

Telecom
Valley

Soirée du
< Test Logiciel >

9 Grands Frameworks d'automatisation de test
pour accélérer la transformation de vos tests

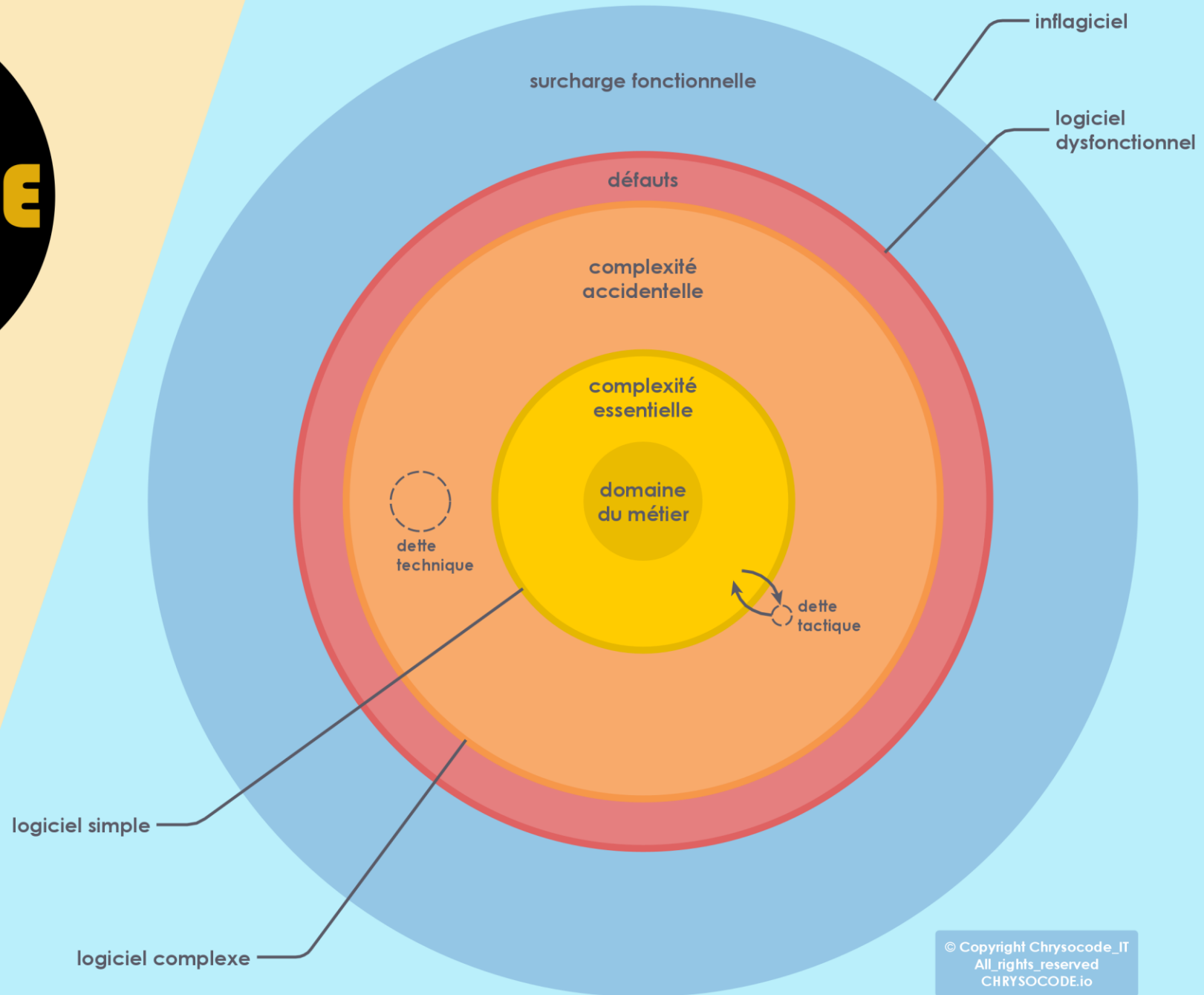
Xavier Pigeon



10/12/2020

CHRYSO CODE

**Visez
l'Essentiel(le)**





Nous sommes convaincus qu'un logiciel pensé pour être utile doit aussi être un produit fiable et bien construit, conçu pour absorber le changement continuellement, pour être livré à volonté en toute sérénité, pour durer sans douleur tant qu'il rencontre sa cible.

chrysocode.io

Transformation des tests



Framework d'automatisation de test



Cadre de travail méthodologique pour mener des projets de développement de test automatisé, et qui se différencie par un parti pris dans la manière d'automatiser les tests.



1. Framework par script linéaire

9. Framework de test hybride

2. Framework de test modulaire

3. Framework de bibliothèques de fonctions de test

8. Framework de test fondé sur des modèles

4. Framework de test piloté par les données

5. Framework de test piloté par mots d'action

6. Framework de test fondé sur des composants

7. Framework de test piloté par les comportements

Que peut devenir un cas de test à automatiser ?

Cas de test manuel		
Action 1	➔	Résultat 1
Action 2	➔	Résultat 2
Action 3	➔	Résultat 3

