

Et si l'IA rendait l'automatisation inutile ?

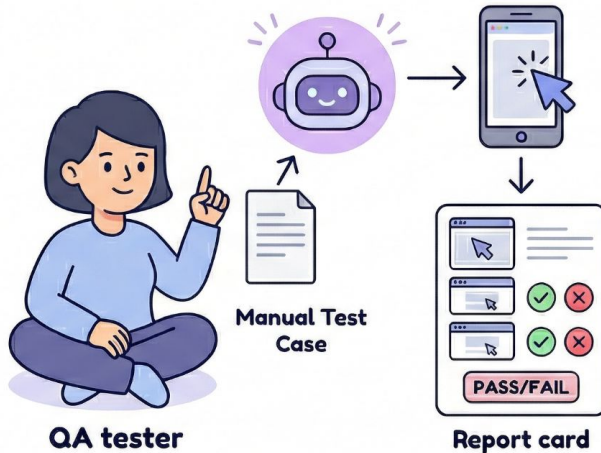
Bruno Legeard - Smartesting



PARITESTCONF

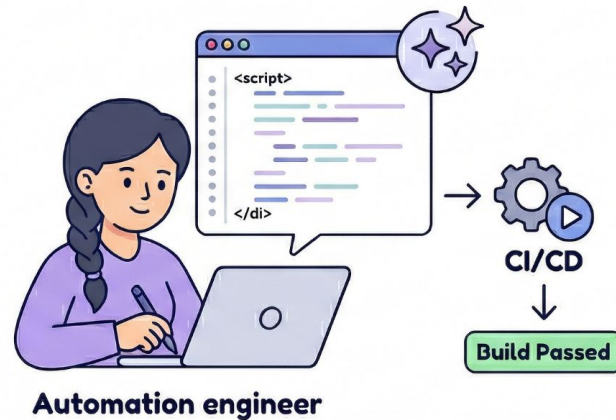
Automatisation avec l'IA : deux voies

Agentic AI Test Execution



VS

AI-Assisted Scripting



Pour démarrer : une petite démo



L'agent web open-source *Browser Use* au travail pour exécuter un scénario Gherkin

Run the following test scenario in Gherkin style on <https://demo.osclasspoint.com/>. At the end, provide a detailed test execution status.

**** Test scenario

Scenario: Perform keyword search

Given I am on the home page

When I enter the keyword "iPhone" in the search field

And I click the search button

Then I should see listings that contain "iPhone" in the title or description

And I should see the total number of listings found

And the results should be sorted by relevance, considering frequency and recency

Summary Table

Step	Gherkin Step	Status
1	Given I am on the detailed search page	✅ PASSED
2	When I enter the keyword "iPhone" in the search field	✅ PASSED
3	And I click the search button	✅ PASSED
4	Then I should see listings containing "iPhone" in title or description	✅ PASSED
5	And I should see the total number of listings found	✅ PASSED
6	And the results should be sorted by relevance (frequency + recency)	⚠️ PARTIAL

Defects / Observations

#	Severity	Observation
1	Low	The default sort on load is "Newly listed", not "Relevance" — users must manually switch to relevance sorting.
2	Low	Listing descriptions contain lorem ipsum placeholder text — the keyword "iPhone" did not appear in descriptions, only in titles.
3	Info	With only 2 demo listings sharing identical dates, relevance algorithm depth (frequency + recency weighting) cannot be fully validated. A production dataset would be required for complete verification.

Browser · Compile and report detailed test execution status 7/7

Petite démo : constats

Cette expérience avec l'agent Browser Use nous apprend que :

- Sans automatisation, sans code, de façon autonome, un agent IA exécute correctement un scénario de test sur une application web
 - Le résultat obtenu est correct : un "FAIL" sur la dernière étape car le tri n'est pas celui attendu → l'agent a détecté l'anomalie
 - L'exécution dure 28 secondes et coûte \$0,08 avec Browser Use.
 - L'agent renvoie des comptes rendus d'exécution pour chaque step, avec les images écran associées et explique pourquoi le résultat est "FAIL" ou "PARTIAL"
- Cette capacité nouvelle de l'IA pose une question : *pourquoi automatiser et maintenir l'automatisation alors que l'IA agentique sait exécuter les tests écrits en langage naturel ?*

Agenda de la présentation

- Agent IA / Agent IA d'exécution des tests : de quoi parlons-nous ?
- Évaluation des agents testeurs et dialogue agent/humain
- Agent IA testeur vs Automatisation (aidée par l'IA)
- Conclusion : mixer agent testeur et automatisation ?

Agents IA GUI - de quoi parlons nous ?

