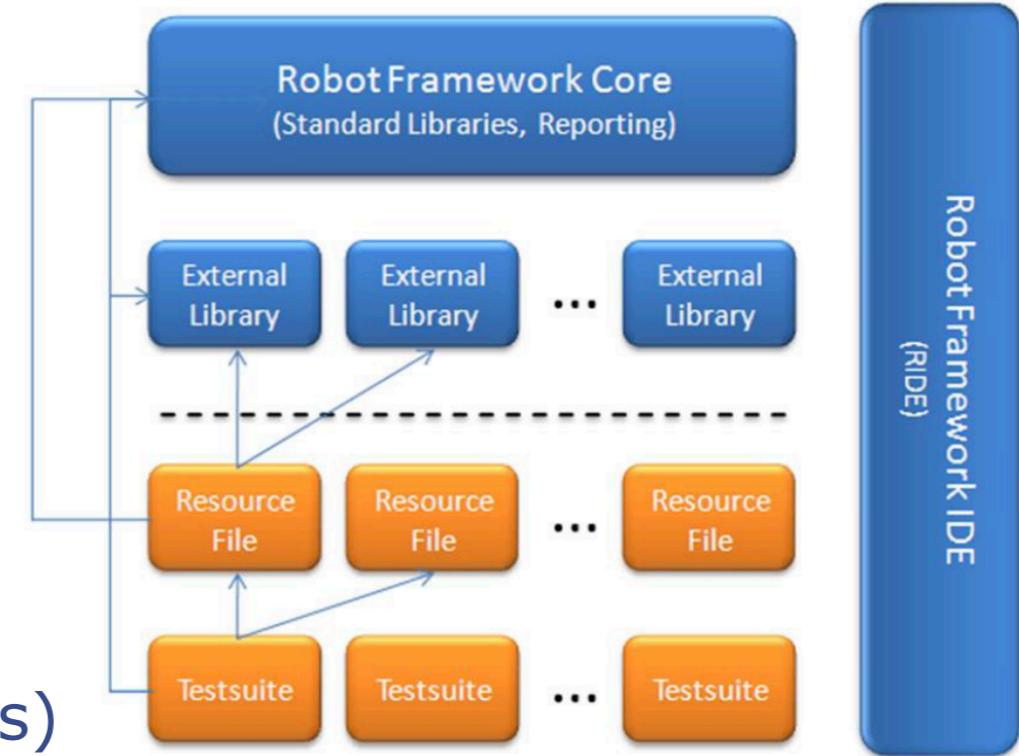
- Automatisation des tâches redondantes et pénibles
- Amélioration de la couverture des tests
- Fiabilisation des changements lors des développements itératifs
 - ➔ un socle de tests de non-régression
- Gain de temps et d'efforts
 - ➔ Réduction des coûts
- Fournit un retour au plutôt sur la qualité des produits
 - ➔Améliorer la confiance dans le produit
 - ✓ Tests systématiques et réguliers
 - ✓ Regard objectif sur le fonctionnement du produit
 - ✓ Mesurer la qualité dans le temps (entre livraisons)





RobotFramework : Introduction

- Solution d'automatisation des tests
- Framework Open source → robotframework.org
- Langage base sur les mots clés
 - Écriture facile, lisible et compréhensible
 - Utilisable par des non-développeurs
- Rapports et Logs lisibles
 - Investigation rapide
 - Exploitables (HTML, XML ..)
- Multi plateforme: Windows, Linux
- Multi technologie: Python, Java
- Nombreux outils associés (Squash, Jenkins)

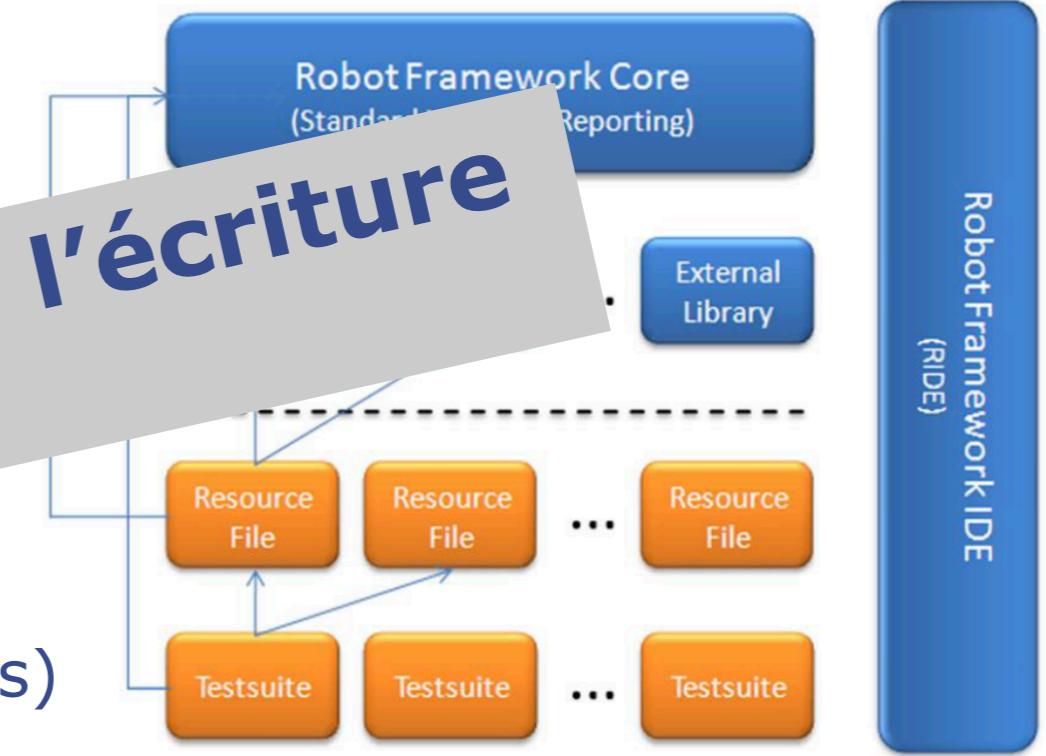




RobotFramework : Introduction

- Solution d'automatisation des tests
- Un framework Open source → robotframework.org
- Langage base sur les mots clés
 - Écriture facile, lisible et compréhensible
 - Utilisable par des non-développeurs
- Rapports et Logs lisibles
 - Investigation rapide
 - Exploitables (HTTP, API)
- Multilingue (English, French, German, Spanish, etc.)
- Multiplateforme (Python, Java, .NET, Ruby, etc.)
- Outils associés (Squash, Jenkins)

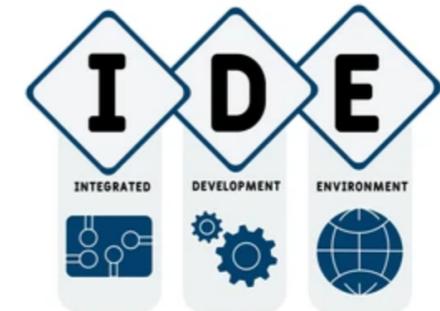
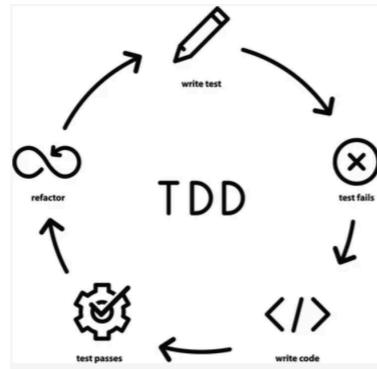
→ → Efforts concentrés sur l'écriture des tests





