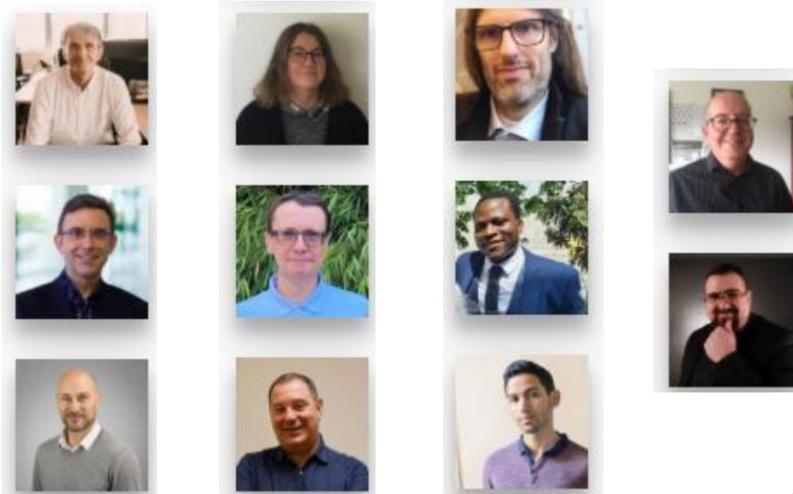


Groupe de Travail sur l'IA du CFTL

Cas d'usage de l'IA pour les tests :
Exploration et mise en pratique

16 JUIN 2025
BEFFROI DE MONTROUGE



JOURNÉE
TUTORIELS
IETL 2025

Introduction

Bruno Legeard



Groupe de travail du CFTL

GT TAIA

“Tester avec l’IA générative”

Contacts

- bruno.legeardcftl@gmail.com
- etienne.dufour@axa.fr
- michael@nocode-testing.com

- **JFTL - Tutoriel Cas d’usage de l’IA**
 - Notre tutoriel aujourd’hui avec les ressources en open-source

- **Enquête IA annuelle - Novembre**
 - IA dans les organisations et pratiques de test
 - Résultats 2024

<https://cftl.fr/cf-tl/ressources/>

3

- **Journée Thématique IA (JTIA)**
 - Retours d’expérience
 - Ateliers de pratique
 - <https://cftl.fr/actualites/jtia-2/>

TAIA - Tester avec l’IA

Groupe de travail du CFTL sur l’IA et le test

Accéder à la documentation

Accéder aux cas d’usages

TUTORIEL JFTL 2025

ENQUÊTE
CFTL
IA
2025

SAVE THE DATE

JEUDI
11
DÉCEMBRE
2025

AGENDA

1. Introduction

Groupe de travail IA du CFTL - Agenda du tutoriel

2. Cas d'usage et mise en pratique

Fiches cas d'usage et workflow IA - Présentation des ateliers

3. Atelier 1 - Analyse des user stories

4. Atelier 2 - Génération de cas de test et optimisation de la couverture

5. Atelier 3 - Génération de tests automatisés

6. Discussion - Echanges

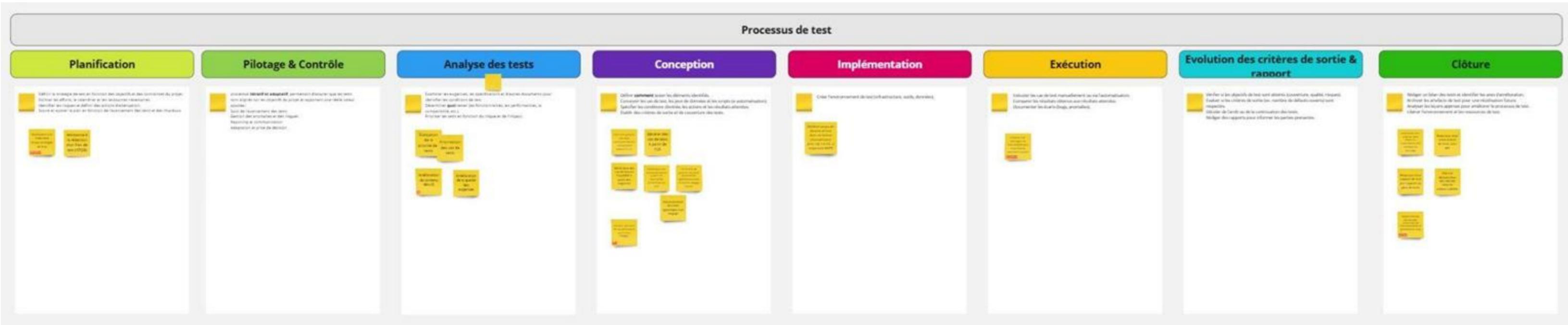
Référentiel cas d'usage de l'IA dans le test

Michael Granier (Nocode Testing)

Réflexion...

**Quid de créer un espace de référence
avec des cas d'utilisation de l'IA dans le test
et leurs tâches de prompts ?**

Quels cas d'usages ?



**Atelier de création d'un backlog de cas d'usage
autour des groupes d'activités de tests**

Quelles informations utiles?

Template fiche cas d'usage

Titre / Description :

Problématique(s) traitée(s) :

- Liste

Groupe d'activité de test ISTQB :

- Types de tests :

Métier / Profil :

 Développeurs, Testeurs, Managers de tests, Product Owners...

Bénéfices attendus / constatés (Méthode d'évaluation du ROI):

- Liste
- Evaluation du ROI (KPI)
 - Liste

Risques:

- Liste

Ressources :

- Liste

Références :

Lier à des tâches (prompts modèles)

Titre / Description : Analyse détaillée des résultats des tests automatisés et génération de bilans exploitables pour l'amélioration continue

Problématique(s) traitée(s) :

- Interpréter les résultats des tests d'exécution automatisés de manière synthétique
- Identifier rapidement les tendances, anomalies et zones de risque grâce à une approche historique
- Permettre de générer des rapports adaptés aux différents profils (managers, développeurs, testeurs) à partir de prompts ou de jeux de données structurés

Groupe d'activité de test ISTQB :

- Activités de tests
- Types de tests : Tests fonctionnels automatisés (pour besoins de tests)

Métier / Profil :

 Développeurs, Testeurs, Managers de tests, Product Owners...

