



Think Tank – Do Tank

sur l'IA Générative appliquée au domaine de l'Ingénierie Logicielle

3ème édition – Mercredi 28 janvier 2026

Groupe de Travail : Automatisation des tests.

GT « Automatisation des tests »

Réunion Générale - 3ème édition

Groupe de Travail : Automatisation des tests .



Massinissa MECHAREK : Expert Test & IA



Francis VACHETTE : Robot Framework Adept



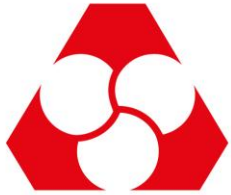
Sylvain CAILLIAU : Directeur Technique France de CAST



01

Introduction

Les sociétés ayant participé au GT



Crédit  Mutuel

amadeus

hagergroup



MICHELIN

bpifrance



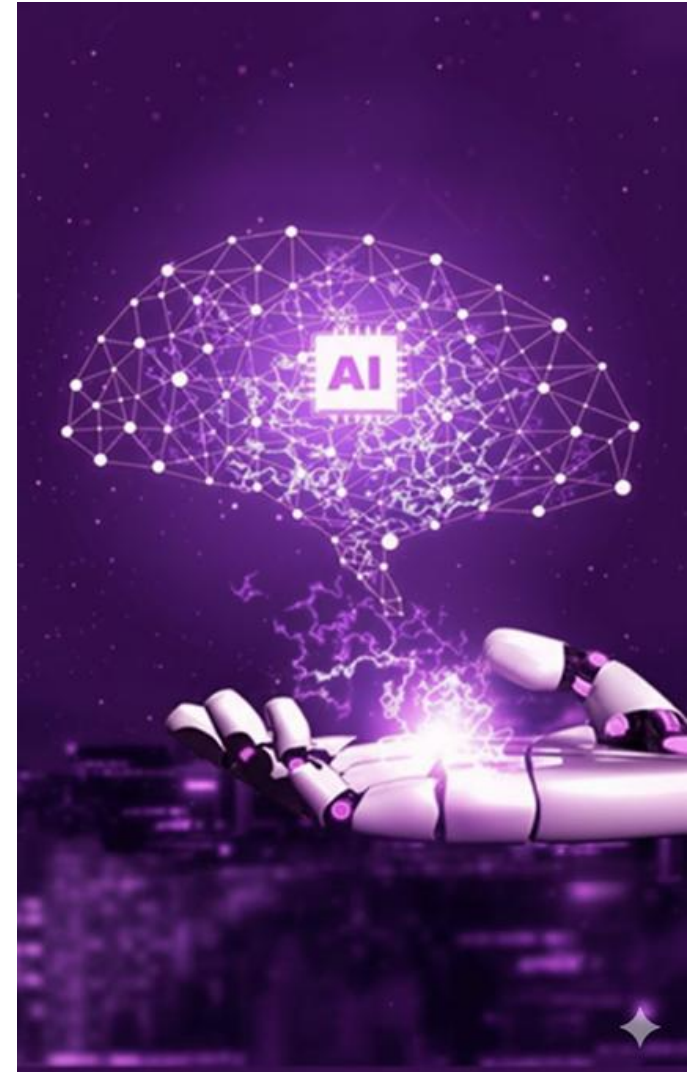
ODDO BHF

02

La Révolution de l'IA Générative pour le cycle du Test

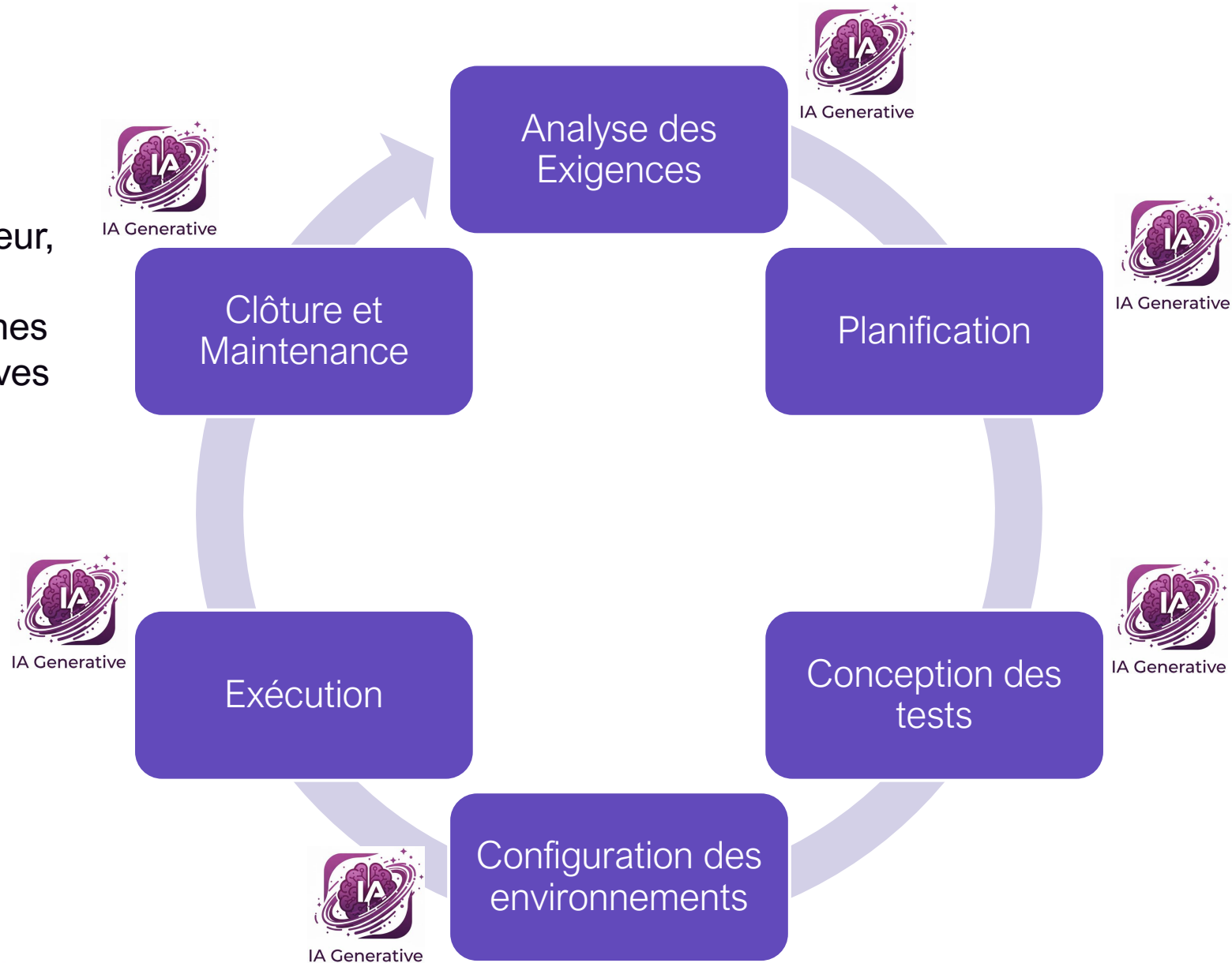
Conclusions Think Tank 2025

- Dans un contexte structuré/normé l'IA générative va nous permettre **d'accélérer la production des tests.**