RobotFramework : Introduction

- Écriture des tests
 - Approche dirigée par les mots clés
 - Développement Dirigé par les Tests / Test Driven Development (TDD)
 - Test piloté par les données / Data-Driven Testing (DDT)
 - Nombreux plug-in pour IDE





RobotFramework : Installation

- Installation de Python
 - Vérifier si “Pip” est installé
 - Ligne commande (CMD) :
 - ✓ “py --version”
 - ✓ “py -m pip --version”
- pip3 install robotframework
- pip3 install robotframework-selenium2library
- pip3 install --upgrade robotframework-selenium2library
- Pip show robotframework → infos sur la version installée
- Pip list → liste des librairies installées

Download the latest version for macOS

[Download Python 3.11.3](#)

Looking for Python with a different OS? Python for [Windows](#), [Linux/UNIX](#), [macOS](#), [Other](#)





RobotFramework : Structure d'un test

*** Settings ***

Ici on inclut les librairies requises

Documentation exemple de scénario de test

*** Variables ***

Ici on déclare les variables au besoin

*** Test Cases ***

Ici on va écrire les scripts de tests / scénarios de tests

*** Keyword ***

Ici on déclare les mots clés





RobotFramework : Exemple 1



```
1 *** Settings ***
2 Documentation      TP BUT3 (2023-2024) simple exemple utilisant SeleniumLibrary.
3 Library            SeleniumLibrary
4
5 Load in Interactive Console
6 *** Variables ***
7 ${LOGIN URL}       https://google.fr
8 ${BROWSER}          ff
9
10 *** Test Cases ***
11 Run | Debug | Run in Interactive Console
12 → Test une connexion Valide à google
13 → Open Browser    ${LOGIN URL}    ${BROWSER}
```

1

```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL ROBOT DOCUMENTATION ROBOT OUTPUT
● PS C:\VScode\ProjetsLahcen> robot c:\VScode\ProjetsLahcen\testOpenNavigateur.robot
```

2

```
PROBLEMS 2 OUTPUT DEBUG CONSOLE TERMINAL ROBOT DOCUMENTATION ROBOT OUTPUT
● -----
testOpenNavigateur :: TP BUT3 (2023-2024) simple exemple utilisant Selenium...
-----
Test une connexion Valide à google | PASS |
-----
testOpenNavigateur :: TP BUT3 (2023-2024) simple exemple utilisant... | PASS |
1 test, 1 passed, 0 failed
-----
Output:  C:\VScode\ProjetsLahcen\output.xml
Log:     C:\VScode\ProjetsLahcen\log.html
Report:  C:\VScode\ProjetsLahcen\report.html
● PS C:\VScode\ProjetsLahcen> 
```

Mots clés

3





RobotFramework : Exemple 1



testOpenNavigateur Report

Generated
20230510 11:59:04 UTC+02:00
1 minute 42 seconds ago

Summary Information

Status:	All tests passed
Documentation:	TP BUT3 (2023-2024) simple exemple utilisant SeleniumLibrary.
Start Time:	20230510 11:58:54.591
End Time:	20230510 11:59:04.368
Elapsed Time:	00:00:09.777
Log File:	log.html

Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	1	1	0	0	00:00:08	<div style="width: 100%; background-color: #6aa84f; height: 10px;"></div>
Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
No Tags						<div style="width: 0%; background-color: #e0e0e0; height: 10px;"></div>
Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
testOpenNavigateur	1	1	0	0	00:00:10	<div style="width: 100%; background-color: #6aa84f; height: 10px;"></div>





RobotFramework : Exemple 1



testOpenNavigateur Log

Generated
20230510 11:59:04 UTC+02:00
2 minutes 34 seconds ago

Test Statistics

Total Statistics	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
All Tests	1	1	0	0	00:00:08	<div style="width: 100%; background-color: #2e7131;"></div>
Statistics by Tag	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
No Tags						<div style="width: 0%; background-color: #d3d3d3;"></div>
Statistics by Suite	Total	Pass	Fail	Skip	Elapsed	Pass / Fail / Skip
testOpenNavigateur	1	1	0	0	00:00:10	<div style="width: 100%; background-color: #2e7131;"></div>

Test Execution Log

- SUITE	testOpenNavigateur
Full Name:	testOpenNavigateur
Documentation:	TP BUT3 (2023-2024) simple exemple utilisant SeleniumLibrary.
Source:	c:\VScode\ProjetsLahcen\testOpenNavigateur.robot
Start / End / Elapsed:	20230510 11:58:54.591 / 20230510 11:59:04.368 / 00:00:09.777
Status:	1 test total, 1 passed, 0 failed, 0 skipped
- TEST	Test une connexion Valide à google
Full Name:	testOpenNavigateur.Test une connexion Valide à google
Start / End / Elapsed:	20230510 11:58:55.948 / 20230510 11:59:04.366 / 00:00:08.418
Status:	PASS
+ KEYWORD	SeleniumLibrary .Open Browser \${LOGIN URL}, \${BROWSER}





RobotFramework : Librairies Standards

GETTING STARTED | RESOURCES | COMMUNITY | DEVELOPMENT | DOCS ▾

LIBRARIES BUILT-IN TOOLS

Libraries and tools that are bundled with the framework. Libraries provide the actual automation and testing capabilities to Robot Framework by providing keywords.

Filter by tag

Name	Description	Tags
Builtin	Provides a set of often needed generic keywords. Always automatically available without imports.	LIBRARY
Collections	Provides a set of keywords for handling Python lists and dictionaries.	LIBRARY
DateTime	Library for date and time conversions.	LIBRARY
Dialogs	Provides means for pausing the execution and getting input from users.	LIBRARY
Libdoc	Generate keyword documentation for test libraries and resource files.	TOOL
OperatingSystem	Enables various operating system related tasks to be performed in the system where Robot Framework is running.	LIBRARY
Process	Library for running processes in the system.	LIBRARY

SHOW MORE

→ <https://robotframework.org/#test-libraries>





RobotFramework : Librairies externes

Automatisation des tests avec plusieurs technologies

- SeleniumLibrary : Application WEB
- AppiumLibrary : Application Mobile
- AutoItLibrary : Application Windows
- DatabaseLibrary : Base de données

GETTING STARTED | RESOURCES | COMMUNITY | DEVELOPMENT | DOCS ▾

LIBRARIES BUILT-IN TOOLS

Separately developed external libraries that can be installed based on your needs. Creating your own libraries is a breeze. For instructions, see [creating test libraries](#) in Robot Framework User Guide.

Filter by tag

Name	Description	Stars	Tags
SeleniumLibrary	Web testing library that uses popular Selenium tool internally.	1230	WEB, SELENIUM
RPA framework	Collection of open-source libraries and tools for Robotic Process Automation (RPA), designed to be used both with Robot Framework and Python.	787	RPA
HTTP	HTTP level testing using Python Requests internally.	443	HTTP
RequestsLibrary (Python)			
Browser Library	A modern web testing library powered by Playwright. Aiming for speed, reliability and visibility.	368	WEB
AppiumLibrary	Android and iOS testing. Uses Appium internally.	341	MOBILE
RESTinstance	Test library for HTTP JSON APIs.	187	HTTP
Database Library (Python)	Python based library for database testing. Works with any Python interpreter, including Jython.	134	DB

→ <https://robotframework.org/#test-libraries>





RobotFramework : Mots clés

Exemple : Selenium Keyword

- Open Browser <url> <browser>
- Maximize Browser Window
- Input Text <locator> <value>
- Click Element <locator>
- Click Link <locator>
- Click Button <locator>
- Close Browser