Bénéfices attendus / constatés (Méthode d'évaluation du ROI):

- Analyse automatisée des métriques de tests (taux de succès, erreurs, etc.)
- Réduction du temps d'analyse des résultats des tests
- Informations plus rapides et fiables sur l'état des tests
- Amélioration continue grâce à l'identification rapide des tendances et des anomalies
- Évaluation du ROI (KPI)
 - Temps d'analyse réduit
 - Temps de résolution des problèmes / anomalies réduits par les exigences spécifiques

Risques:

- Mauvaise interprétation des données et rapports ne tenant pas compte des profils utilisateurs
- Coût de mise à jour de données et de mise à jour de l'IA
- Dépendance excessive à l'IA

Références :

- Historique des tests précédents
 - Qualité des tests et de la génération de tests
- Format de sortie des tests automatisés (rapports de tests automatisés)
 - Qualité des tests automatisés (taux de succès, erreurs, etc.)
 - Qualité de génération des rapports (ex: Allure Report, custom dashboards)
- Format "prompts" et agencement de la solution sous-jacente (ex: prompts, JSON)
- Recommandations ISTQB pour les tests automatisés, rapports de tests

Références : Voir à côté

Lier à des tâches (prompts modèles)

- Interpréter et analyser des logs volumineux de tests automatisés
 - Créer un résumé des anomalies et des résultats
- Génération automatique de rapports d'exécution
 - Graphiques synthétiques (ex: code / échec, durée, couverture)
 - Lien des anomalies observées vers la phase de validation
 - Traçabilité des résultats par rapport à l'historique

Titre / Description : Générer des tests API automatisés à partir d'un Swagger

Problématique(s) traitée(s) :

- Automatiser la création de tests API à partir d'une documentation Swagger afin de réduire le temps de conception des tests
- Assurer la conformité des tests avec la spécification de l'API
- Définir les scénarios de tests pour la mise à jour des services et l'évaluation de l'API
- Améliorer la qualité et la couverture des tests API

Groupe d'activité de test ISTQB :

- Conception des tests
- Exécution des tests

Types de tests :

- Tests API
- Tests de régression

Métier / Profil :

- Développeurs
- Testeurs
- Managers de tests
- Product Owners

Bénéfices attendus / constatés (Méthode d'évaluation du ROI):

- Réduction du temps de mise en place des tests API
- Amélioration de la couverture des tests API grâce à la génération automatique
- Maintien automatisé des tests avec une adaptation rapide aux évolutions de Swagger

Évaluation du ROI (KPI):

- Temps de conception des tests avant l'implémentation
- Taux de couverture des tests API
- Qualité des tests générés (taux de succès, erreurs, etc.)
- Temps de mise à jour des tests lors d'une évolution de l'API

Risques:

- Qualité des tests dépendante de la qualité de Swagger
- Manque de visibilité / traçabilité des tests ne pouvant pas couvrir les cas limites
- Difficulté à gérer des cas complexes de validation, scénarios récurrents

Références :

- Documentation Swagger / OpenAPI
- Outils/frameworks de test API (ex: RestAssured, Cypress, Playwright)
- Exemples de tests automatisés avec Swagger
- Bénéfices pratiques ISTQB pour les tests API

Références :

Lier à des tâches (prompts modèles)

- Extraction automatique des endpoints et des paramètres d'un Swagger
- Génération automatique de tests API avec validation des réponses attendues
- Exécution et reporting des tests API avec analyse des erreurs
- Adaptation des tests aux mises à jour de Swagger
- Intégration avec les pipelines CI/CD pour automatiser l'exécution des tests API

Amélioration des exigences via l'IA générative

Problématique(s) traitée(s) :

- L'IA ne respecte pas les contraintes de qualité de l'exigence (ex: format)
- Améliorer le langage des exigences dans la qualité de l'exigence (ex: clarté, précision)
- Améliorer la valeur des exigences en utilisant le gén IA comme 4ème étage

Groupe d'activité de test ISTQB :

- Analyse des tests
- Conception des tests

Métier / Profil :

- Développeurs
- Product Owners
- Business Analysts
- Test Analysts
- Test Managers

Bénéfices attendus / constatés (Méthode d'évaluation du ROI):

- Plus de tests automatisés créés par l'IA grâce à un prompt (ex: moyen de validation)
- Meilleure qualité des exigences (ex: clarté, précision)
- Réduction du nombre de bugs de par une meilleure compréhension du besoin via l'IA

Risques:

- Accès plus limité à l'IA / perte de confidentialité des données

Références :

- Manuel de bonnes pratiques de rédaction

Références :

- IA Générative

Lier à des tâches (prompts modèles)

- Automatiser la production de tests à partir de Swagger (ex: JSON, rapport en JSON)

Assistance à la rédaction d'une stratégie de test

Problématique(s) traitée(s) :

- Manque de visibilité sur le plan stratégique de l'équipe de test (ex: manque de visibilité sur les risques)
- Quelles sont les meilleures pratiques de test ? Quelles informations doivent contenir ?

Problématique(s) traitée(s) :

- Définir et rédiger une stratégie de test globale en tenant compte des enjeux opérationnels, techniques, métier, exigences réglementaires et non fonctionnelles, ...
- Définir une stratégie de test pragmatique, adaptable et évolutif
- Définir une stratégie de test alignée avec la stratégie de business de l'entreprise (ex: qualité, innovation, etc.)

Groupe d'activité de test ISTQB :

- Planification des tests

Métier / Profil :

- Développeurs
- Product Owners
- Business Analysts
- Test Managers
- Test Analysts
- Chief de projet
- Spécialistes
- Responsable Qualité

Bénéfices attendus / constatés (Méthode d'évaluation du ROI):

- Analyse de risques améliorée
- Réponses aux tests et opérations de test améliorées
- Maintien à jour des tests et des exigences
- Maintien des communications entre les différents parties prenantes (challenge le statut de test)
- Automatisation des tâches répétitives par l'IA (ex: génération de tests)

Risques:

- Stratégie de test non opérationnelle, non pragmatique, non maintenable dans le temps
- Manque de visibilité sur l'état des tests
- Manque de visibilité sur l'état des tests
- Manque de visibilité sur l'état des tests

Références :

- ISO 22719-2

Références :

- ISTQB

Références :

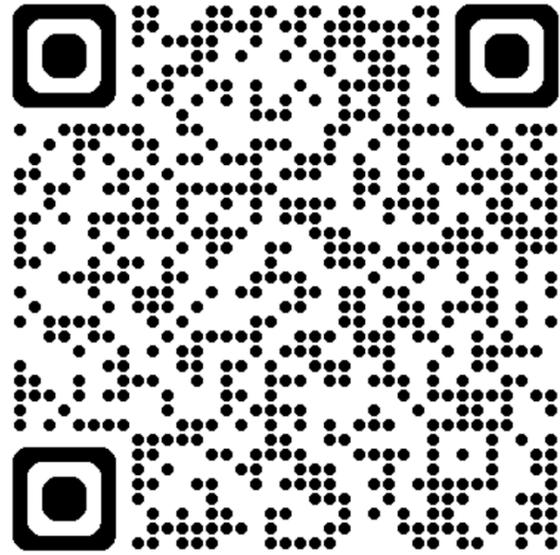
- ISO 22719-2

Lier à des tâches (prompts modèles)

- Production d'un plan de test global pour un projet, exigences métier, non fonctionnelles, etc.

Création d'une fiche simple à remplir pour qualifier les cas d'usages (on peut même la faire remplir par IA :p)

Une plateforme pour partager...