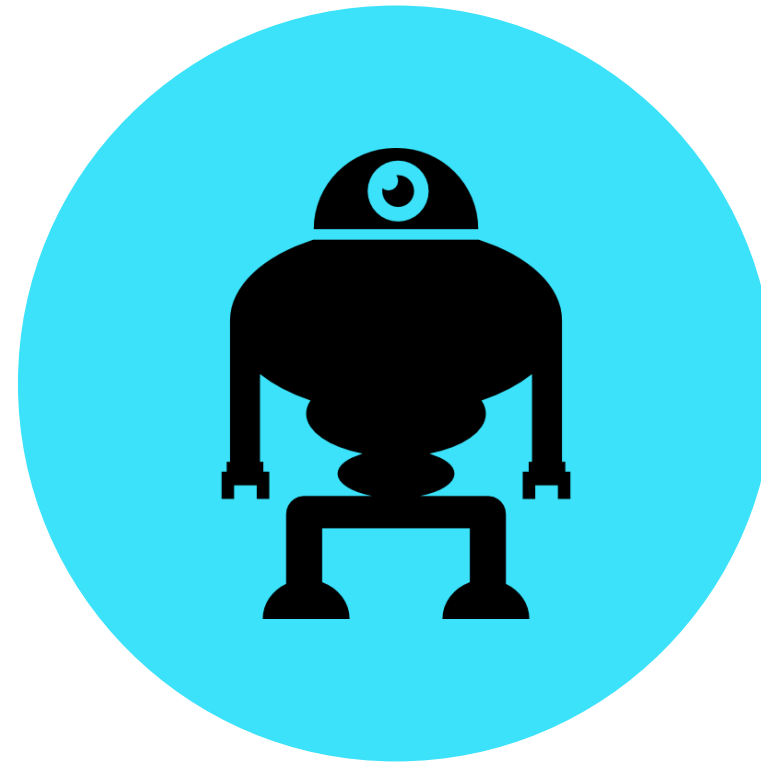
1



Framework par script linéaire

le plus
basique

linear scripting
framework





- Données et code fusionnés
- Pas de scénario explicite

Framework par script linéaire

À utiliser...

- en l'absence de compétence en programmation et en automatisation ;
- pour produire rapidement un filet d'anti-régression sur des parties d'un logiciel qui n'évoluent plus, mais qui pourraient subir des régressions distantes.

À implémenter avec...

- TestComplete Platform
- Katalon Studio
- Micro Focus Unified Functional Testing (UFT)
- mabl
- Ranorex
- Selenium IDE
- LEAPWORK
- Appium
- Eggplant Functional



Solution *nocode* comme Agilitest

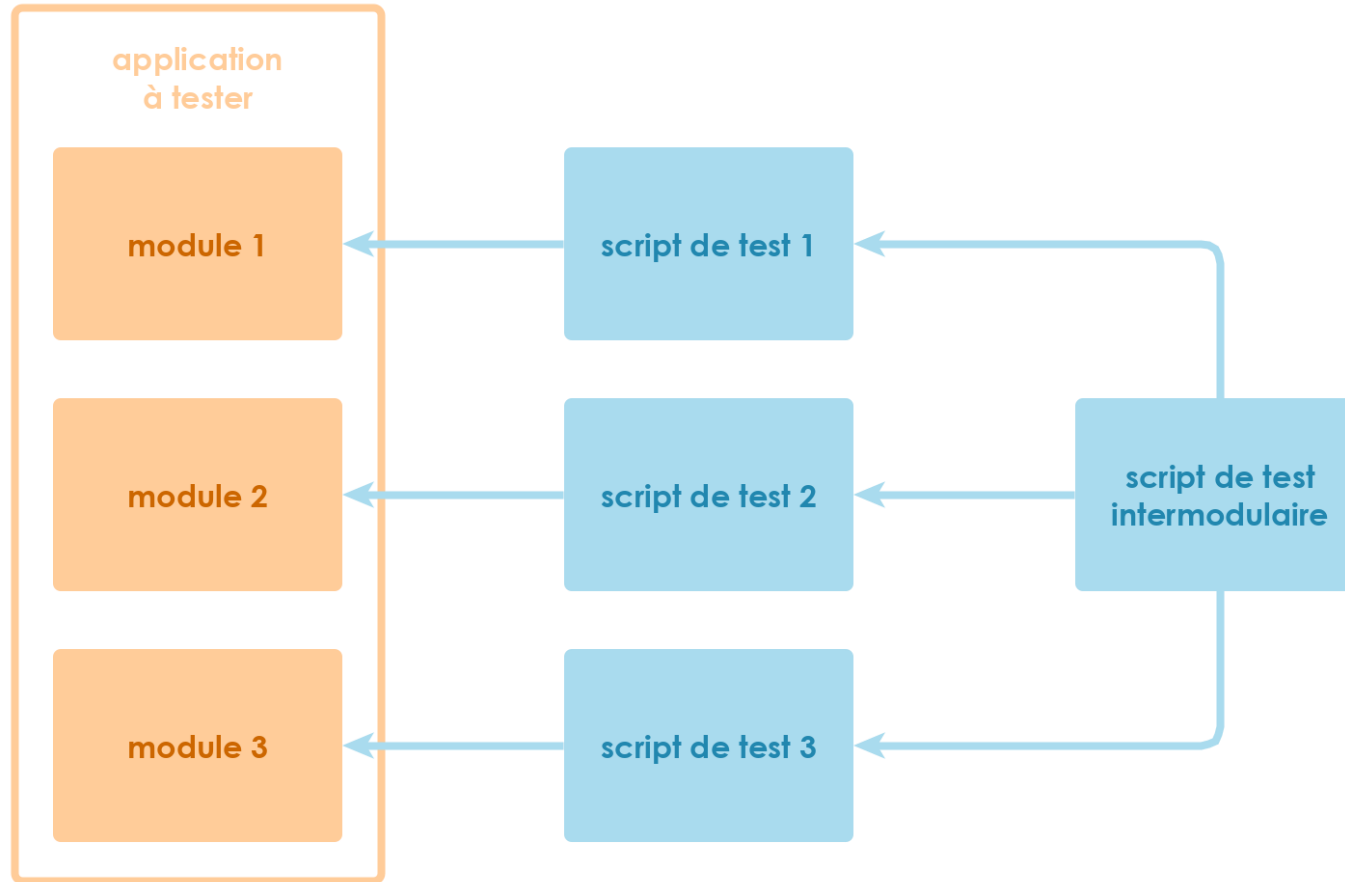
2



Framework de test modulaire

un peu moins
basique

modular-based
testing framework



cc_by-nc-nd xavier_pigeon
xavierpigeon.com



- Données et code fusionnés
- Pas de scénario explicite

Framework de test modulaire

À utiliser...