→ <https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html>





RobotFramework : Mots clés

Exemple : Selenium Keyword

- Wait Until Page Contains <text>
- Page Should Contain <text>
- Title Should Be <text>
- Mouse Down <locator>
- Element Should Contain

→ **<https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html>**





RobotFramework : Mots clés

SeleniumLibrary

Search

Keywords (177)

- Add Cookie · Add Location Strategy · Alert Should Be Present ·
- Alert Should Not Be Present · Assign Id To Element ·
- Capture Element Screenshot · Capture Page Screenshot ·
- Checkbox Should Be Selected · Checkbox Should Not Be Selected ·
- Choose File · Clear Element Text · Click Button · Click Element ·
- Click Element At Coordinates · Click Image · Click Link ·
- Close All Browsers · Close Browser · Close Window · Cover Element ·
- Create Webdriver · Current Frame Should Contain ·
- Current Frame Should Not Contain · Delete All Cookies · Delete Cookie ·
- Double Click Element · Drag And Drop · Drag And Drop By Offset ·
- Element Attribute Value Should Be · Element Should Be Disabled ·
- Element Should Be Enabled · Element Should Be Focused ·
- Element Should Be Visible · Element Should Contain ·
- Element Should Not Be Visible · Element Should Not Contain ·
- Element Text Should Be · Element Text Should Not Be ·
- Execute Async Javascript · Execute Javascript · Frame Should Contain ·
- Get Action Chain Delay · Get All Links · Get Browser Aliases ·
- Get Browser Ids · Get Cookie · Get Cookies · Get Element Attribute ·
- Get Element Count · Get Element Size · Get Horizontal Position ·
- Get List Items · Get Location · Get Locations · Get Selected List Label ·
- Get Selected List Labels · Get Selected List Value ·
- Get Selected List Values · Get Selenium Implicit Wait ·
- Get Selenium Page Load Timeout · Get Selenium Speed ·
- Get Selenium Timeout · Get Session Id · Get Source · Get Table Cell ·
- Get Text · Get Title · Get Value · Get Vertical Position ·
- Get WebElement · Get WebElements · Get Window Handles ·
- Get Window Identifiers · Get Window Names · Get Window Position ·

- Set Selenium Speed · Set Selenium Timeout · Set Window Position ·
- Set Window Size · Simulate Event · Submit Form · Switch Browser ·
- Switch Window · Table Cell Should Contain ·
- Table Column Should Contain · Table Footer Should Contain ·
- Table Header Should Contain · Table Row Should Contain ·
- Table Should Contain · Textarea Should Contain ·
- Textarea Value Should Be · Textfield Should Contain ·
- Textfield Value Should Be · Title Should Be · Unselect All From List ·
- Unselect Checkbox · Unselect Frame · Unselect From List By Index ·
- Unselect From List By Label · Unselect From List By Value ·
- Wait For Condition · Wait Until Element Contains ·
- Wait Until Element Does Not Contain · Wait Until Element Is Enabled ·
- Wait Until Element Is Not Visible · Wait Until Element Is Visible ·
- Wait Until Location Contains · Wait Until Location Does Not Contain ·
- Wait Until Location Is · Wait Until Location Is Not ·
- Wait Until Page Contains · Wait Until Page Contains Element ·
- Wait Until Page Does Not Contain ·
- Wait Until Page Does Not Contain Element ·

<https://robotframework.org/SeleniumLibrary/SeleniumLibrary.html>





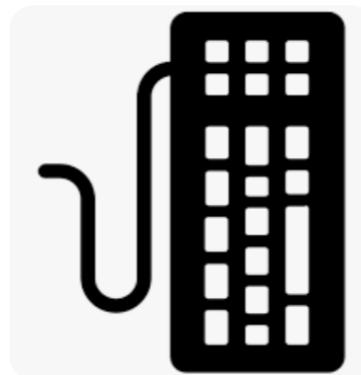
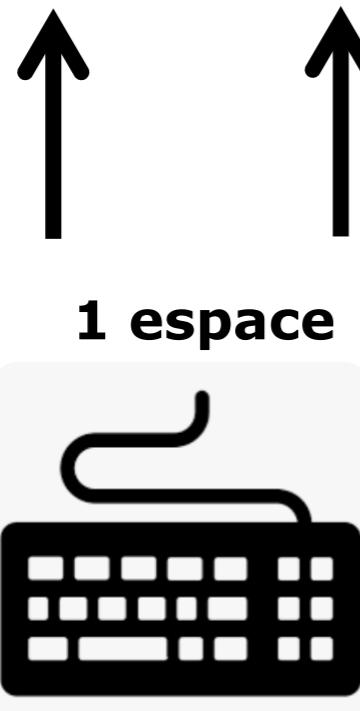
RobotFramework : Mots clés

Attention à ...

Page Should Contain BUT3

4 espaces

↓





RobotFramework : Mots clés



*** Settings ***

Documentation Simple exemple d'utilisation mots-clés SeleniumLibrary.
Library SeleniumLibrary

*** Variables ***

\${LOGIN URL} http://localhost:8080
 \${BROWSER} Chrome

*** Test Cases ***

Valid Login

Open Browser To Login Page

Input Username BUT3

Input Password 2023!/?

Submit IDs

Welcome Page Should Be Open

Nom du scénario

Mot clé 1

Mot clé 3

*** Keywords ***

Open Browser To Login Page

Open Browser \${LOGIN URL} \${BROWSER}

Title Should Be Login Page

Input Username

[Arguments] \${username}

Input Text username_field \${username}

Input Password

[Arguments] \${password}

Input Text password_field \${password}

Submit IDs

Click Button login_button

Welcome Page Should Be Open

Title Should Be Welcome Page





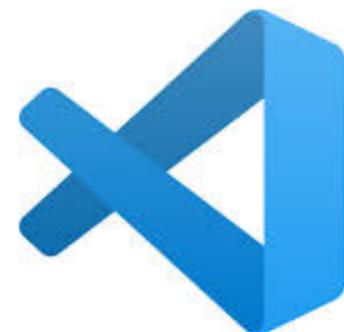
RobotFramework : éditeurs de scripts



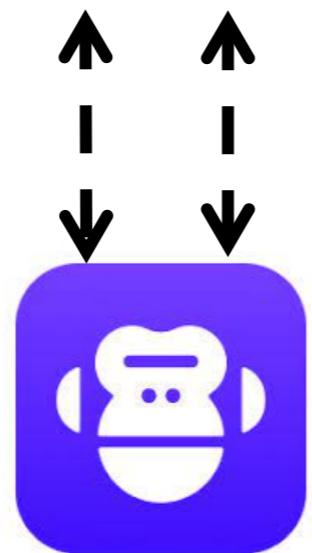
- Éditeurs de scripts pour Robot Framework
- Fonctionnalités d'auto-complétion et mise en forme
- Gestion des Keywords
- Documentation des librairies