TAIA - Tester avec l'IA

Groupe de travail du CFTL sur l'IA et le test

Accéder à la documentation

Accéder aux cas d'usages

TUTORIEL JFTL 2025



<https://cftl-taia.github.io/>



Présentation des ateliers

Etienne Dufour (AXA France)

Présentation des ateliers

**Analyse des
User Stories**

**Génération de
cas de test
et amélioration de la
couverture de test**

**Génération de
tests automatisés**

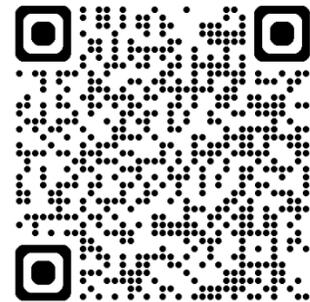
35 minutes par atelier

Introduction et démo : 10 minutes
Mise en pratique : 15 minutes
Discussion : 10
minutes

20 minutes de discussions finales

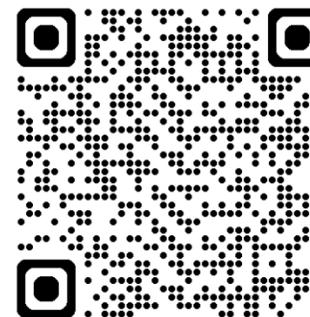
Trois plateformes d'accès LLMs pour nos ateliers

compar:IA



<https://www.comparia.beta.gouv.fr/>

LMArena



<https://lmarena.ai/>

***Instance
Librechat***



[Librechat](#)

AGENDA

1. Introduction

Groupe de travail IA du CFTL - Agenda du tutoriel

2. Cas d'usage et mise en pratique

Fiches cas d'usage et workflow IA - Présentation des ateliers



3. Atelier 1 - Analyse des user stories

4. Atelier 2 - Génération de cas de test et optimisation de la couverture

5. Atelier 3 - Génération de tests automatisés

6. Discussion - Echanges

Atelier 1. Analyse et amélioration des Users Stories

Caroline Distriquin (CEA)

Sarah Leroy (Kereval)

Florent Vaution (Ouest France)

La tâche : Analyser l'US et évaluer le respect des critères qualités

USER STORY

**Référentiels :
BABOK, IREB,
IEEE380, INVEST**

**ANALYSE DE
LA USER
STORY**

**RÉSULTAT DE
L'ANALYSE**

- Critère analysé
- Respect : oui, partiel, non
- Explications

La problématique et les bénéfices attendus

Les problématiques à résoudre

- Difficulté à garantir la qualité des US
- Risque de non-conformité aux standards d'écriture des US entraînant des ambiguïtés, des oublis ou des US non testables
- Charge de relecture et d'analyse manuelle importante pour les PO, Scrum Masters et testeurs.

Les bénéfices attendus

- Amélioration de la qualité et de la clarté des US
- Réduction du temps de revue et de correction des US
- Diminution des défauts liés à des exigences mal définies
- Accélération du cycle de développement grâce à US prêtes à être développées et testées
- Standardisation de l'analyse grâce à l'application systématique de critères objectifs

Proposition de prompt générique

Conduis-toi comme un **[variable1]**.

Je vais te fournir une user story, analyse cette user story selon les critères **[variable2: + liste critères]**.

Pour chaque critère je veux savoir s'il est respecté (oui, partiellement, non) avec une explication.

Le résultat doit être fourni sous la forme d'un tableau contenant les colonnes suivantes :

- Critère **[variable2]** (par exemple **[exemple critère]**)
- Respect : oui, partiel, non
- Explication

Début user story

Fin user story

Les variables

Variable (1) Rôle

Rôle + “du domaine du développement logiciel avec plusieurs années d’expérience dans le contexte Agile Scrum”

Rôles :

- Product owner
- Scrum Master
- Coach Agile
- Ingénieur d’exigences

Variable (2) Qualité

nom critères + liste

- **BABOK** : atomique, complet, cohérent, concis, réalisable, non ambigu, testable, priorisé, compréhensible
- **IEEE830** : correct, non ambigu, complet, cohérent, classé en fonction de son importance et/ou de sa stabilité, vérifiable, modifiable, traçable
- **IREB** : adéquate, nécessaire, non ambigu, complète, compréhensible, vérifiable
- **INVEST** : Indépendante, Négociable, Valeur utilisateur, Estimable, Small, Testable

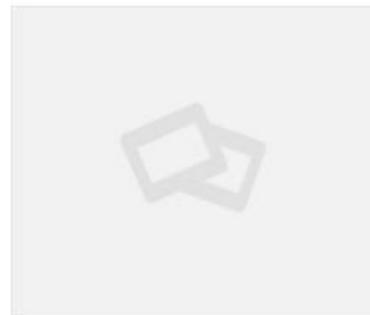
Le site de petites annonces

 **OsClass** Login Publish Ad

What are you looking for today?

Keyword Category Search

Latest Listings



Genius Tech Media – Your ...
Check with seller



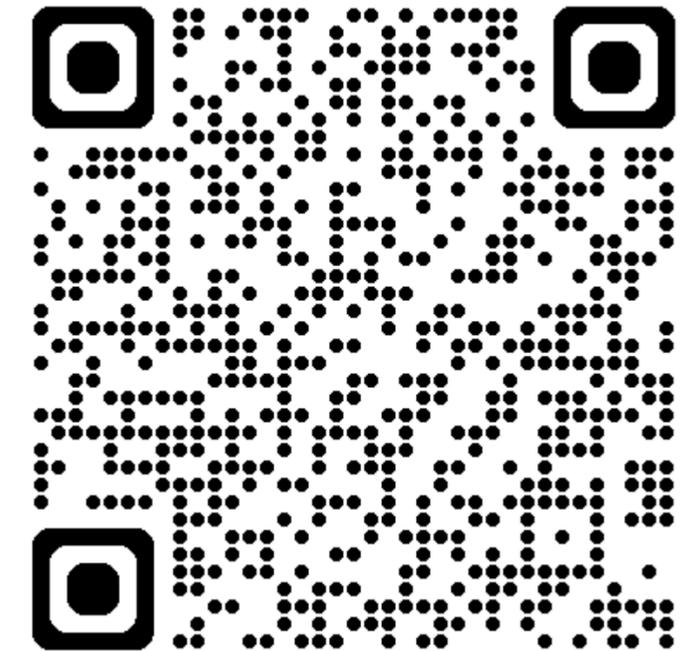
Acer Predator + GeForce
1600.00 \$



Samsung LED 32inch
Free



AlienWare 2019 3.0Ghz 2...
1950.00 \$



<https://demo.osclasspoint.com/en/>

La user story testée

US3 - Publier une annonce

**En tant qu'utilisateur connecté,
Je veux pouvoir publier une annonce
Afin de vendre ou proposer un service.**

Critères d'acceptation (Gherkin)

Scénario: Publication réussie

Étant donné que je suis connecté

Quand je remplis tous les champs obligatoires du formulaire de publication

Et que je valide

Alors l'annonce est publiée avec succès

Scénario: Champs obligatoires manquants

Étant donné que je remplis partiellement le formulaire

Quand j'essaie de valider

Alors un message d'erreur indique les champs manquants

Le Champ Category est obligatoire, si une catégorie n'est pas sélectionnée dans la liste déroulante alors afficher le message d'erreur « Choose one category. ».