- pour organiser le code de test ;
- pour rendre le code de test réutilisable pour des cas de test à l'échelle de plusieurs modules.
- Uniquement adapté à une architecture modulaire, où chaque module fonctionnel peut être testé en isolation.

À implémenter avec...

- N'importe quelle API de test.



Framework par
script linéaire

Écriture de
scripts from
scratch

Framework de
test modulaire

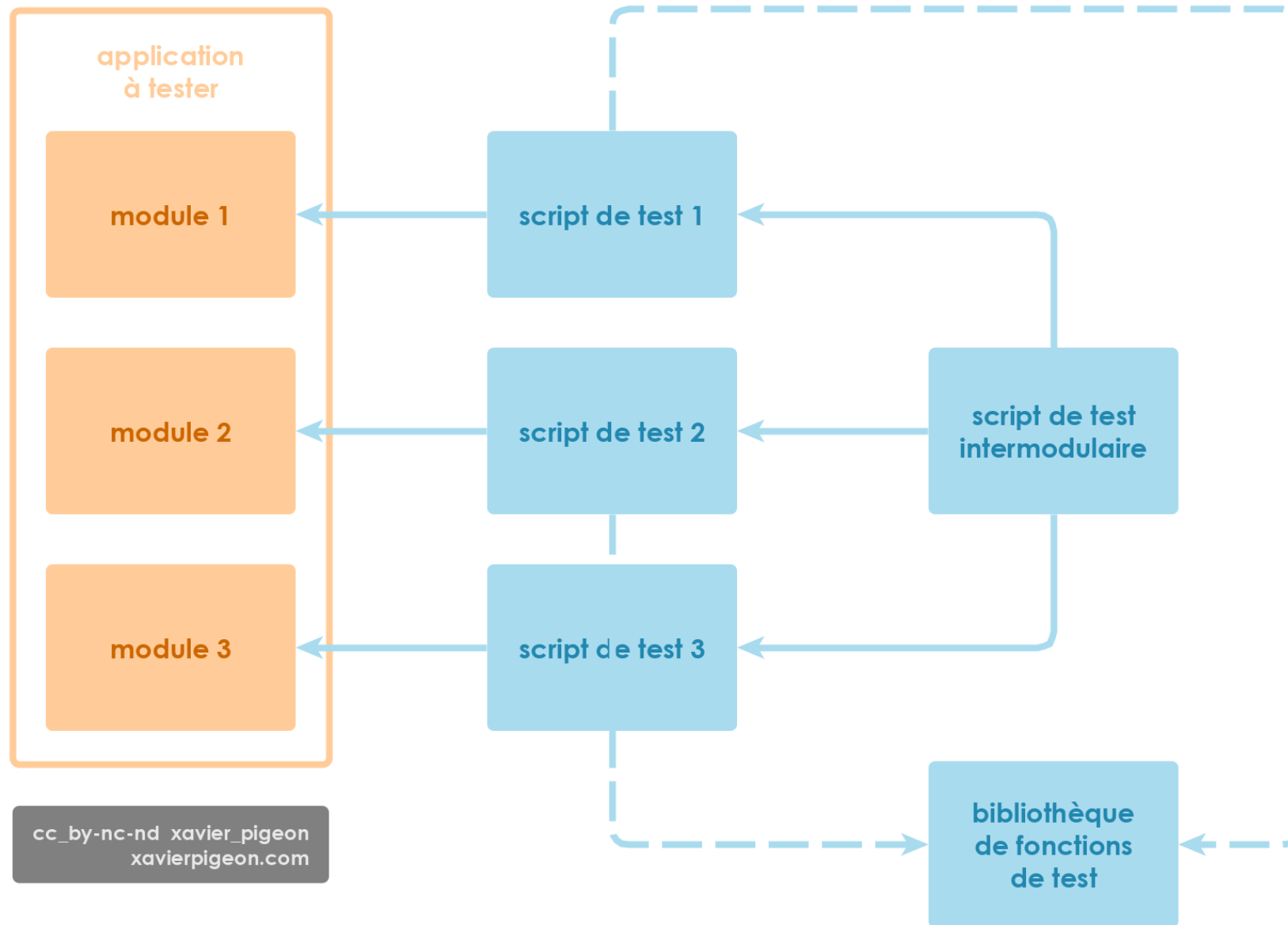
3



Framework de bibliothèques de fonction de test

le plus
structuré

function library-based
testing framework



cc_by-nc-nd xavier_pigeon
xavierpigeon.com



Données & Code

- Séparation du code par module
- Séparation des données et des fonctions de test (paramètres) dans les scripts
- Données fusionnées dans les scripts

Pas de scénario explicite

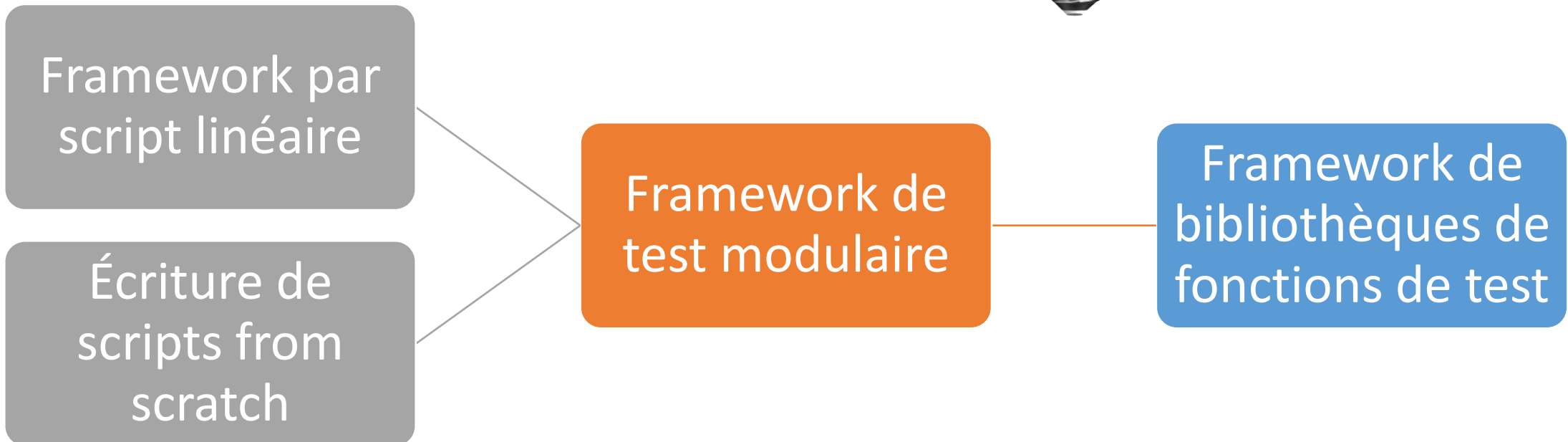
Framework de bibliothèques de fonctions de test