PyCharm



VS Code



RoboCorp



Ride



Eclipse





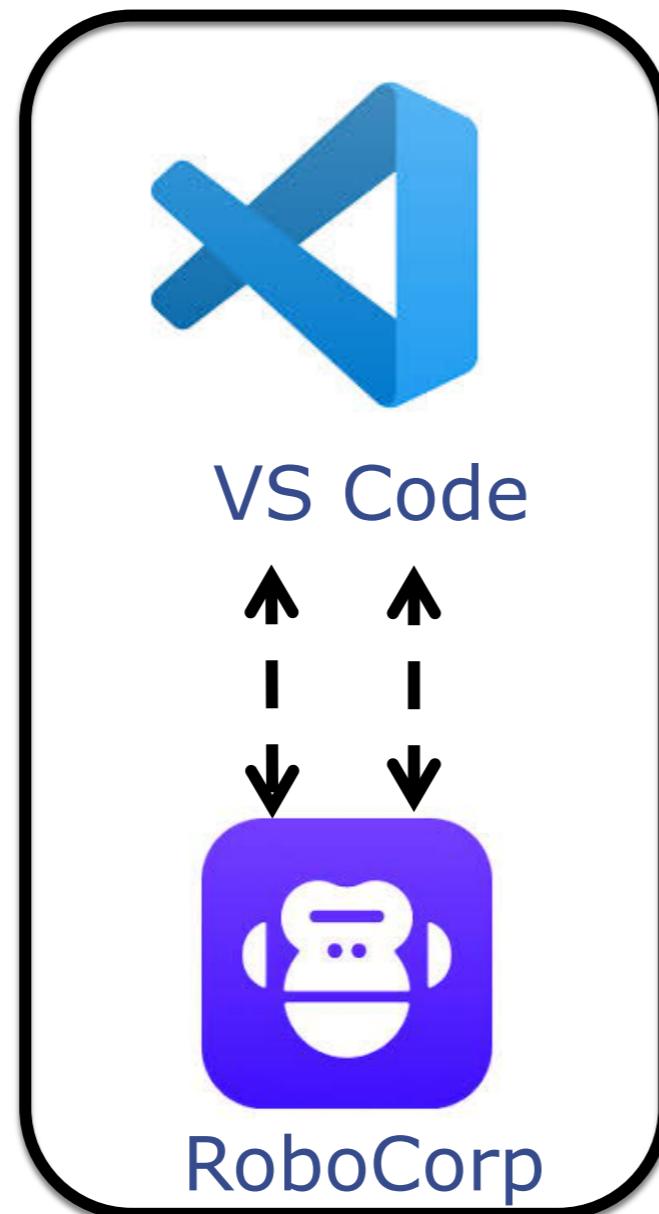
RobotFramework : éditeurs de scripts



- Éditeurs de scripts pour Robot Framework
- Fonctionnalités d'auto-complétion et mise en forme
- Gestion des Keywords
- Documentation des librairies



PyCharm



Ride



Eclipse





RobotFramework : Structure d'un projet



The screenshot shows the VS Code interface with the Robot Framework project structure on the left and a code editor and terminal on the right.

Project Explorer:

- PROJETROBOTFRAMEWORKBUT3
 - Resources
 - Import.robot
 - Resources.robot
 - Results
 - log.html
 - output.xml
 - report.html
 - ScenariosTest
 - Test1.robot
 - Test2.robot
 - Test3.robot
 - Test4.robot
 - testOpenNavigateur.robot

Code Editor (testOpenNavigateur.robot):

```
*** Settings ***
Documentation    TP BUT3 (2023-2024) simple exemple
...      utilisant SeleniumLibrary.
Library      SeleniumLibrary

Load in Interactive Console
*** Variables ***
${LOGIN URL}    https://google.fr
${BROWSER}      ff

*** Test Cases ***
Run | Debug | Run in Interactive Console
Test une connexion Valide à google
  Open Browser    ${LOGIN URL}    ${BROWSER}
```

Terminal Output:

```
Test une connexion Valide à google | PASS |
testOpenNavigateur :: TP BUT3 (2023-2024) simple exemple utilisant... | PASS |
1 test, 1 passed, 0 failed
=====
Output:  C:\VScode\ProjetRobotFrameworkBUT3\output.xml
Log:    C:\VScode\ProjetRobotFrameworkBUT3\log.html
Report: C:\VScode\ProjetRobotFrameworkBUT3\report.html
PS C:\VScode\ProjetRobotFrameworkBUT3>
```