- A l'heure actuelle **pas de solution éditeur opérationnelle** mais on constate un engouement du marché.
- **L'expertise est toujours nécessaire** pour garantir la complétude des tests
- L'enjeu pour réussir l'IA dans les tests est de **structurer les patrimoines documentaires.**
- Evaluation des gains & passage à l'échelle à instruire.

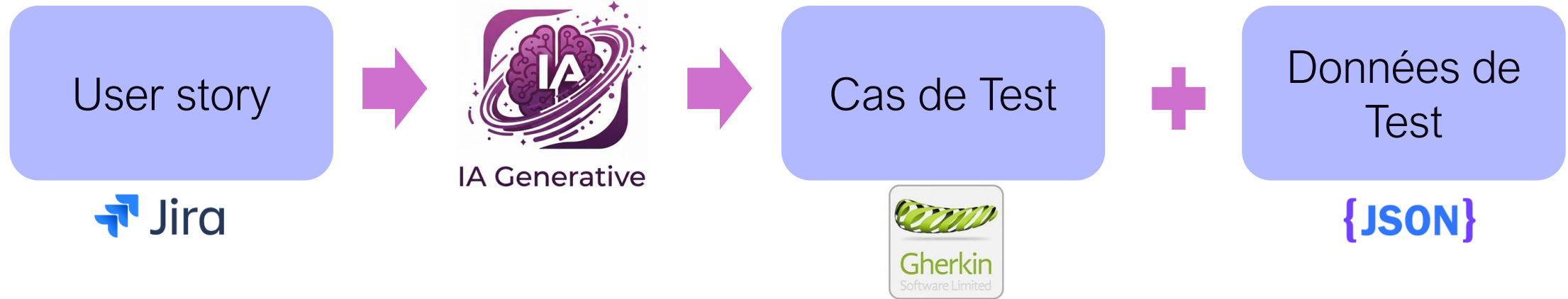


L'IA Générative : Un Accélérateur pour Chaque Phase

L'IA Générative ne remplace pas le testeur, elle l'augmente, en automatisant les tâches cognitives et répétitives



De la Spécification au Plan de Test



- **Génération de cas de test** à partir des exigences en langage naturel.
- **Identification des zones à risque** en analysant la complexité du code.
- **Génération de données de test** variées et pertinentes.