Un agent IA GUI est un système d'IA générative doté de capacités de planification, de perception de son environnement, d'actions sur l'interface du PC/mobile via des outils et d'une mémoire court terme et long terme.

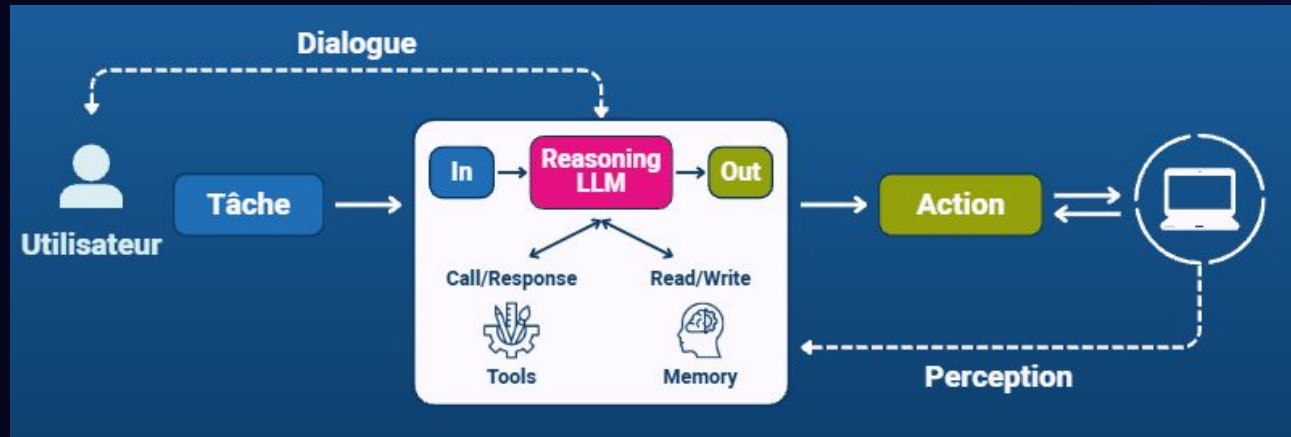


Figure : Schéma d'un système d'agent IA GUI Source : auteur

Agents IA GUI pour exécuter des tests sur l'IHM

Dans cette présentation, nous nous focalisons sur l'exécution des tests fonctionnels sur IHM à partir de scénarios en langage naturel (cas de tests manuels ou scénarios style Gherkin).

Les **agents qui agissent sur l'IHM** sont appelés "Computer Use" ou GUI. Ils utilisent des modèles de vision pour analyser les captures d'écran et décider des actions à réaliser en fonction de la consigne de l'utilisateur.

Les agents GUI gèrent le web, le mobile et les applications du PC (desktop).

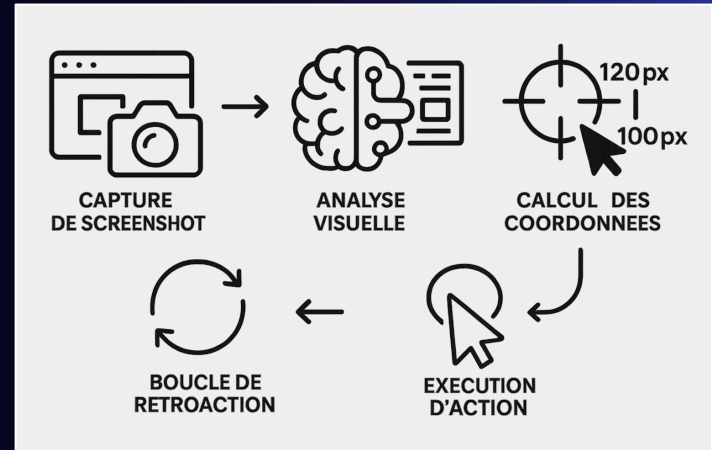


Figure : Schéma d'une étape d'un agent sur IHM
Source : auteur avec IA

Agents IA GUI testeurs : typologie

Pour exécuter des tests sur IHM via Agent IA, on trouve actuellement trois possibilités :

- Les agents GUI généralistes, intégrés dans les grands modèles IA commerciaux ou open-source et disponibles via API ou en chatbot IA.
- Les plateformes agentiques de test qui combinent la conception et l'exécution.
- Les agents GUI testeurs spécialisés dans l'exécution des tests manuels et en Gherkin, intégrés dans l'outil de gestion des tests.