À utiliser...

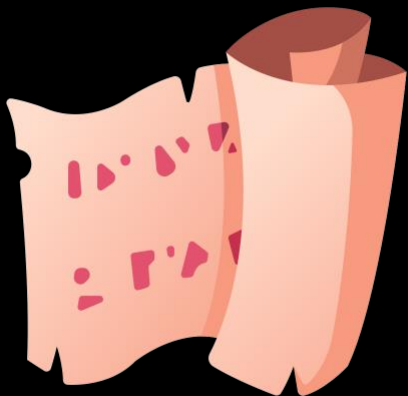
- pour rendre le code de test réutilisable pour différents cas de test.

À implémenter...

- avec n'importe quelle API de test



4



Framework de test piloté par les données

le moins formel
parmi les
formalistes

data-driven testing
framework

Intitulé	Entrée 1	Entrée 2	Sortie
Test 1	Alpha	123	alpha123
Test 2	Bêta	456	beta456
Test 3	Gamma	789	gamma789

Composant (classe)
- entree1 - entree2 - sortie
+ setEntree1() + setEntree2() + setSortie() + execute()



Exécution



Données et code séparés
Pas de scénario explicite

Framework de test piloté par les données

À utiliser...

- quand un même scénario doit être réexécuté pour différents jeux de données, sachant qu'un jeu de données en l'occurrence définit un cas de test particulier.

À implémenter avec...

- FitNesse
- QTP
- Robot Framework
- N'importe quelle API de test et une bibliothèque pour lire des fichiers Excel (Apache POI, JXL API)