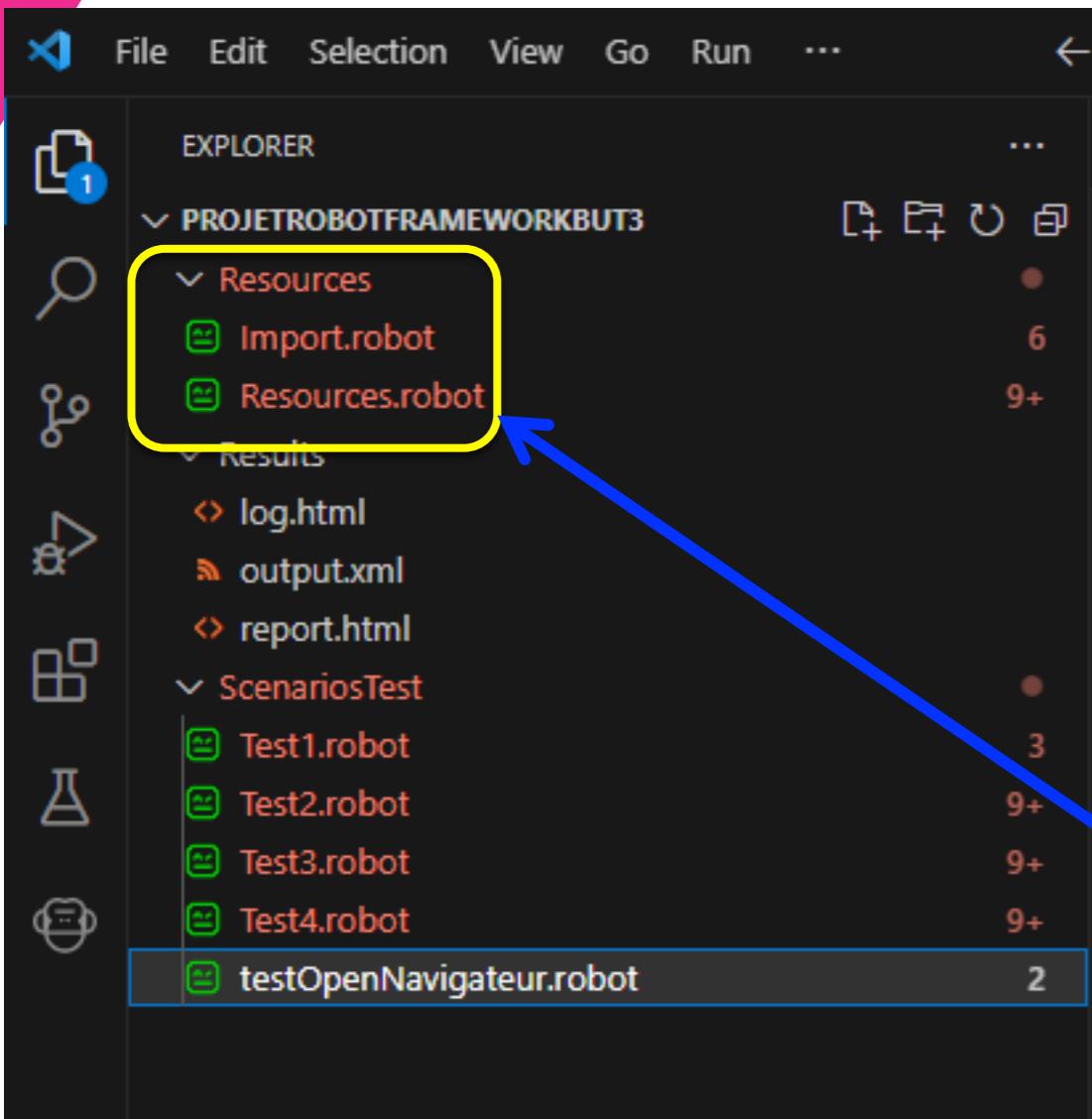


Améliorer la réutilisabilité et la maintenabilité





RobotFramework : Structure d'un projet



Centraliser les mots clés et les variables

```
*** Settings ***
Documentation Simple exemple d'utilisation
... mots-clés SeleniumLibrary.
Library SeleniumLibrary

*** Variables ***
${LOGIN URL} http://localhost:8080
${BROWSER} Chrome

*** Keywords ***
Open Browser To Login Page
Open Browser ${LOGIN URL} ${BROWSER}
Title Should Be Login Page

Input Username
[Arguments] ${username}
Input Text username_field ${username}

Input Password
[Arguments] ${password}
Input Text password_field ${password}

Submit IDs
Click Button login_button

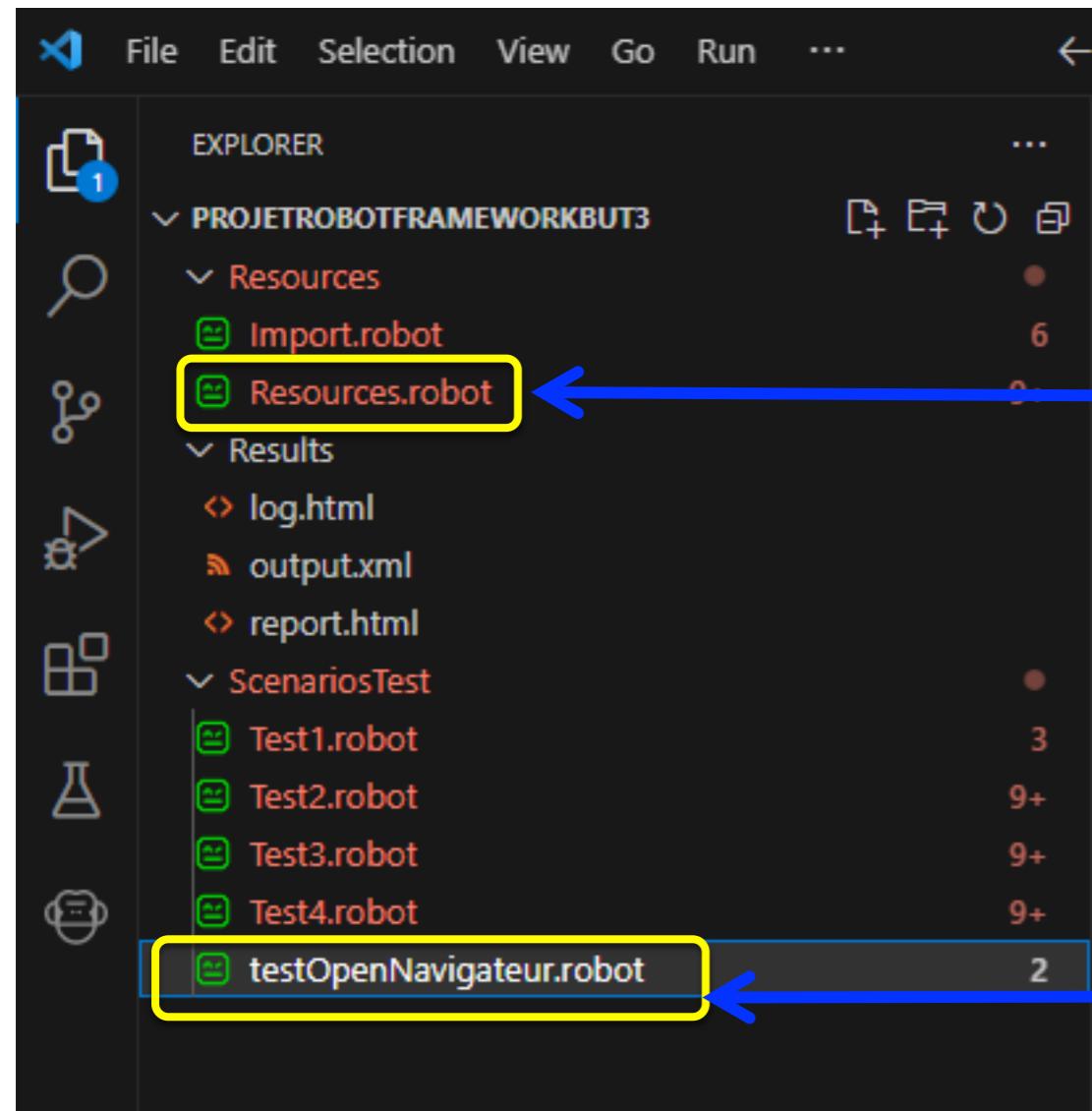
Welcome Page Should Be Open
Title Should Be Welcome Page
```

Ressources.robot





RobotFramework : Structure d'un projet



```
Importer le fichier ressources.robot

*** Settings ***
Documentation Simple exemple d'utilisation
mots-clés SeleniumLibrary.

Ressources .../Ressources/Resources.robot

Test Teardown Close Browser

*** Test Cases ***
Valid Login
  Open Browser To Login Page
  Input Username BUT3
  Input Password 2023!?
  Submit IDs
  Welcome Page Should Be Open

TestOpenNavigateur.robot
```

Améliorer la réutilisabilité et la maintenabilité





RobotFramework : exécution des scénarios de test



- Test Setup
- Test Teardown
- Suite Setup
- Suite Teardown

Début de chaque cas de test
Fin de chaque cas de test
Début d'une suite de tests
Fin d'une suite de tests





RobotFramework : exécution des scénarios de test



Intégration Continue & Jenkins

Build

Exécuter une ligne de commande batch Windows

Commande

```
robot --outputdir tests/*test.robot
```

Voir [la liste des variables d'environnement disponibles](#)

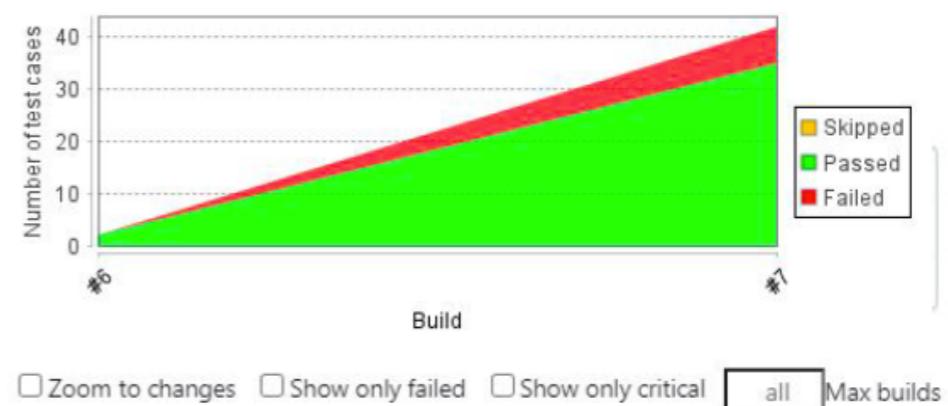
Ajouter une étape au build ▾

Actions à la suite du build

Publish Robot Framework test results

Directory of Robot output

Path to directory containing robot xml and html files (relative to build workspace)