Agents IA testeurs : apport

Caractéristiques des agents IA testeurs :

- Sans script ni capture & replay
- Langage naturel ou Gherkin
- Sans changement ni retouche des tests, sans implémentation ni code
- Permet de lancer des exécutions multiples en parallèle
- Preuves et un reporting clairs
- Intégrés au référentiel de test



⇒ courte démo de Lynqa for Jira ici





Agenda de la présentation


- Agent IA / Agent IA d'exécution des tests : de quoi parlons-nous ?
- **Évaluation des agents testeurs et dialogue agent/humain**
- Agent IA testeur vs Automatisation (aidée par l'IA)
- Conclusion : mixer agent testeur et automatisation ?

Évaluer les agents testeurs : Métriques (1/2)

Objectif : vérifier la cohérence entre les verdicts de l'agent testeur et ceux du testeur humain.

Cas possibles :




-  Vrai FAIL : le test est identifié comme FAIL par l'agent testeur et par le testeur humain.
-  Vrai PASS : le test est identifié comme PASS par l'agent testeur et par le testeur humain.
-  Faux FAIL : l'agent testeur indique FAIL, alors que le testeur humain considère le test PASS.
-  Faux PASS : l'agent testeur indique PASS, alors que le testeur humain considère le test FAIL.

 Ces écarts permettent d'évaluer la fiabilité du jugement de l'agent testeur face à la référence humaine.

Évaluer les agents testeurs : Métriques (2/2)

Objectif : mesurer la précision et la fiabilité des verdicts produits par l'agent testeur.

Métriques principales :

-  **Exactitude** = $(\text{Vrai FAIL} + \text{Vrai PASS}) / \text{Total des cas de test}$
→ Pourcentage de verdicts corrects par rapport à l'évaluation humaine.
-  **Détection correcte des FAIL**
= $\text{Vrai FAIL} / (\text{Vrai FAIL} + \text{Faux PASS})$
→ Capacité de l'agent testeur à bien identifier les tests réellement en échec.
-  **Détection correcte des PASS**
= $\text{Vrai PASS} / (\text{Vrai PASS} + \text{Faux FAIL})$
→ Capacité de l'agent testeur à reconnaître correctement les tests réellement réussis.

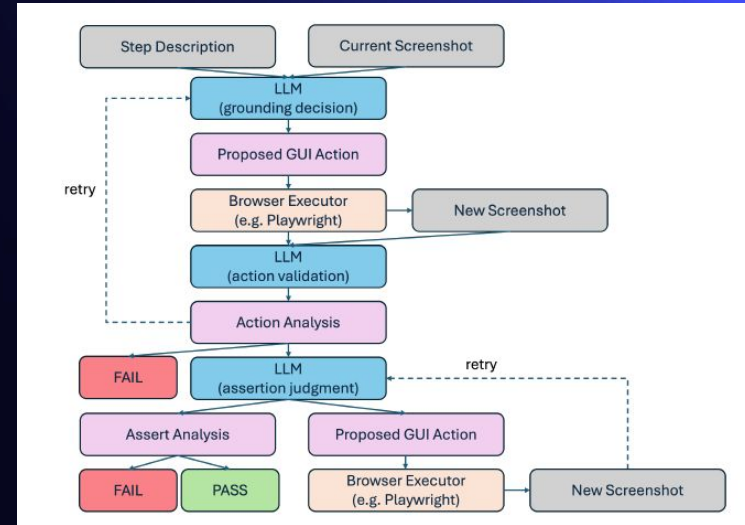
Évaluer les agents testeurs : Résultats

Sur benchmark de > 150 scénarios représentatifs sur >30 applications web :

> 85% de résultats d'exécutions totalement correcte

Dans les 15% restants :

- incapacités (l'agent manque d'outils) → 3 % des cas
- 12% - incertitude de l'agent → variabilité → Dialogue Agent / Testeur QA → acquisition de connaissances



Article Soumis ISSTA 2026 - International Symposium on Software Testing and Analysis - Collaboration Lab IA SMA - LaBRI - Femto-st

Dialogue Agent-Humain & Acquisition de connaissances

Un exemple de connaissance ajoutée lorsque l'agent testeur requête l'humain pour résoudre une incertitude dans l'interprétation du scénario de test

The screenshot displays a testing interface with three main sections:

- Execution Steps:** A vertical list of five steps. Step 4 is highlighted with a green checkmark and a 'Needs Review' label. The overall progress bar shows 3 passed and 2 failed steps.
- Run Details:** Shows a 'Run Failed' status with a 'Rerun Scenario' button. The target URL is 'https://cerberus-testing.com/'.
- Feedback Panel (Step 4):** Contains a 'Knowledge' tab with a text box: 'This tab shows saved knowledge items for the scenario. Pending proposals are reviewed in the Requests tab.' Below this, a section titled 'CREATED DURING THIS RUN' shows a knowledge item: 'verifying feature descriptions claim to be current or up-to-date'. The item includes a 'Why?' dropdown and buttons for 'Go to Step 2' and 'Open Origin Request'.