Le Champ Title est obligatoire, si le titre n'est pas saisi alors afficher le message d'erreur « Title: this field is required ». Si le titre contient moins de 5 caractères alors afficher le message d'erreur « Title: enter at least 5 characters ». Si le titre contient plus de 100 caractères alors afficher le message d'erreur « Title too long (en_US). ».

Le champ « Description » devient obligatoire si les champs Category et Title sont renseignés ; dans ce cas si la description contient moins de 3 caractères alors afficher le message d'erreur « Description too short (en_US). » et si la description contient plus de 4994 caractères alors afficher le message d'erreur « Description too long (en_US). ».

Scénario: Ajout de photos

Étant donné que je publie une annonce

Quand j'ajoute des photos

Alors je peux en ajouter jusqu'à 4

Si l'image n'est pas au format .png, .gif, .jpg ou .jpeg alors afficher le message d'erreur « You can only upload: image/png, image/gif, image/jpg, image/jpeg ».

Si la taille de l'image dépasse 4096kb alors afficher le message d'erreur « exceeds maximum allowed size of 4 MB ».

Si on tente d'insérer une cinquième image alors afficher le message d'erreur « You cannot upload more images. Image limit reached ! ».

Démonstration Prompt

**A vous de jouer !
Vous avez 15 minutes**

***Parcourez le prompt
Choisissez vos critères qualités
Accédez au Chatbot et lancer le prompt
Analysez et évaluez les résultats***

***Qu'auriez vous fait différemment ? Que peut-on améliorer ?
Préparez l'échange final***

Synthèse

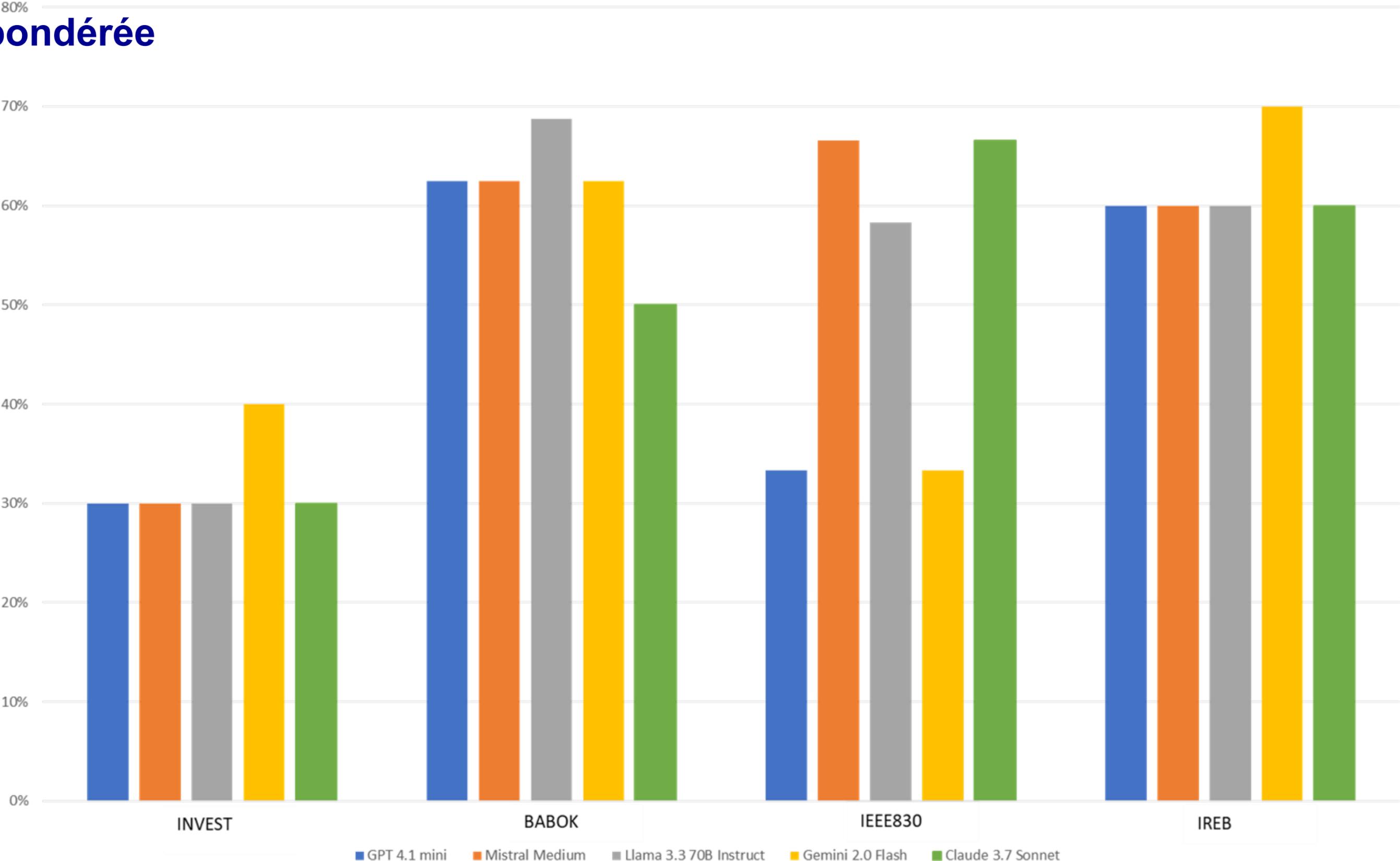
Évaluation

- Métrique initiale : proportion de réponses correctes (accuracy) sur l'analyse du respect des critères
→ Est-ce que le LLM a fourni la même réponse (oui, partiel, non) que l'analyse manuelle ?
- Métrique “pondérée” : pour mieux prendre en compte la spécificité de la réponse “partiel”

	Oui	Partiel	Non
Oui	1	0,5	0
Partiel	0,5	1	0,5
Non	0	0,5	1

Résultats de l'évaluation sur l'US3

Accuracypondérée



Réflexions sur les critères

**Critères non évaluables ⇒
manque de contexte suffisant (projet,
équipe, compétences, dépendances...)**

**Un LLM dit rarement Non ⇒
identification d'éléments manquants**

**Pertinence des critères ⇒
plutôt BABOK mais dépendent du
contexte**

**Une "bonne" US ? ⇒
place à la subjectivité**

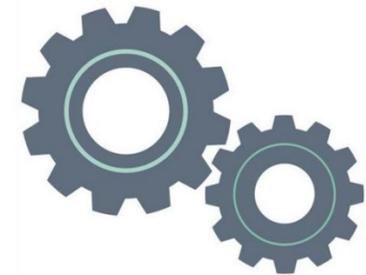
**Proposition de scénarii de tests ⇒
scénarii d'acceptation ?**