Génération de Scripts de Test

1. Un testeur clique sur une UI. 2. L'IA "observe". 3. L'IA génère le code du script Playwright (ou Cypress...)



- **"Record & Replay" Augmenté** : L'IA transforme les actions manuelles en code propre et robuste.
- **Génération de code de test** directement depuis le code source de l'application.



Outils/Solutions



GitHub
Copilot
for QA



applitools
ScriptWriter



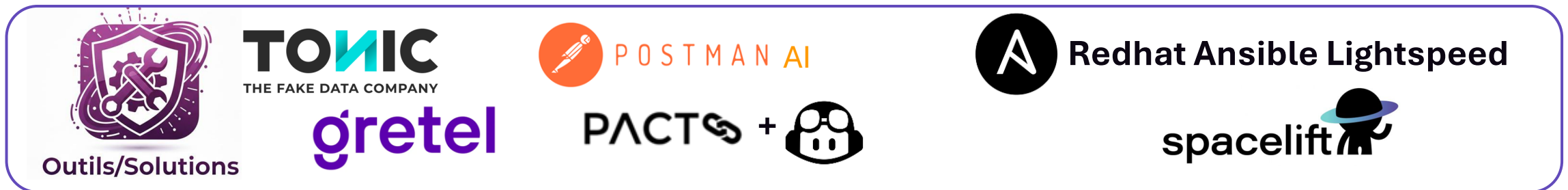
Playwright



codeium

Maintenance des Ressources de Test

Données de Tests	Mocks et Bouchons	Environnements
Défi : Données obsolètes ou peu réalistes.	Défi : Dérive des contrats d'API.	Défi : Dérive de configuration (QA ≠ Prod).
Solutions IA : 1. Génération Synthétique : L'IA lit le schéma de la DB et génère des données réalistes et variées. 2. Auto-Adaptation : L'IA détecte un changement de schéma (migration) et met à jour le corpus de données.	Solutions IA : 1. Génération depuis Contrat : L'IA lit une spécification OpenAPI et génère un serveur de bouchons complet. 2. Auto-Correction : L'IA détecte un changement sur l'API réelle et propose une Pull Request pour mettre à jour le bouchon.	Solutions IA : 1. Génération d'Infrastructure-as-Code : L'IA génère du code Terraform, Ansible ou des manifestes Kubernetes à partir d'une description (langage naturel) 2. Détection et Correction de Dérive : L'IA compare les configurations de QA et de Prod, identifie les écarts et génère le script pour corriger la dérive.



Du Constat à l'Analyse de Cause Racine (Root Cause)



- **Analyse de cause racine (Root Cause Analysis) :** Corrélation des logs, traces et changements de code pour identifier la source de l'erreur.
- **Tri intelligent des bugs :** L'IA peut identifier les bugs dupliqués ou prédire la sévérité d'un bug.