Step 4 Details:

ACTION	EXPECTED RESULT
Check for any claims made about the product (e.g., performance, compatibility, pricing). Verify these claims against reliable sources or product documentation. Confirm that edition-specific claims (e.g., 'FREE FOREVER' for community editions) are clearly labeled and do not misrepresent the pricing	All claims made about the product are accurate and substantiated. Claims about specific product editions (e.g., 'FREE FOREVER' for community/self-hosted) are clearly attributed to those editions. Paid tiers are not misrepresented as free. There are no false or misleading statements about product offerings overall.

Dialogue Agent-Humain

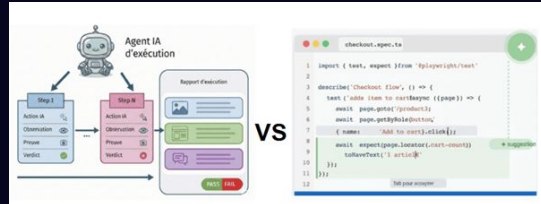
Synthèse :

- L'agent IA testeur ressemble à un testeur humain externe au projet, qui sait exécuter un test, mais ne connaît pas le métier
- Sur certains tests, cela crée des incertitudes pour l'agent, qui sont détectables par un agent secondaire qui lance alors un dialogue
- La séquence de dialogue conduit soit :
 - à une création d'un item de connaissance (au niveau du scénario de test, de la suite de test ou du système testé - au choix du testeur humain).
 - à la suggestion d'une reformulation de l'action ou du résultat attendu de l'étape incertaine, que le testeur humain peut accepter / refuser.

Agenda de la présentation

- Agent IA / Agent IA d'exécution des tests : de quoi parlons-nous ?
- Fiabilité des agents testeurs
- **Agent IA testeur vs Automatisation (aidée par l'IA)**
- Conclusion : mixer agent testeur et automatisation ?

Deux approches, deux forces

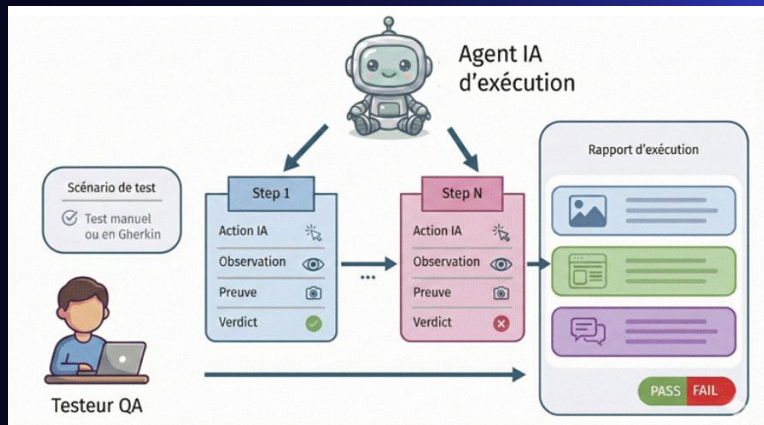


L'automatisation des tests par l'IA : la synergie entre agent testeur et assistant de scripting.

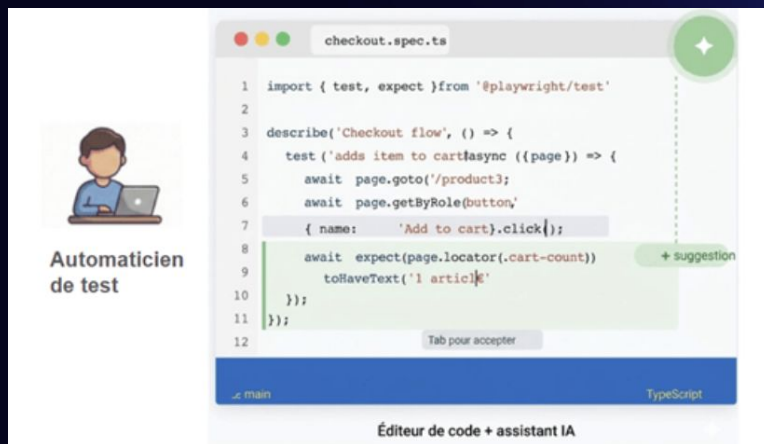
Critère	Agent IA visuel d'exécution	IA assistant de Scripting
Profil utilisateur	QA fonctionnel, analyste de test	Automaticien, développeur QA
Point de départ	Cas de test manuel ou scénario Gherkin à exécuter	Cas de test ou user story à automatiser
Compétence requise	Expertise fonctionnelle et conception de tests	Maîtrise d'un framework (Playwright, Cypress...)
Résultat produit	Rapport d'exécution avec captures d'écran et analyse des vérifications	Script de test maintenable dans un framework d'automatisation scriptée
Maintenance	Portée par l'agent (adaptatif)	Portée par l'automaticien avec aide de l'IA
Scalabilité	Bonne pour les tests fonctionnels et de sprint	Excellente pour la régression stable
Intégration CI/CD	Possible, mais moins native	Native et recommandée
Valeur principale	Exécution native IA, sans script ni code, et disponibilité des agents	Robustesse, performance, réutilisation, habitude d'automatisation scriptée