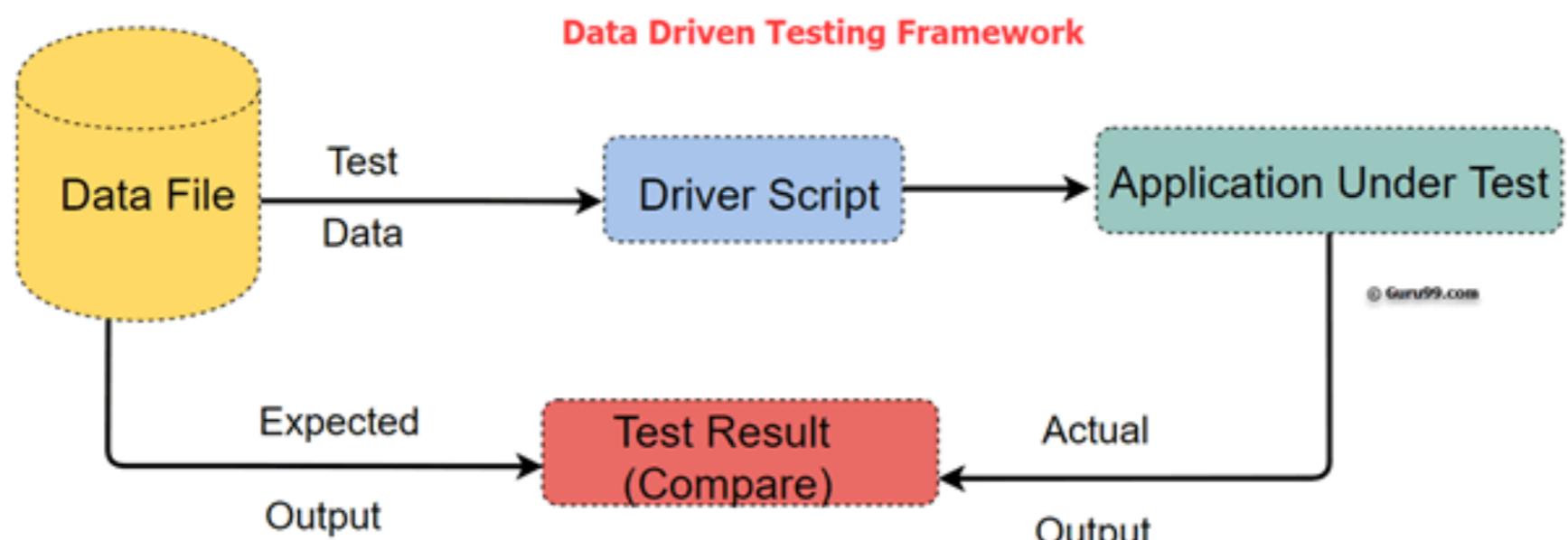
Partie 2 (Monsieur Himdi)





RobotFramework : Data-Driven Testing

- Qu'est-ce que le Data-Driven Testing (DDT) ?
 - AUSSI : test piloté par les données (ou DDT)
 - Méthodologie de test de logiciels
 - Ensemble de conditions pour alimenter
 - ✓ Les données d'entrée du test,
 - ✓ Les données de sortie attendues
 - ✓ Les variables d'environnement et de contrôle.



Source d'image : Guru99.com





RobotFramework : Data-Driven Testing

- Data-Driven Testing (DDT) : avantages



- ✓ Automatisation des tests avec différentes données d'entrée
- ✓ Lecture des données (builtin, fichiers externes (CSV, Excel), base de données)
- ✓ Meilleure couverture des tests
- ✓ Meilleure maintenabilité des scripts de tests
 - ✧ Optimisation de nombre de scripts de test : exécution d'un script avec plusieurs jeux de données
 - ✧ Simplification de la mise à jour des scripts de test
 - ✧ Plus de flexibilité : tests positifs et tests négatifs avec le même script (MAJ des données)





RobotFramework : Data-Driven Testing

- Data-Driven Testing (DDT) : avantages



- ✓ Automatisation des tests avec différentes données d'entrée
- ✓ Lecture des données (builtin, fichiers externes (CSV, Excel), base de données)
- ✓ Meilleure couverture des tests
- ✓ Meilleure maintenabilité
 - ✧ Optimisation : exécution d'un seul test : exécution d'un seul scénario avec mise à jour des scripts de test
 - ✓ Flexibilité : tests positifs et tests négatifs avec le même script (MAJ des données)

Approche pour concevoir les données de test
→ → pour concevoir les données de test





RobotFramework : Data-Driven Testing



- Data-Driven Testing (DDT) : Défis /challenges



- ✓ Une planification importante
 - L'automatisation des tests doit être bien planifiée en termes de scripts de test, de données de test et de leurs interactions.
- ✓ Complexité de la mise en place : un démarrage qui prend plus de temps
- ✓ Gestion et maintenance des jeux de données de test
 - Jeux de données pertinents, complets, et représentatifs des différents scénarios de test
 - Mise en place des mécanismes de validation et de nettoyage des données.
 - Gestion des versions des jeux de données, résolution des conflits, et validité et la représentativité au fil du temps.
- ✓ Gestion des dépendances : jeux de données et scripts
- ✓ Évolutivité
 - prévoir une architecture et une structure de données évolutives pour répondre aux futurs besoins du projet de test.





RobotFramework : Data-Driven Testing



- Data-Driven Testing (DDT) : exemple

- ✓ Test la connexion à ent de l'université → <https://ent.univ-lemans.fr/fr/index.html>
- ✓ Cas 1 : log et mdp valides
- ✓ Cas 2 : log valide et mdp non valide
- ✓ Cas 3 : log non valide et mdp valide
- ✓ Cas 4 : log vide et mdp vide
- ✓ Cas 5 : log valide et mdp vide
- ✓ Cas 6 : log vide et mdp valide



IDENTIFIANT

MOT DE PASSE

SE CONNECTER

[Mot de passe oublié](#) · [1ère connexion](#)

JE SUIS :

ÉTUDIANT

PERSONNEL





RobotFramework : Data-Driven Testing



Data-Driven Testing (DDT) : exemple /script1

```
1  *** Settings ***
2      Documentation      TP BUT3-INFO (IUT-LAVAL)  exemple 1 Test DDT .
3      Library            Selenium2Library
4  *** Variables ***
5      ${LOGIN URL}       https://ent.univ-lemans.fr/fr/index.html
6      ${BROWSER}          ff
7      ${link-personnel}   xpath://html/body/main/div/div/div/ul/li[2]/a
8      ${link-personnel}   xpath://*[contains(text(), "personnel")]
9      ${txtbox-username}  //*[@id="username"]
10     ${txtbox-password}  //*[@id="password"]
11     ${btn-seconnecter}  xpath://html/body/section/div/div/form/div[3]/div/button
12     ${btn-seconnecter}  //*[contains(text(), "SE CONNECTER")]
13     ${txt-error}        xpath://html/body/section/div/div/form/div[1]/span
14     ${txt-error}        //*[@class="alert alert-danger"]
15  *** Test Cases ***
16  Test une connexion à ent Le Mans Université avec log/mdp valide
17      Open Browser      ${LOGIN URL}      ${BROWSER}
18      Maximize Browser Window
19      Sleep             5s
20      Click Link        ${link-personnel}
21      Input Text        ${txtbox-username}   xxxxx
22      Input Text        ${txtbox-password}   xxxxx
23      Click Button      ${btn-seconnecter}
24      Sleep             2s
25      close browser
```

Cas 1





RobotFramework : Data-Driven Testing



Data-Driven Testing (DDT) : exemple /Script 1

```
1  *** Settings ***
2  Documentation      TP BUT3-INFO (IUT-LAVAL)  exemple 1 Test DDT .
3  Library            Selenium2Library
4
5  Test une connexion à ent Le Mans Université avec mdp non valide Cas 2
6    Open Browser      ${LOGIN URL}      ${BROWSER}
7    Maximize Browser Window
8    Click Link        ${link-personnel}
9    Input Text        ${txtbox-username}    llahcen
10   Input Text        ${txtbox-password}    xxxx
11   Click Button      ${btn-seconnecter}
12   Sleep             2s
13   Element Should Be Visible      ${txt-error}    Mauvais identifiant / mot de passe.
14
15  Test une connexion à ent Le Mans Université avec log non valide Cas 3
16    Open Browser      ${LOGIN URL}      ${BROWSER}
17    Maximize Browser Window
18    Click Link        ${link-personnel}
19    Input Text        ${txtbox-username}    llahen
20    Input Text        ${txtbox-password}    xxxx
21    Click Button      ${btn-seconnecter}
22    Sleep             2s
23    Element Should Be Visible      ${txt-error}    Mauvais identifiant / mot de passe.
```