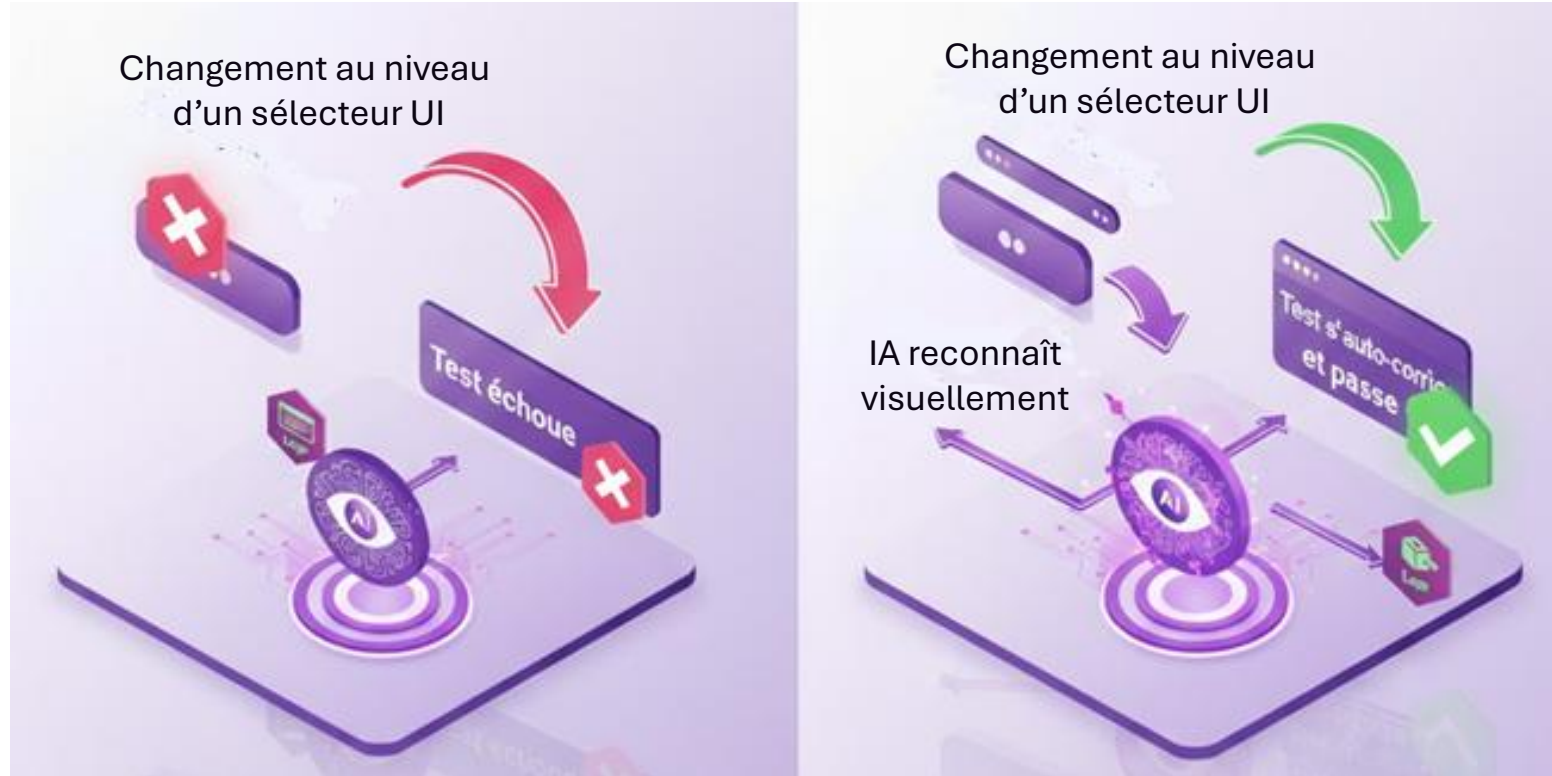
Outils/Solutions

testim
AI Root Cause



Maintenance du Corpus de Tests

L'Auto-Réparation (Auto-Healing) des Tests



- **Détection de tests obsolètes** après suppression de code.
- **Identification de tests redondants** par analyse sémantique.



Outils/Solutions



mabl



applitools



functionize

03

Synthèse et Vision

Tableau Comparatif : Avant et Après l'IA Générative

Tâche	Approche Classique (Manuelle/Automatisée)	Approche Augmentée (IA Générative)
Conception des tests	Manuelle, basée sur l'expérience	Automatisée, basée sur les exigences
Génération de données	Manuelle ou via scripts simples	Synthétique, réaliste et auto-adaptative
Maintenance des tests UI	Manuelle, coûteuse, réactive	Automatisée, pro-active (Auto-Healing)
Analyse des échecs	Manuelle, investigation longue	Automatisée, analyse de cause racine

Mais attention aux différences intrinsèques entre les générations d'applications:

- Le test des applications modernes s'appuie sur une conception découplée permettant une automatisation granulaire et continue,
- Tandis que le monolithe historique impose une approche "boîte noire" où l'adhérence des composants interdit toute validation isolée sans une reconstruction préalable des contrats d'interface.



Le Nouveau Rôle du Testeur

Rééquilibrage des Activités : Moins de Tâches, Plus de Stratégie



L'IA ne remplace pas l'ingénieur QA, elle doit être considéré comme un assistant qui le libère des tâches à faible valeur pour le laisser devenir un véritable stratège de la qualité produit.

En baisse

- Écriture de code de test répétitif (boilerplate).
- Réparation manuelle des tests UI (auto healing).
- Création manuelle de données de test basiques.
- Exécution et documentation des tests de non-régression.