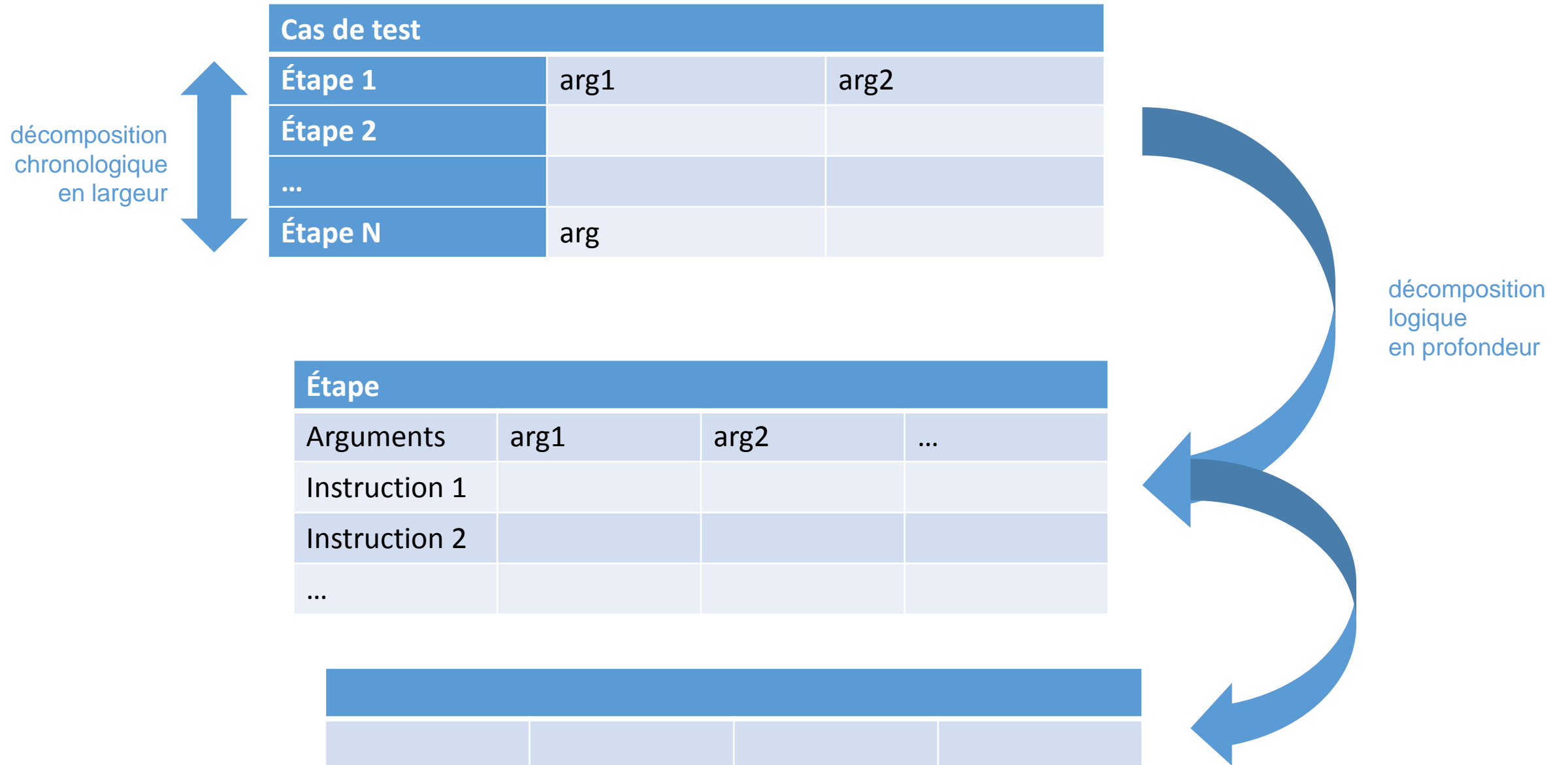
5

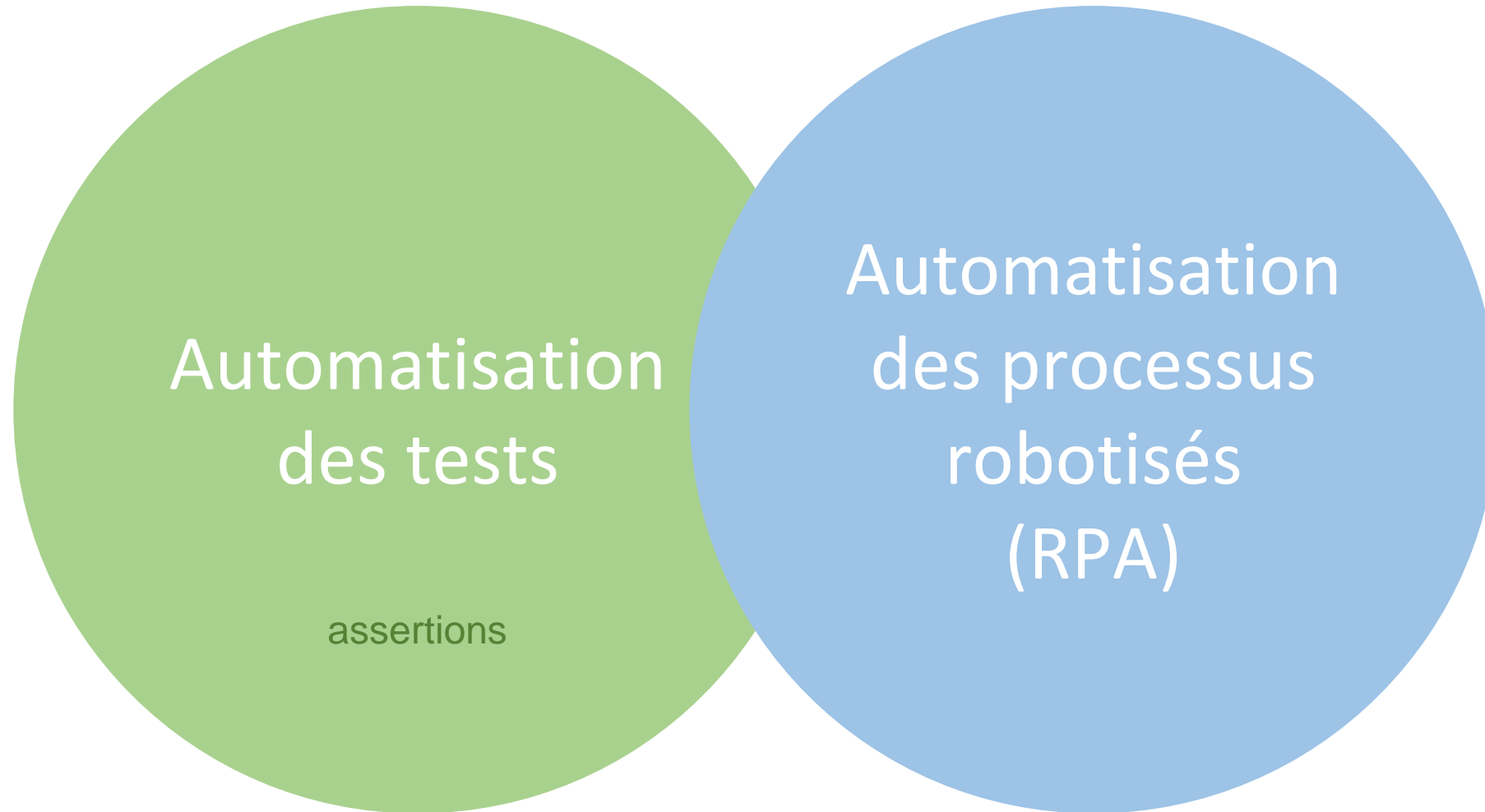


Framework de test
piloté par mots
d'action

le plus
avancé

keyword-driven
testing framework







- ✓ Séparation des données et des mots d'action (paramètres) dans les tests
- ✓ Séparation des mots d'action et des fonctions de test grâce à des bibliothèques

Scénario et code séparés
Scénario explicite

Framework de test piloté par mots d'action

À utiliser...

- pour favoriser la collaboration entre les acteurs contribuant à un produit ;
- pour que les tests deviennent une spécification exécutable du produit et adopter ATDD / BDD ;
- quand l'automatisation repose sur des technologies multiples et hétérogènes.

À implémenter avec...