RobotFramework : Data-Driven Testing

Data-Driven Testing (DDT) : exemple / script 2

```
*** Settings ***
Documentation      TP BUT3 (2023-2024)  exemple 1 Test DDT .
Library            Selenium2Library
Test Setup          Debut du test
Test Teardown       Close All Browsers
Resource           RessourcesDDT.robot
Test Template       se connecter avec des identifiants non valides
*** Keywords ***
se connecter avec des identifiants non valides
[Arguments]          ${usernameValid}  ${passwordValid}  ${txterror}
    Click Link        ${link-personnel}
    Input Text        ${txtbox-username}  ${usernameValid}
    Input Text        ${txtbox-password}  ${passwordValid}
    Click Button      ${btn-seconnecter}
    Sleep             2s
    element should contain  ${txterror}  ${txterrorLogNonvalide}
*** Test Cases ***
Test avec un log non valide    USERNAME          PASSWORD          TXTERROR
                                admin            ${passwordValid}  ${txterror}
Test avec un mdp non valide   ${usernameValid}  xxxx            ${txterror}
Test avec mdp vide           ${usernameValid}  ${EMPTY}          ${txterror}
Test avec log vide           ${EMPTY}          ${passwordValid}  ${txterror}
pip3 install --upgrade robotframework-datadriver
```





RobotFramework : Data-Driven Testing



Data-Driven Testing (DDT) : exemple / script 2

```
RessourcesDDT.robot x Test2DDT.robot x
1  *** Settings ***
2  Documentation      TP BUT3-INFO (IUT LAVAL)  exemple 1 Test DDT .
3  Library            Selenium2Library
4  *** Variables ***
5  ${LOGIN URL}       https://ent.univ-lemans.fr/fr/index.html
6  ${BROWSER}          ff
7  ${link-personnel}   xpath://*[contains(text(),"personnel")]
8  ${txtbox-username}  //*[@id="username"]
9  ${txtbox-password}  //*[@id="password"]
10 ${btn-seconnecter}  //*[@contains(text(),"SE CONNECTER")]
11 ${txterror}          //*[@class="alert alert-danger"]
12 ${usernameValid}    llahcen
13 ${passwordValid}    topsecret
14 ${txterrorLogNonvalide}  Mauvais identifiant / mot de passe.
15 ${txterrorLogNonvalide1}  Vous devez entrer votre identifiant.
16 ${txterrorLogNonvalide2}  Vous devez entrer votre mot de passe.
17 *** Keywords ***
18 Debut du test
19  Open Browser      ${LOGIN URL}      ${BROWSER}
20  Maximize Browser Window
21  Sleep             2s
```





RobotFramework : Data-Driven Testing



Data-Driven Testing (DDT) : exemple / Fichier CSV

The screenshot shows the Robot Framework IDE interface. On the left, the test code is displayed:

```
*** Settings ***
Documentation      TP BUT3-INFO (IUT LAVAL) exemple Test data f
Library           Selenium2Library
#Library          DataDriver TestData/DataConLeMansUniv.csv
Library          DataDriver DataConLeMansUniv.csv
Test Setup        Debut du test
Test Teardown     Close All Browsers
Resource          Ressources3DDT.robot
Test Template     se connecter avec des identifiants non valides

*** Keywords ***
se connecter avec des identifiants non valides
    [Arguments]    ${usernameValid}    ${passwordValid}    ${errorMsg}
    Click Link      ${link-personnel}
    Input Text      ${txtbox-username}    ${usernameValid}
    Input Text      ${txtbox-password}    ${passwordValid}
    Click Button    ${btn-seconnecter}
    Sleep           2s
    Element Should Contain    ${txterror}    ${errorMsg}

*** Test Cases ***
Tests de la connexion avec des identifiants non valides    ${usernameValid}    ${passwordValid}    ${errorMsg}
```

A blue arrow points from the line "Library DataDriver DataConLeMansUniv.csv" in the settings section to the CSV file content on the right. The CSV file content is:

	Plugins supporting *.csv files found.
1	\${usernameValid};\${passwordValid};\${errorMsg}
2	"";"Vous devez entrer votre identifiant.
3	"";aaa;Vous devez entrer votre identifiant.
4	bbb;"Vous devez entrer votre mot de passe
5	aaa;bbb;Mauvais identifiant / mot de passe.

```
pip3 install --upgrade robotframework-datadriver
```





RobotFramework : Data-Driven Testing



Data-Driven Testing (DDT) : exemple 1/ Fichier Excel

```
*** Settings ***
Documentation      TP BUT3-INFO (IUT LAVAL) exemple Test data fichier Excel.
Library            Selenium2Library
#Library          DataDriver    TestData/DataConLeMansUniv1.xlsx sheet_name=Feuille1
Library          DataDriver    DataConLeMansUniv1.xlsx sheet_name=Feuille1
Test Setup        Debut du test
Test Teardown     Close All Browsers
Resource          Ressources3DDT.robot
Test Template     se connecter avec des identifiants non valides

*** Keywords ***
se connecter avec des identifiants non valides
[Arguments]    ${usernameValid}    ${passwordValid}
Click Link       ${link-personnel}
Input Text      ${txtbox-username}    ${username}
Input Text      ${txtbox-password}    ${password}
Click Button    ${btn-seconnecter}
Sleep           2s
Element Should Contain   ${txterror}    ${error}
*** Test Cases ***
Tests de la connexion avec des identifiants non valides    ${usernameValid}    ${passwordValid}    ${errorMsg}
```

A blue arrow points from the highlighted section in the Robot Framework code to the corresponding row in the Excel spreadsheet.

A	B	C
\${usernameValid}	\${passwordValid}	\${errorMsg}
	aaa	Vous devez entrer votre identifiant.
bbb		Vous devez entrer votre mot de passe
bbb	aaa	Mauvais identifiant / mot de passe.



pip install -u robotframework-datadriver[XLS]