Complémentarité

Phase Sprint, nouvelles fonctionnalités \Rightarrow Agent IA testeur



Industrialisation des tests de régression \Rightarrow scripting assisté par l'IA



Comparatif – résumé

Critère	Agent IA Testeur	Automatisation scriptée
Mise en route	Immédiate (prompt = Gherkin)	Code/locators (IA aide)
Déterminisme	Variable (stochastique)	Fort (mêmes steps → mêmes résultats)
Robustesse UI	Haute (perception visuelle)	Moyenne (maintenance locators)
Vitesse d'exécution	Lente (captures/raisonnement)	Rapide, massivement parallèle
Traçabilité	Explications + screenshots	Logs + steps + coverage
Coût	Coût IA	Coût automatisation et maintenance
Données/Envs	Doit être piloté	Fixtures, mocks, API
Oracles complexes	Analyse visuelle	Assertions codées
Stabilité CI/CD	Sensible au bruit & variabilité	Très stable avec bonnes pratiques

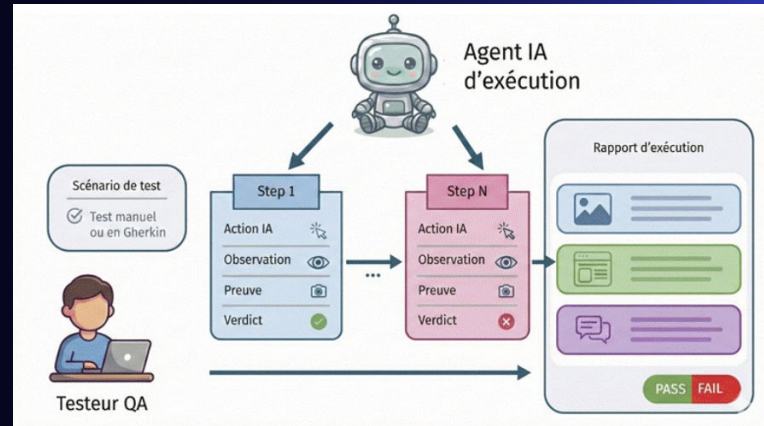
Agenda de la présentation

- Agent IA / Agent IA d'exécution des tests : de quoi parlons-nous ?
- Évaluation des agents testeurs et dialogue agent/humain
- Agent IA testeur vs Automatisation (aidée par l'IA)
- **Conclusion : mixer agent testeur et automatisation ?**

Synthèse : ce qu'on a appris

- Les agents exécutent “sans code” des scénarios en langage naturels et expliquent leurs verdicts (captures + rationale).
- Fiabilité mesurable : définir vrai/faux PASS/FAIL et suivre Exactitude + Détection correcte - actuellement > 85 %.
- Dialogue agent/humain avec acquisition de connaissances / reformulation.
- Deux voies pour automatiser les tests avec l'IA :
 - agents IA = robustesse & explication → utilisateurs : QA - PO
 - scripts générés par IA = vitesse & déterminisme → utilisateurs : automaticiens
- Règle pratique : agent IA pour l'exécution automatique des tests manuels, scripting assisté par l'IA pour l'industrialisation des tests de régression.

Agents IA testeurs et assistants de scripting sont deux voies complémentaires de l'IA.



```
1 import { test, expect } from '@playwright/test'
2
3 describe('Checkout flow', () => {
4   test('adds item to cart', async ({ page }) => {
5     await page.goto('/product3');
6     await page.getByRole('button',
7       { name: 'Add to cart' }).click();
8     await expect(page.locator('.cart-count'))
9       .toHaveText('1 article');
10   });
11 });
```

Automaticien de test

Éditeur de code + assistant IA

Et si l'IA rendait l'automatisation inutile ?

Bruno Legeard - Lab IA Smartesting



PARISTESTCONF