En hausse

- **Stratégie de Qualité** : Définir *quoi* tester et *comment*, en fonction des risques.
- **Tests Exploratoires** : Rechercher les bugs complexes et imprévus par la créativité et l'intuition.
- **Analyse de Risques** : Comprendre l'impact métier des fonctionnalités.
- **Pilotage de l'IA** : "Prompter" et valider les suggestions de l'IA, gérer la chaîne d'outils.
- **Collaboration** : Participer plus en amont à la conception (Shift Left).

Feuille de Route vers le Test Augmenté

3mois

- Intégrer Copilot/Codeium dans l'IDE* pour aider à l'écriture des tests unitaires.

6mois

- Adopter une plateforme de test de bout-en-bout avec des capacités d'auto-réparation (Mabl, Testim).

12 mois

- Mettre en place une solution de génération de données de test synthétiques (Tonic.ai).

18+ mois

- Intégrer une plateforme d'intelligence de test (Launchable) pour optimiser l'exécution.

* IDE : Integrated Development Environment – Environnement de Développement Intégré.

Synthèse des Apports et Conclusion Générale

Synthèse : Vitesse, Qualité, Coûts



- Réduction drastique du temps de création et de maintenance des tests.
- Feedback plus rapide dans le pipeline CI/CD.



- Meilleure couverture des cas limites.
- Tests plus robustes et moins "flaky".



- Réduction du coût de la maintenance.
- Détection des bugs plus tôt dans le cycle (Shift Left).

En 2026

- (Toujours) **Pas de solution "magique"** : La performance vient d'une chaîne d'outils spécialisés et bien intégrés, pas d'un seul ou d'une suite cohérente.
- **L'effort se déplace** néanmoins : Moins de temps sur l'écriture manuelle, plus sur l'intégration et le pilotage d'outils IA.
- **L'humain reste au centre** : La stratégie, la créativité (tests exploratoires) et la compréhension du métier restent irremplaçables.