- Robot Framework
- Ranorex
- TestComplete

6



Framework de test
fondé sur des
composants

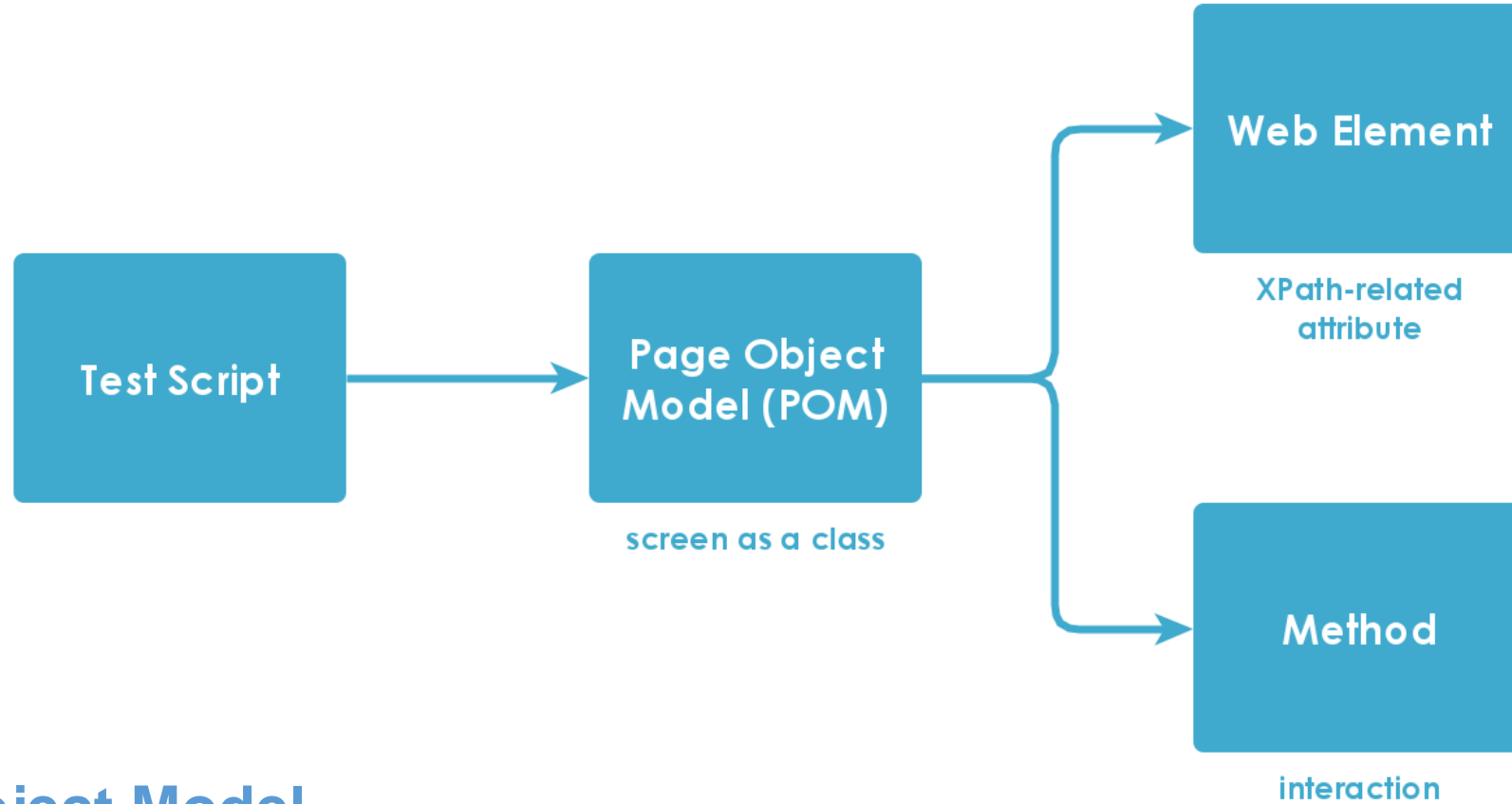
le plus avancé
en POO

component-based
testing framework



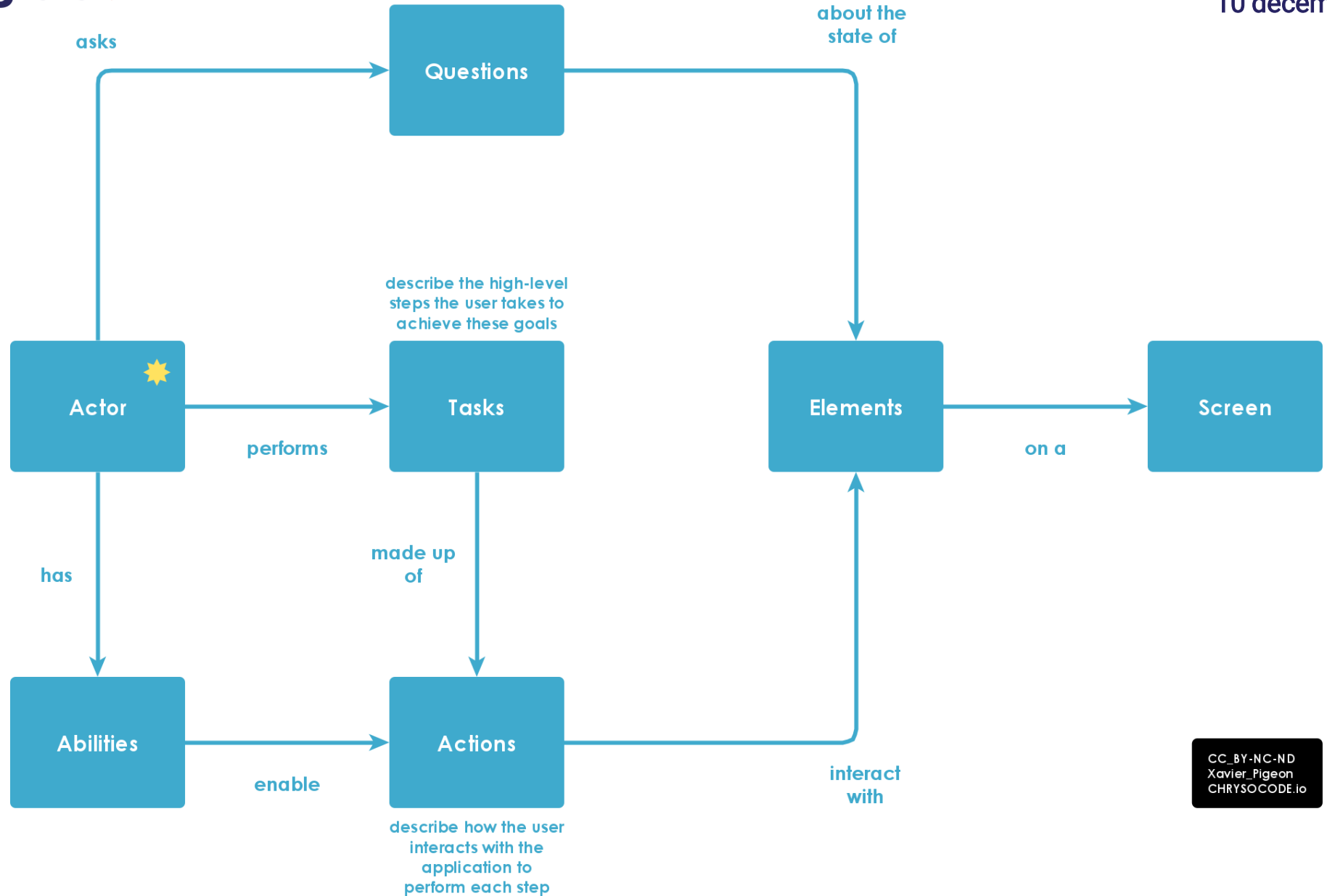
Modèle de programmation
appliqué au test