Réflexions



Pistes d'amélioration du prompt

- Précisions sur les **attentes** par rapport aux critères
- Proposition d'**amélioration** de l'US
- Vérification des **règles d'écriture** de l'US
- Intégration dans une démarche d'**automatisation** (sortie structurée)



Reste à faire

- Évaluer la pertinence de la variable **rôle** : les premiers résultats montrent que les évaluations sont proches cependant l'explication proposée peut différer

Discussion - Échanges

AGENDA

1. Introduction

Groupe de travail IA du CFTL - Agenda du tutoriel

2. Cas d'usage et mise en pratique

Fiches cas d'usage et workflow IA - Présentation des ateliers

3. Atelier 1 - Analyse des user stories



4. Atelier 2 - Génération de cas de test et optimisation de la couverture

29

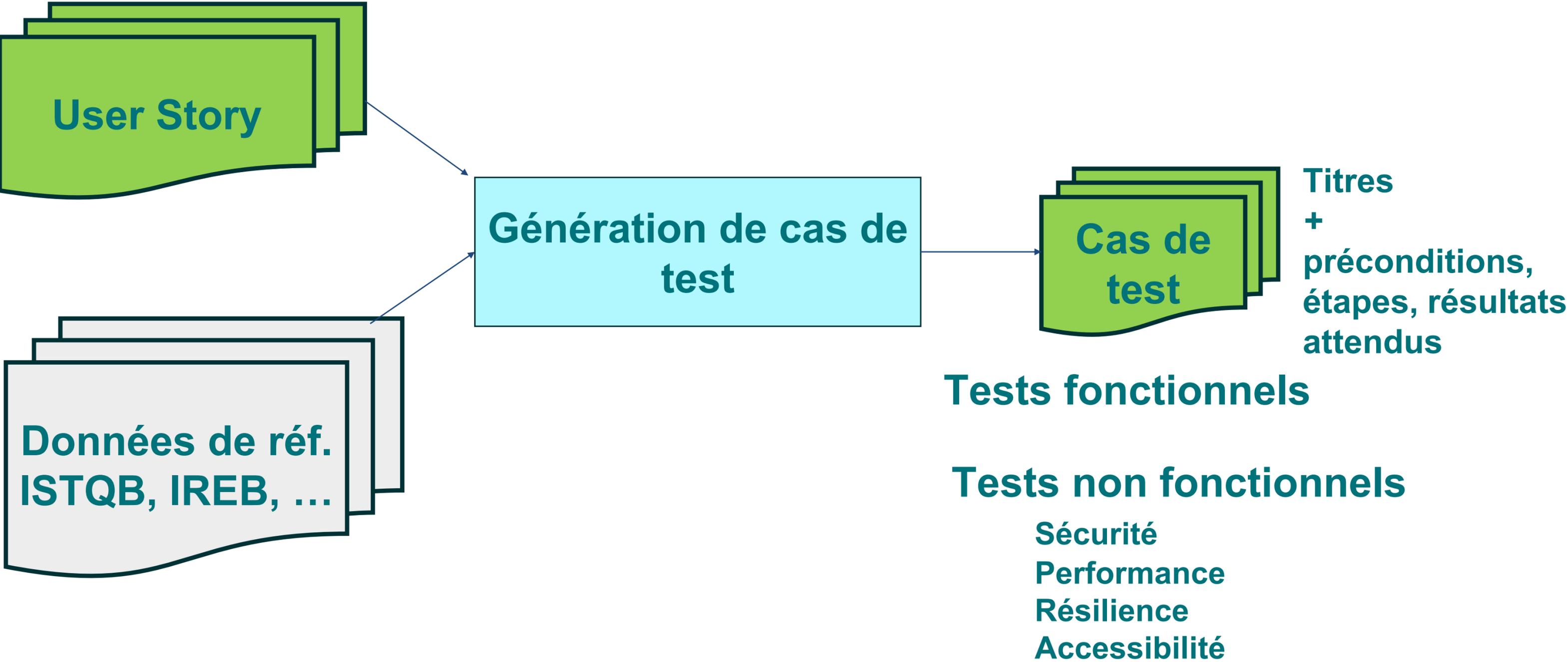
5. Atelier 3 - Génération de tests automatisés

6. Discussion - Echanges

Atelier 2. Génération de cas de test et optimisation de la couverture

Etienne Dufour, David James (AXA France)

La tâche : Génération de cas de test à partir d'une US



Le Prompt simple

*Conduis-toi comme un expert du domaine du test logiciel avec plusieurs années d'expérience dans le contexte agile et dans le monde de l'assurance de biens et de personnes. Adopte une vision critique et disruptive
Inspire-toi du contenu des référentiels ISTQB Fondation, « 610,12-1990 – IEEE Standard Glossary of Software Engineering Terminology » et « IEEE 829-1983 – IEEE Standard for Software Test Documentation » et des critères de qualité préconisés par l'IREB (International Requirements Engineering Board).*

Je te vais te fournir une user story qui est considérée comme bien rédigée par les 3 Amigos.

Voici la user story

< >

Génère les titres de cas de test pour cette user story.

Abandonné car trop simple

Formule les titres selon le format Verbe et complément(s).

Exemples :