RobotFramework : Data-Driven Testing



Génération aléatoire des données de test : Librairie Faker

https://pypi.org/project/robotframework-faker/

robotframework-faker 5.0.0

`pip install robotframework-faker`

Add Provider · Address · Am Pm · Android Platform Token · Ascii Company Email · Ascii Email · Ascii Free Email · Ascii Safe Email · Bank Country · Bban · Binary · Boolean · Bothify · Bs · Building Number · Catch Phrase · Century · Chrome · City · City Prefix · City Suffix · Color · Color Name · Company · Company Email · Company Suffix · Coordinate · Country · Country Code · Credit Card Expire · Credit Card Full · Credit Card Number · Credit Card Provider · Credit Card Security Code · Cryptocurrency · Cryptocurrency Code · Cryptocurrency Name · Currency · Currency Code · Currency Name · Date · Date Between · Date Between Dates · Date Object · Date Of Birth · Date This Century · Date This Decade · Date This Month · Date This Year · Date Time · Date Time Ad · Date Time Between · Date Time Between Dates · Date Time This Century · Date Time This Decade · Date Time This Month · Date Time This Year · Day Of Month · Day Of Week · Domain Name · Domain Word · Ean · Ean 13 · Ean 8 · Ein · Email · File Extension · File Name · File Path · Firefox · First Name · First Name Female · First Name Male · Format · Free Email · Free Email Domain · Future Date · Future Datetime · Generator Format Token · Get Formatter · Get Providers · Hex Color · Hexify · Hostname · Iban · Image Url · Init · Internet Explorer · Invalid Ssn · Ios Platform Token · Ipv 4 · Ipv 6 · Ipv4 Network Class · Ipv4 Private · Ipv4 Public · ISBN 10 · ISBN 13 · Iso 8601 · Itin · Job · Language Code · Last Name · Last Name Female · Last Name Male · Latitude · Latlng · Lexify · License Plate · Linux Platform Token · Linux Processor · Local Latlng · Locale · Location On Land · Longitude · Mac Address · Mac Platform Token · Mac Processor · Md 5 · Military Apo · Military Dpo · Military Ship · Military State · Mime Type · Month · Month Name · Msisdn · Name · Name Female · Name Male · Null Boolean · Numerify · Opera · Paragraph · Paragraphs · Parse · Password · Past Date · Past Datetime · Phone Number · Postcode · Postcode In State · Postcode Plus4 · Postcode · Postcode In State · Prefix · Prefix Female · Prefix Male · Profile · Provider · Pybool · Pydecimal · Pydict · Pyfloat · Pyint · Pyiterable · Pylist · Pyset · Pystr · Pystr Format · Pystruct · Pytuple · Random Choices · Random Digit · Random Digit Not Null · Random Digit Not Null Or Empty · Random Digit Or Empty · Random Element · Random Elements · Random Int · Random Letter · Random Letters · Random Lowercase Letter · Random Number · Random Sample · Random Uppercase Letter · Randomize Nb Elements · Rgb Color · Rgb Css Color · Safari · Safe Color Name · Safe Email · Safe Hex Color · Secondary Address · Seed · Seed Instance · Sentence · Sentences · Set Formatter · Sha 1 · Sha 256 · Simple Profile · Slug · Ssn · State · State Abbr · Street Address · Street Name · Street Suffix · Suffix · Suffix Female · Suffix Male · Text · Texts · Time · Time Delta · Time Object · Time Series · Timezone · Tld · Unix Device · Unix Partition · Unix Time · Upc A · Upc E · Uri · Uri Extension · Uri Page · Uri Path · Uri · User Agent · User Name · Uuid 4 · Windows Platform Token · Word · Words · Year · Zipcode · Zipcode In State · Zipcode Plus4

Documentation <https://guykisel.github.io/robotframework-faker/>





RobotFramework : Data-Driven Testing



Librairie Faker : Exemple

https://pypi.org/project/robotframework-faker/

robotframework-faker 5.0.0

pip install robotframework-faker

```
*** Settings ***
#Library    FakerLibrary    locale=ar_AA    #Argument locale pour definir le pays
Library    FakerLibrary    locale=fr    #Argument locale pour definir le pays
Library    CSVLibrary
Library    SeleniumLibrary

*** Keywords ***
Faker_profile1
[Arguments]    ${user_nombre}
FOR    ${i}    IN RANGE    1    ${user_nombre}
    ${first_name} =  FakerLibrary.first_name
    ${last_name} =  FakerLibrary.last_name
    ${email_user} =  FakerLibrary.Email
    ${email_perso} =  Catenate    ${first_name}.${last_name}@gmail.com
    ${date_naissance} =  FakerLibrary.Date Of Birth    tzinfo=None    minimum_age=18    maximum_age=50
    ${number} =  Random Number
```

Documentation <https://guykisel.github.io/robotframework-faker/>





RobotFramework : XPath in Selenium

```
1  *** Settings ***
2  Documentation      TP BUT3-INFO (IUT-LAVAL) exemple 1 Test DDT .
3  Library            Selenium2Library
4  *** Variables ***
5  ${LOGIN URL}       https://ent.univ-lemans.fr/fr/index.html
6  ${BROWSER}         ff
7  ${link-personnel}  xpath://html/body/main/div/div/div/ul/li[2]/a
8  ${link-personnel}  xpath://*[contains(text(),"personnel")]
9  ${txtbox-username} //*[@id="username"]
10 ${txtbox-password} //*[@id="password"]
11 ${btn-seconnecter} xpath://html/body/section/div/div/form/div[3]/div/button
12 ${btn-seconnecter} //*[contains(text(),"SE CONNECTER")]
13 ${txt-error}        xpath://html/body/section/div/div/form/div[1]/span
14 ${txt-error}        //*[@class="alert alert-danger"]
15 *** Test Cases ***
16 Test une connexion à ent Le Mans Université avec log/mdp valide
17 Open Browser        ${LOGIN URL}    ${BROWSER}
18 Maximize Browser Window
19 Sleep               5s
20 Click Link          ${link-personnel}
21 Input Text          ${txtbox-username}   xxxxx
22 Input Text          ${txtbox-password}    xxxxx
23 Click Button        ${btn-seconnecter}
24 Sleep               2s
25 close browser
```

XPath absolu

XPath relatif





RobotFramework : XPath in Selenium

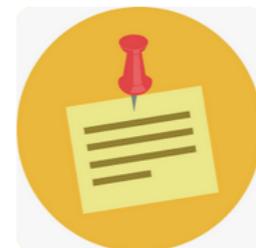
- **XPath absolu**
 - Méthode la plus facile pour avoir un xpath
 - Inconvénient : Non valide si la structure de la page change
 - Exemple : `xpath://html/body/main/div/div/div/div/ul/li[2]/a`
- **XPath relatif**
 - Récupère l'élément HTML directement au milieu de la structure DOM (Document Object Model) → voir norme W3C
 - Exemple : `//*[@class="alert alert-danger"]`

Remarque (Bonnes pratiques)

*Utilisation de **id** (ou les data-attributs dédiés pour les tests ex : "**data-qa**" ou "**data-testid**") reste une meilleure méthode pour manipuler les éléments d'une page*

→ réduire le coût de maintenance des scripts.

==> <https://www.guru99.com/xpath-selenium.html>





RobotFramework ... + + + ...

- rpa-robotframework (Cours 2)
 - ➔ pip install rpa-robotframework
 - ➔ Pip install robotframework-rpa
- AppiumLibrary: Tests applications Mobile
 - ➔ pip install --upgrade robotframework-appiumlibrary
- SikuliLibrary : Tests IHM
 - ➔ pip install robotframework-SikuliLibrary
- DatabaseLibrary: Tests Base de données
 - ➔ pip install robotframework-databaselibrary



Aussi ...Création des nouvelles librairies

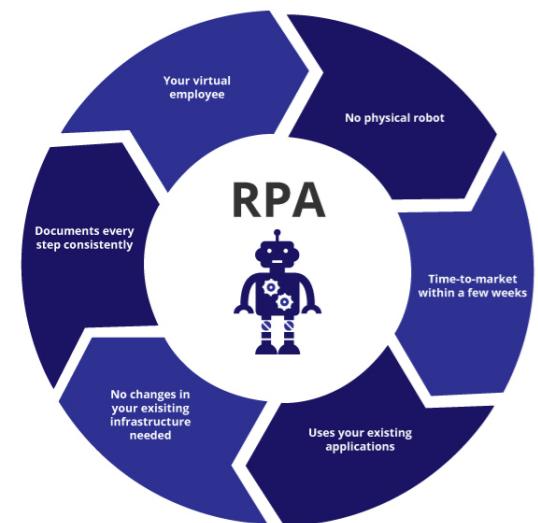




RobotFramework ... suite...



Cours 2 : RPA et Automatisation des tests





```
if (question || questions)  
    Lahcen.answer ();  
  
else  
  
    etudiant.TP (TP1);
```