- ✓ Séparation des données et des fonctions de test (paramètres) dans les tests
- ✓ Séparation des responsabilités dans le code par des composants de test
- ✗ Pas de scénario explicite



Page Object Model (POM)

Screenplay



CC BY-NC-ND
Xavier Pigeon
CHRYSOCODE.io

Framework de test fondé sur des composants

À utiliser...

- si les compétences en programmation orientée objet sont réunies ;
- pour adopter une démarche d'automatisation centrée sur le produit (POM) ou sur l'utilisateur (Screenplay).

À implémenter avec...

- Serenity
- N'importe quelle API de test

7

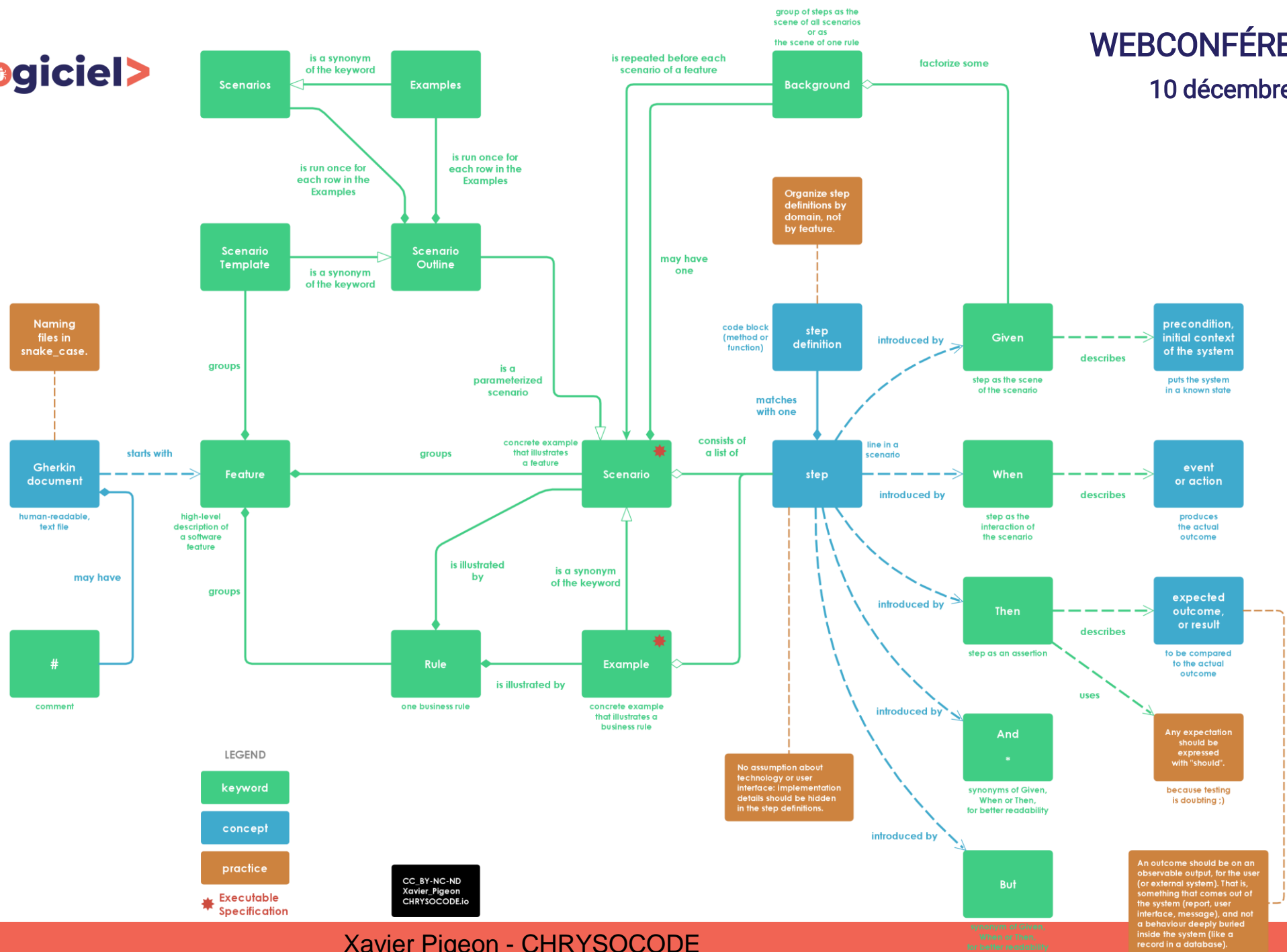


Framework de test piloté par les comportements

le plus prescriptif
en automatisation
littérale

behavior-driven
development
framework

Framework de test fondé sur Gherkin



- ✓ Séparation des données et des étapes de test (paramètres) dans les tests



Scénario (comportement)
et code séparés
Scénario explicite



Données (exemples)
et scénario (patron)
séparables

Framework de test piloté par les comportements

À utiliser...