. saisir un mot de passe valide

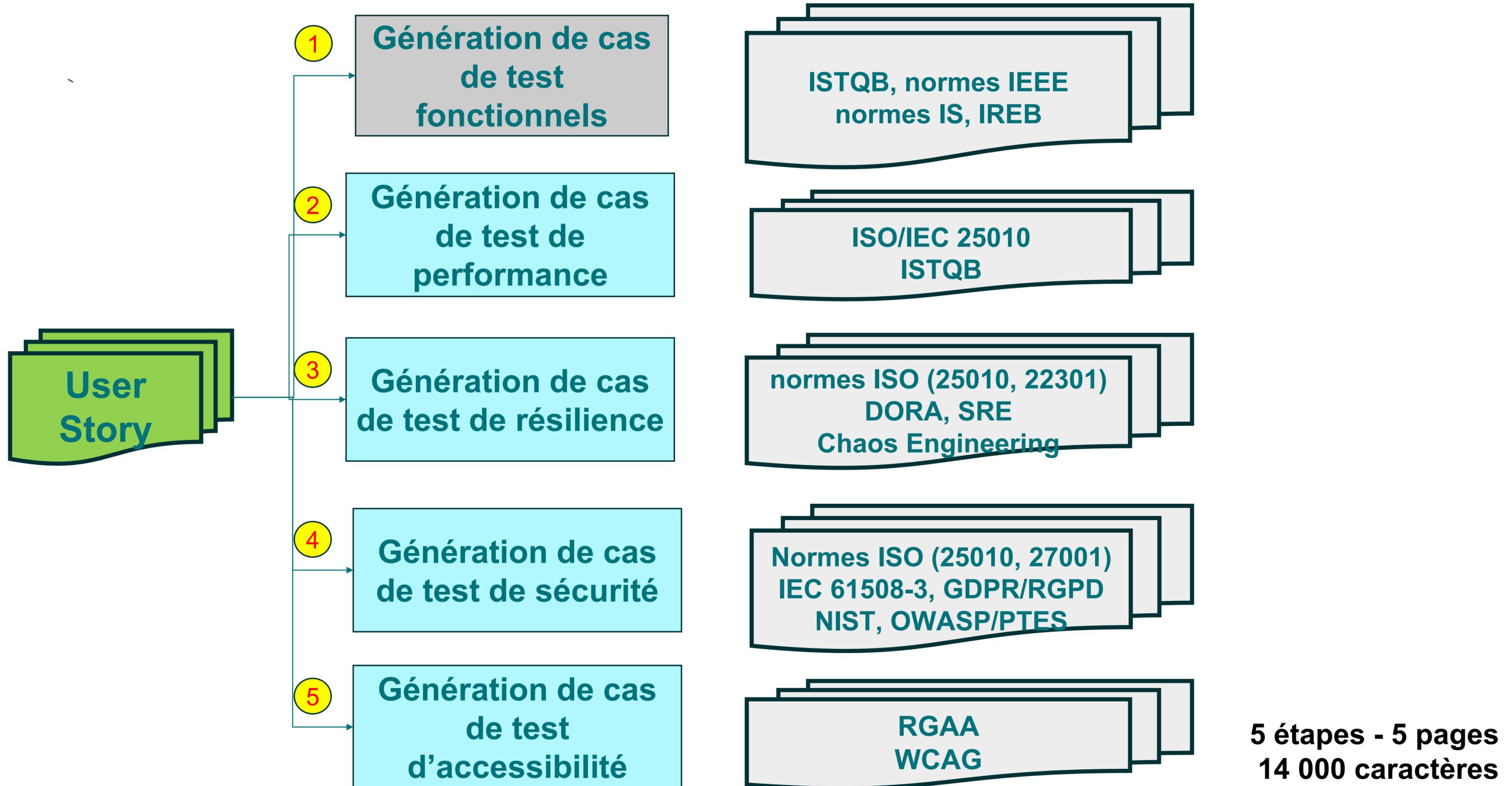
. mesurer le temps de réponse d'affichage

Fournis les résultats sous forme d'un tableau : colonne 1 = compteur de ligne, colonne 2 = titre de cas de test.

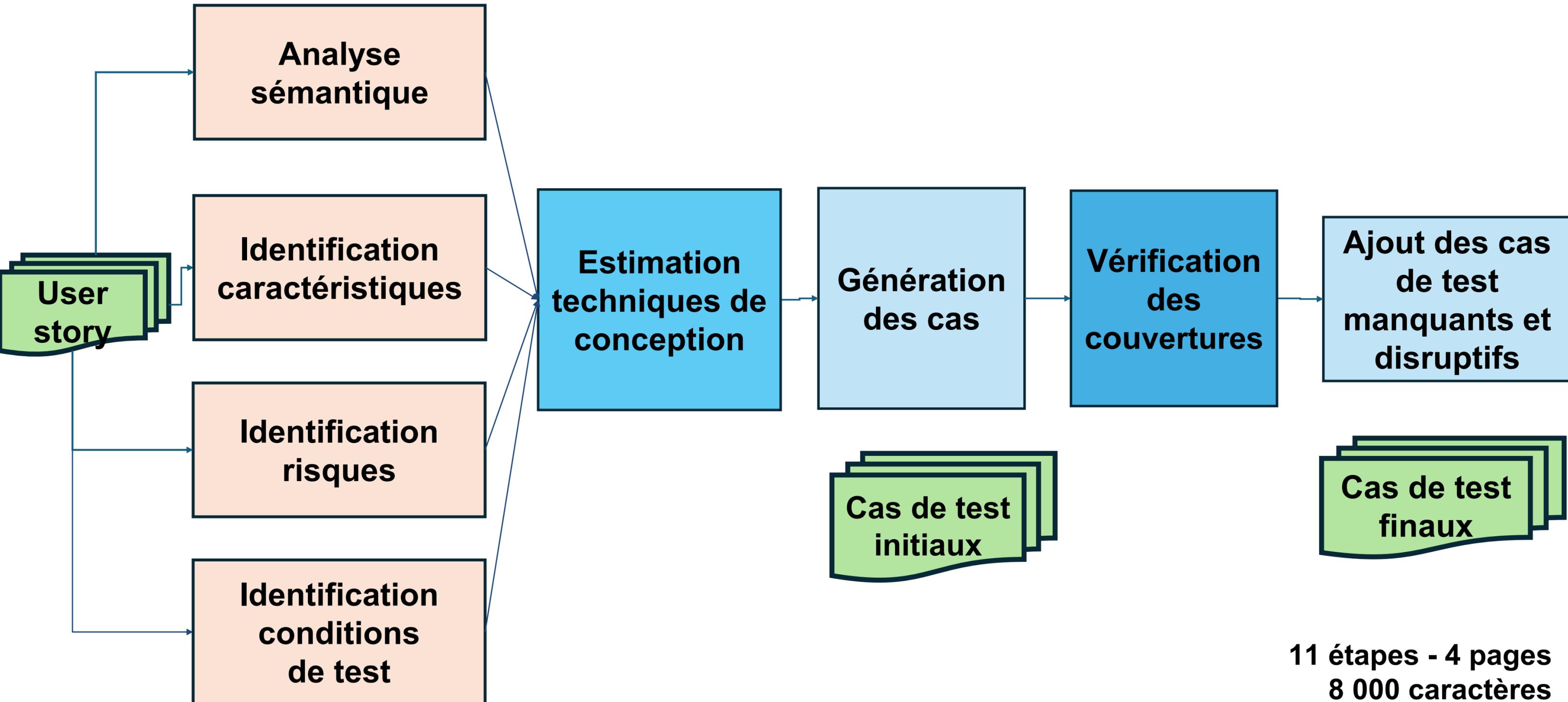
N'ajoute pas de commentaires avant et après tes réponses.