04

Retours d'expérience de Bouygues Telecom

28/01/2026

INTERNE

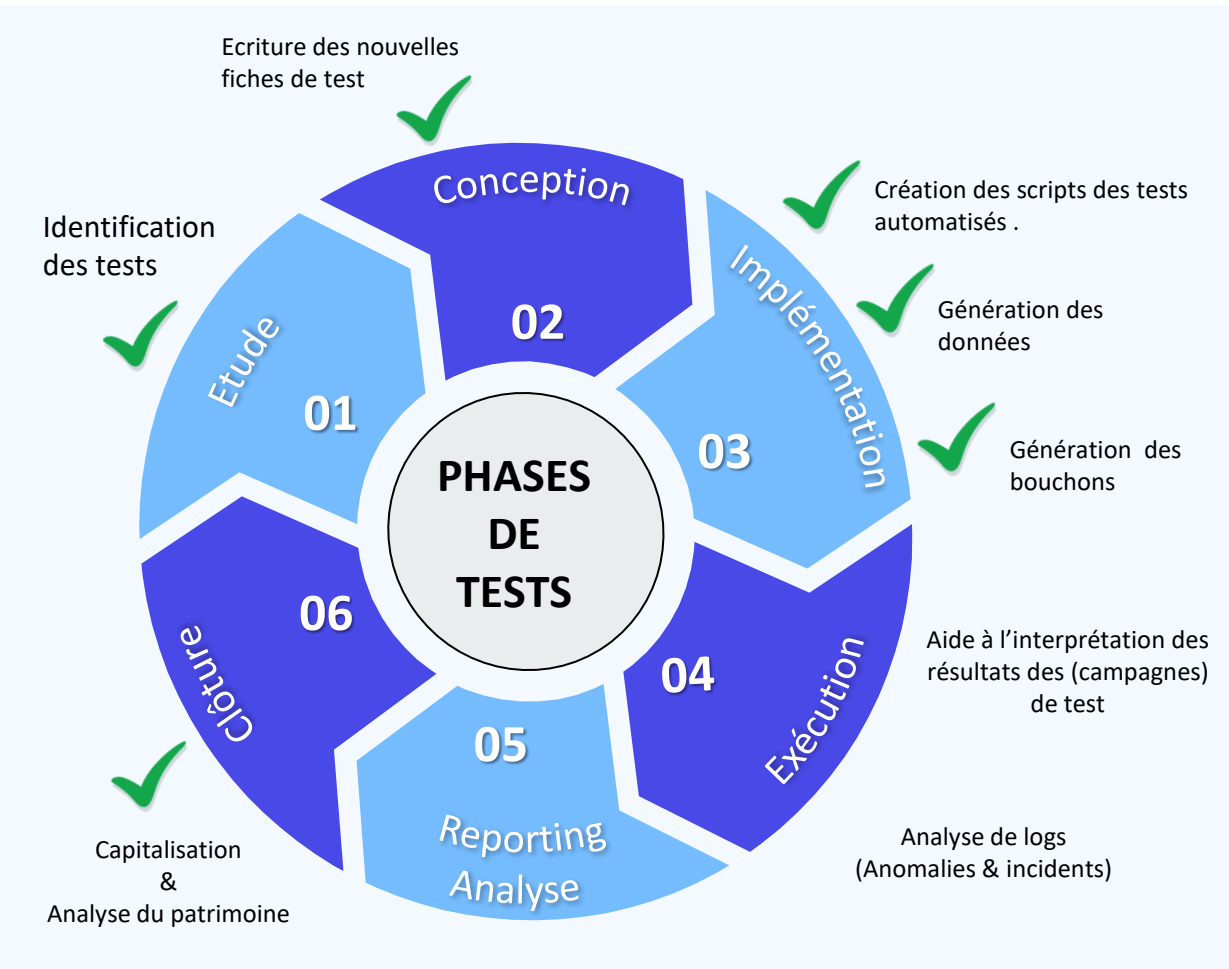
Groupe de travail N4

Automatisation des tests

RETOUR EXPÉRIENCE BOUYGUES TELECOM



Chaine de fabrication des tests



Nos applications



CATIA

- Identification des tests
- Création des scripts des tests automatisés
- Génération des bouchons



LUPIN

- Ecriture des nouvelles fiches de test
- Capitalisation & analyse du patrimoine



CHARLIE

- Réservation des jeux de données

Nos Pocs



MCP

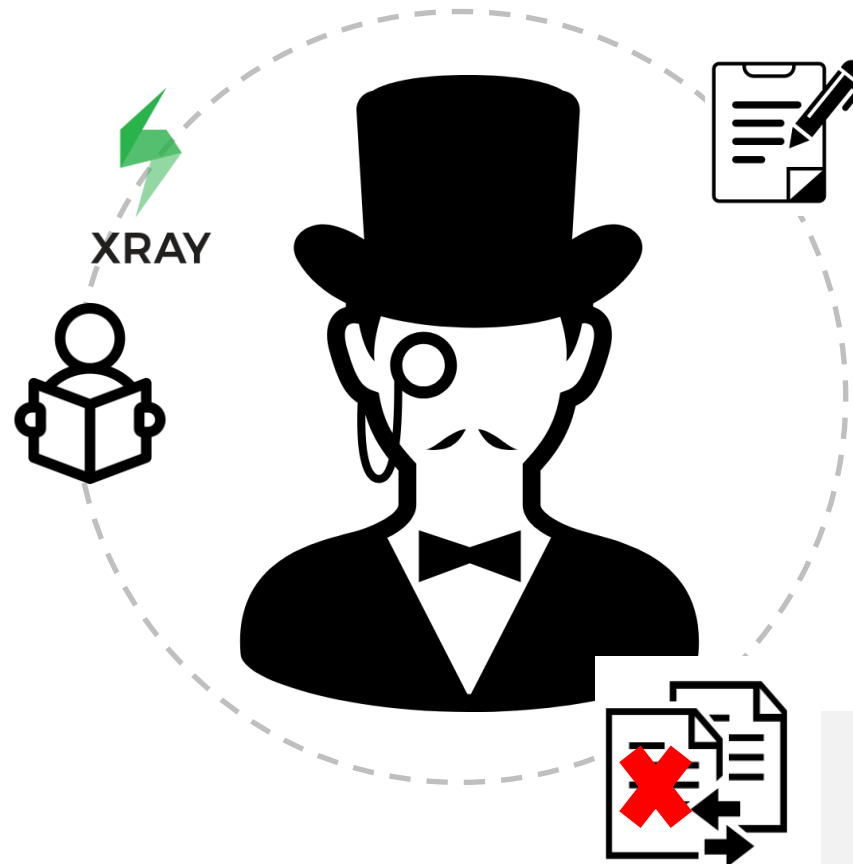
- Création de scripts automatisés

L'application pour accélérer la conception des tests grâce à l'IA



L'application de relecture et contrôle de la qualité des fiches de tests.