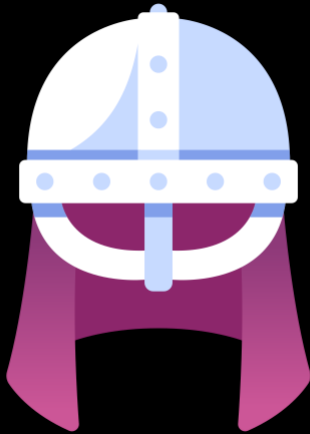
- pour favoriser la collaboration entre les acteurs contribuant à un produit (3 Amigos) ;
- pour que les tests deviennent une spécification exécutable du produit.

À implémenter avec...

- Cucumber
- SpecFlow
- Behave
- JBehave
- Behat
- Robot Framework
- Cypress
- Codecept



8



Framework de test fondé sur des modèles

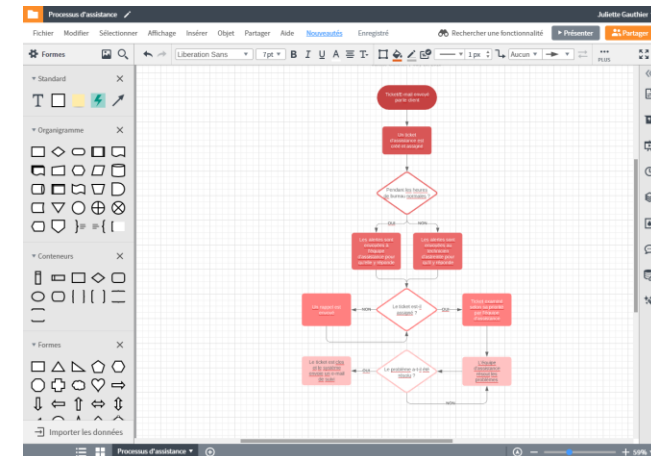
le plus visuel en
conception de
test

model-based testing
framework



- ✓ Séparer la conception (logigrammes) et la définition (1 parcours / chemin d'un logigramme) des cas de test.

Scénario et code séparés
Scénario explicite



Framework de test fondé sur des modèles

À utiliser...

- pour favoriser la collaboration entre les acteurs contribuant à un produit ;
- pour faciliter la définition des cas de test : plusieurs cas de test à partir d'un seul logigramme ;
- pour adopter ATDD / BDD.

À implémenter avec...

- Yest
- MaTeLo

9



Framework hybride

le plus
complet

hybrid testing
framework



Keyword-Driven Testing
+ Data-Driven Testing

Keyword-Driven Testing
+ Data-Driven Testing
+ Model-Based Testing



Keyword-Driven Testing
+ Gherkin (BDD)

Gherkin (BDD)
+ POM

Gherkin (BDD)
+ Screenplay

Framework de test hybride

À utiliser...

- pour tirer profit des forces de différents frameworks et ajouter des cordes à son arc.

À implémenter...

- en combinant des technologies compatibles entre elles (Cucumber.js et Protractor par exemple) ;
- en combinant des fonctionnalités complémentaires de certaines technologies polyvalentes (Robot Framework : keyword-driven style, data-driven style, behavior-driven style).

Frameworks de test : une histoire d'opportunités !

- Séparer les éléments structurants des tests (données, scénario, code, composants) en choisissant une, voire des couches d'abstraction, amènent de nouvelles opportunités :
 - Prise en compte des contraintes et des enjeux de l'entreprise
 - Meilleure ingénierie des tests dans les équipes
 - Consensus autour des bonnes pratiques de test
 - Plus de collaboration entre des métiers variés (marketing, test, développement, opération)
 - Compréhension partagée du besoin
 - Guide à la conception et au développement
 - Qualité logicielle intrinsèque

Obtenez votre
LIVRE
GRATUIT !



Testing-Driven
Craftsmanship
avec CHRYSOCODE

AUTOMATISATION DES TESTS FONCTIONNELS

Frameworks &
Bonnes Pratiques

LIVRE BLANC CO-ÉCRIT ET PUBLIÉ PAR

All4Test





<https://chrysocode.io/contact.html>

Vers plus de maîtrise de la production logicielle

- Ingénierie du Test
 - Stratégie d'automatisation de test
 - Automatisation des tests multi-niveaux
 - Automatisation Littérale & Spécification Exécutable
- Ingénierie Logicielle Agile & Software Craftsmanship
- Ingénierie des exigences
- Expertise-conseil
- Accompagnement / coaching
- Réalisation
- Formation



contact@chrysocode.io

Découvrez notre **proposition de valeur**
à travers notre pitch deck (très visuel) :

<https://tinyurl.com/chrysopitch>





Xavier Pigeon est Ingénieur Logiciel, et évolue aujourd'hui en tant qu'Expert Méthode & Qualité en Stratégie IT. Il est aussi l'auteur du framework méthodologique GOST (gearsoftesting.org) dédié à une approche holistique de gestion de la qualité logicielle et à la conception de stratégies adaptatives de test.

Xavier se consacre au coaching organisationnel et technique, en associant étroitement Agilité, Test et *Software Craftsmanship* (artisanat du code et compagnonnage logiciel). Il accompagne notamment les équipes dans leur cheminement vers l'excellence ingénierique, en les guidant dans leur appropriation de méthodes et pratiques d'ingénierie aligné avec un monde VUCA (volatile, incertain, complexe et ambigu).

En 2019, Xavier a fondé la marque CHRYSOCODE et sa société Chrysocode IT (chrysocode.io) dont il est le dirigeant.