**Une étape - Une demi page
900 caractères**

Le Prompt “analytique”



Le Prompt "dynamique"



11 étapes - 4 pages
8 000 caractères

Démonstration Prompt Dynamique (concis)

 gemini-2.0-flash-001 ▾ vs  gpt-4.1-mini-2025-04-14 ▾

<https://cftl-taia.github.io/docs/TutorielJFTL/US%20Tutoriel>

<https://lmarena.ai/>

**A vous de jouer !
Vous avez 15 minutes**

***Parcourez les 2 prompts
Choisissez celui que vous voulez utiliser
Accédez au Chatbot et lancez le prompt
Analysez et évaluez les résultats***

***Qu'auriez vous fait différemment ? Que peut-on améliorer ?
Préparez l'échange final***

Synthèse

Une stratégie de test complète

Tests quantitatifs 1

Nombre de cas générés

12 LLMs
3 US, 3 prompts

Décisions

LLMs : 12 => 6

Suppression de l'US simple

Tests quantitatifs 2

Nombre de cas générés / attendus

6 LLMs,
2 US, 3 prompts

Décisions

LLMs : 6 => 4

Utilisation de l'US Tutoriel
Suppression du prompt simple.

Tests qualitatifs

Couverture de test fonctionnelle (manuel)

4 LLMs
1 US, 2 prompts

Décisions

4 LLMs validés

Evaluation LLM as judge. Tests multi-itérations

Tests de variabilité