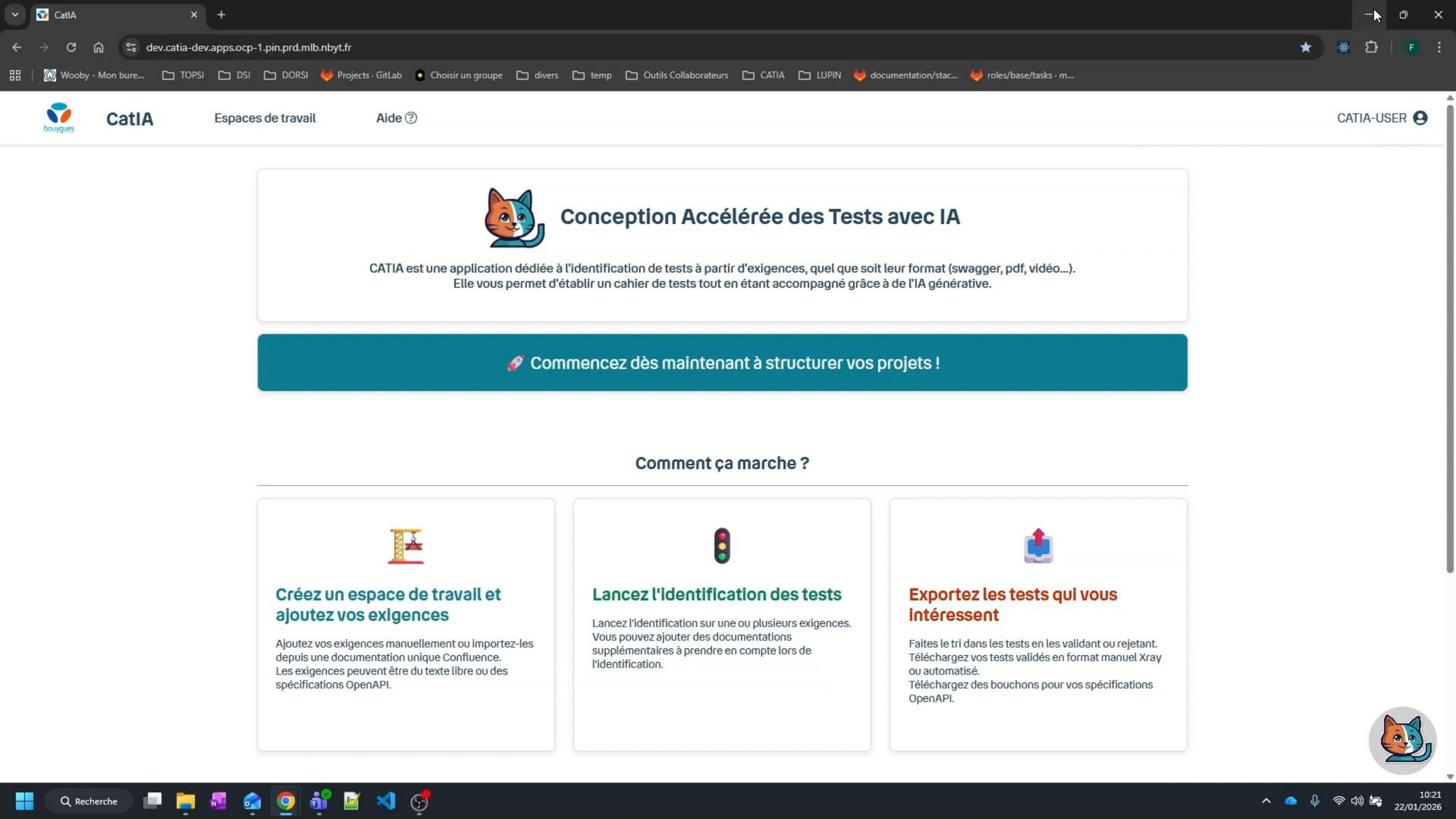
Relecture des
fiches de tests



Réécriture des
fiches de tests

Détection de
doublons





Conception Accélérée des Tests avec IA

CATIA est une application dédiée à l'identification de tests à partir d'exigences, quel que soit leur format (swagger, pdf, vidéo...). Elle vous permet d'établir un cahier de tests tout en étant accompagné grâce à de l'IA générative.

 Commencez dès maintenant à structurer vos projets !

Comment ça marche ?



Créez un espace de travail et ajoutez vos exigences

Ajoutez vos exigences manuellement ou importez-les depuis une documentation unique Confluence. Les exigences peuvent être du texte libre ou des spécifications OpenAPI.



Lancez l'identification des tests

Lancez l'identification sur une ou plusieurs exigences. Vous pouvez ajouter des documentations supplémentaires à prendre en compte lors de l'identification.



Exportez les tests qui vous intéressent

Faites le tri dans les tests en les validant ou rejetant. Téléchargez vos tests validés en format manuel Xray ou automatisé. Téléchargez des bouchons pour vos spécifications OpenAPI.



ENSEIGNEMENTS



Qualité des entrants est
un prérequis



Conduite du changement
indispensable



Expertise métier du
test & IA essentielles



Passage à l'échelle dans la
durée



Supervision humaine
incontournable



Gains a l'échelle
sont a mesurer

28/01/2026

INTERNE

Conception des tests automatisés

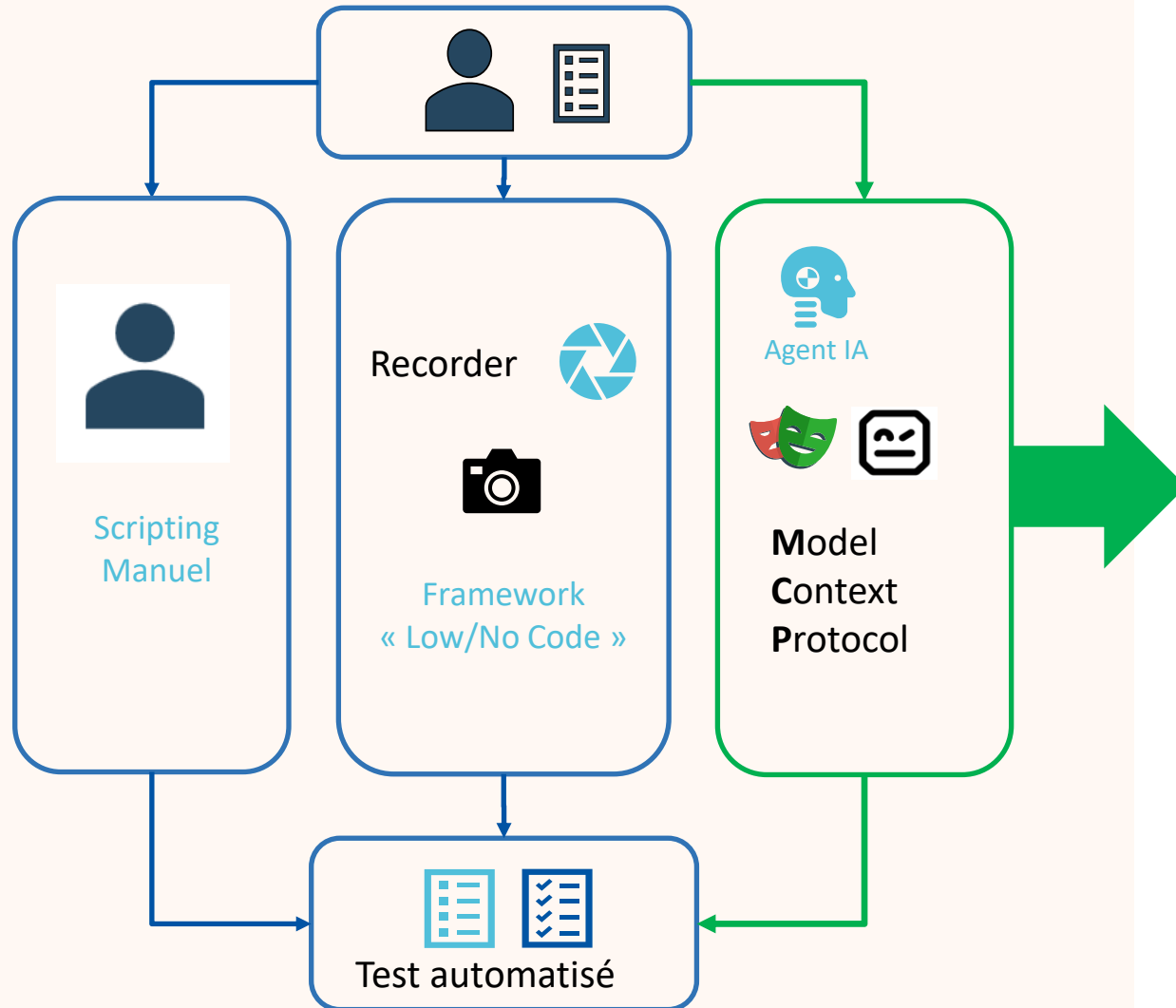
Approche MCP



Conception des tests automatisés

Approche MCP

Automatisation des tests



2 POC (toujours en cours) sur 2 outils MCP

Playwright

Comparer les scripts produits via MCP par rapport au recorder



Test automatisé

Robot Framework

Evaluer les gains sur le chantier d'automatisation



Test manuel



Test automatisé



Test automatisé validé

Premiers Enseignements



Résultats prometteurs dans les
conditions idéales



L'accessibilité
est un facteur de réussite.



Maintenance des tests
automatisés accélérée.



La fiabilité
reste un point faible.