Couverture de test fonctionnelle (LLM) + Multi-itérations

4 LLMs
1 US, 2 prompts

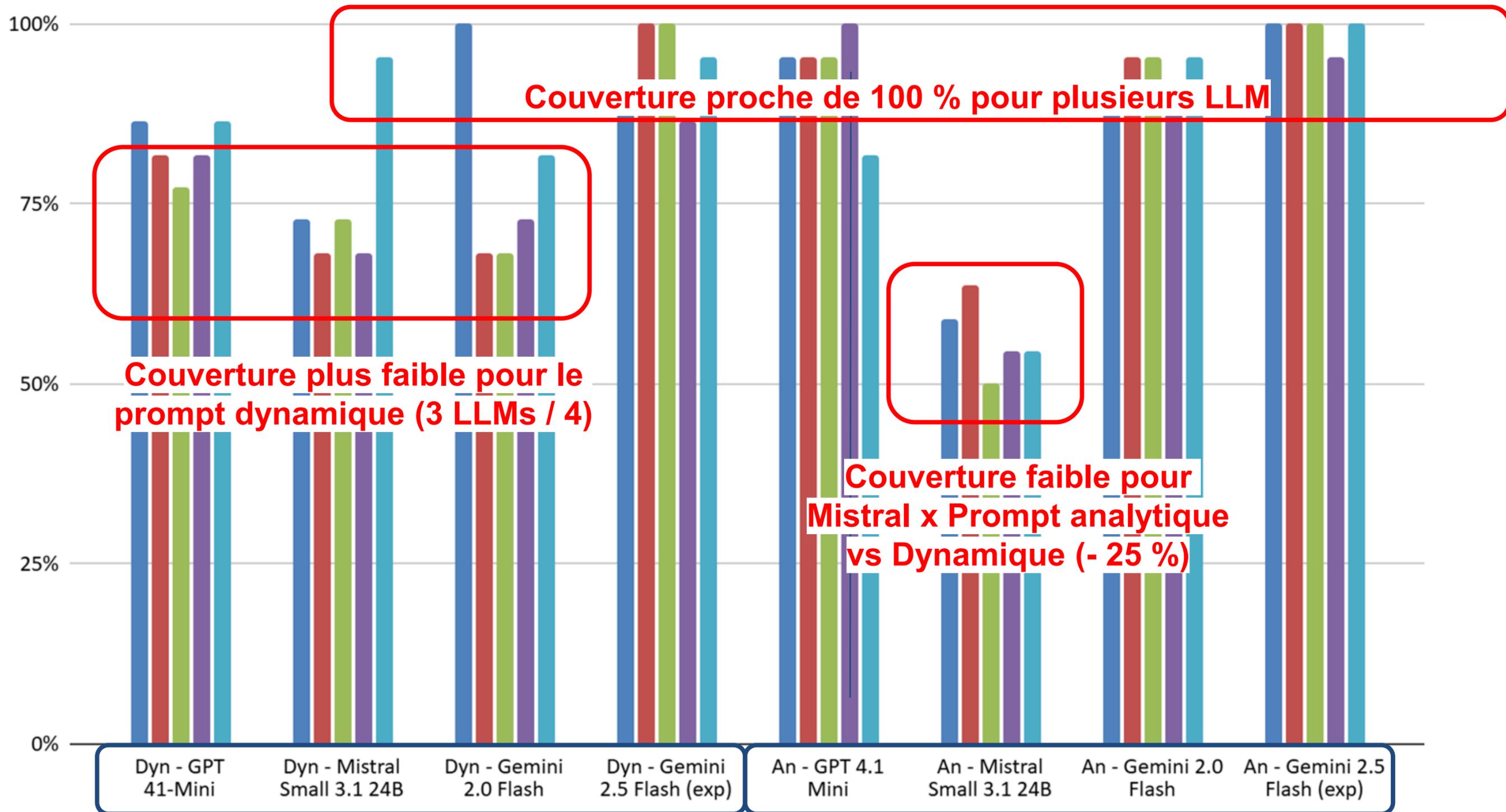
Mesure de conso energie

Wh / CO2

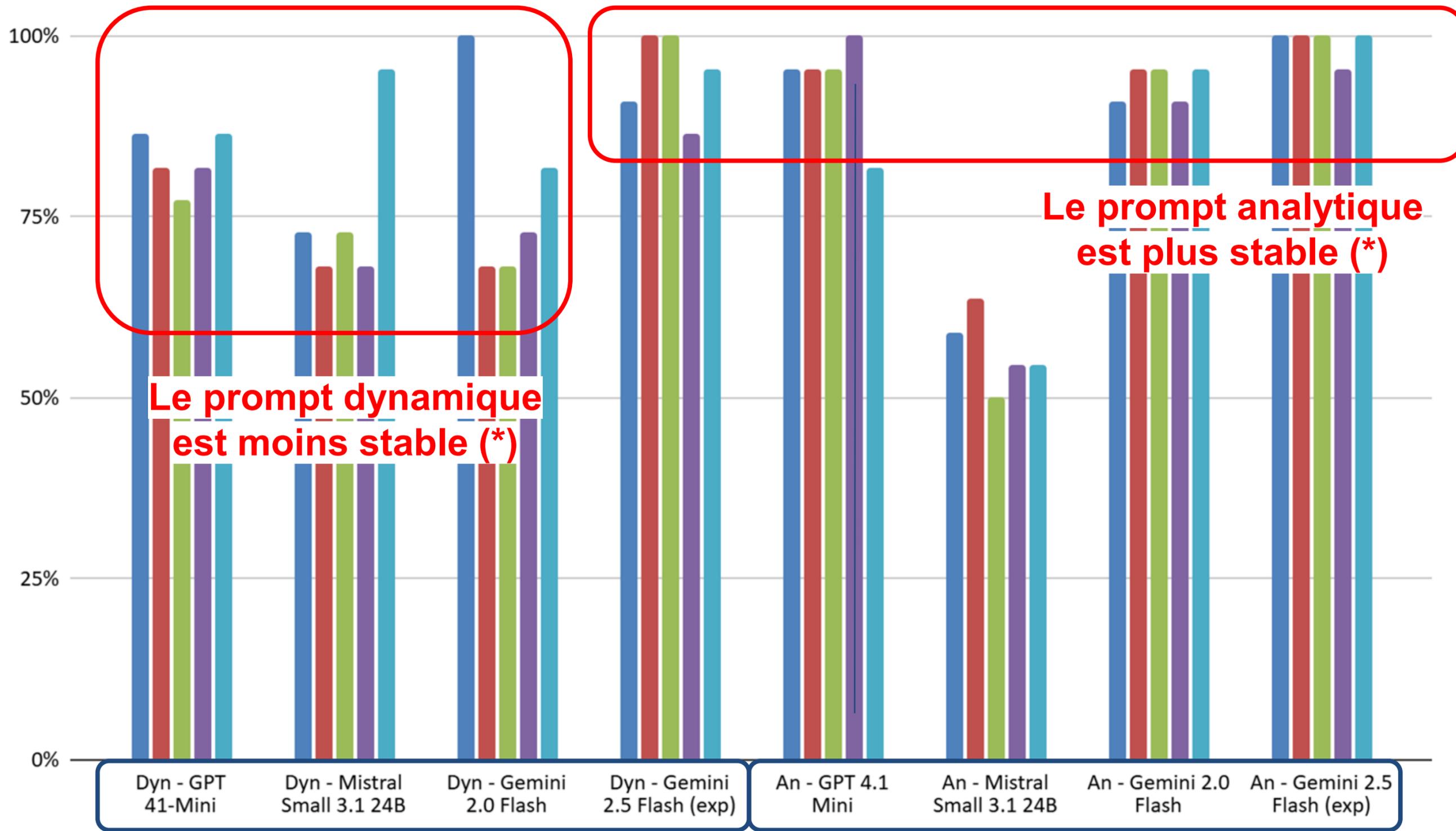
8 LLMs
3 prompts

Amélioration continue des prompts

Couverture de test fonctionnelle et Variabilité



Couverture de test fonctionnelle et Variabilité

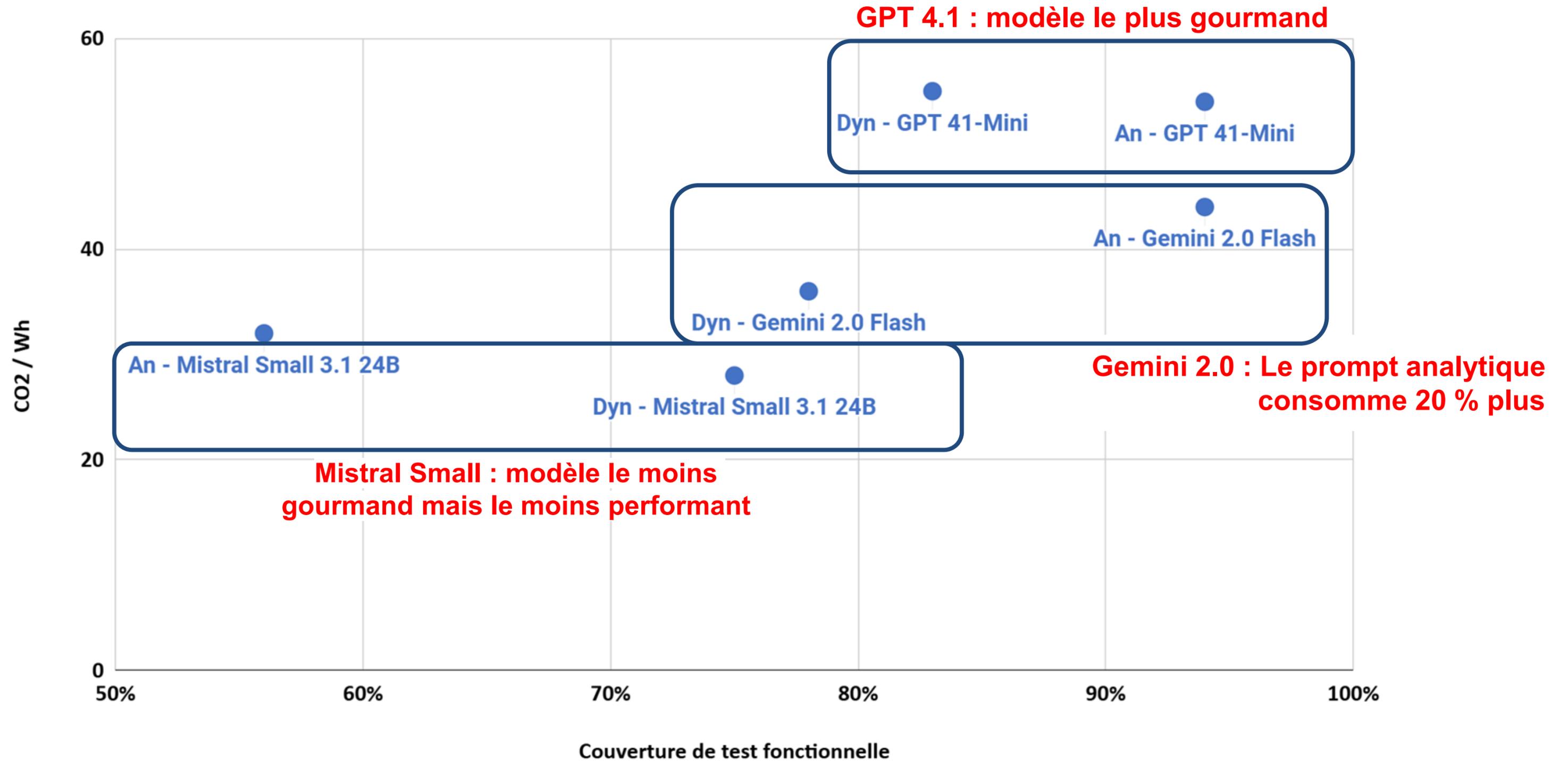


Le prompt dynamique est moins stable (*)

Le prompt analytique est plus stable (*)

(*) les versions initiales des prompts produisaient des résultats TRÈS variables

Consommation en énergie (Wh x Couverture)



Conclusions

- **Les prompts sont des assets informatiques comme les autres. Ils doivent être testés précisément**
- **La qualité des résultats est fonction du prompt ET du LLM**
 - Un prompt simple ne donne pas de bons résultats
 - Les prompts doivent être TRÈS précis et détaillés. *Il faut “contraindre” le LLM*
- **La couverture de test fonctionnelle peut atteindre 100 % (avec des modèles puissants)**
- **La variabilité des réponses doit être mesurée. Elle peut être réduite via les prompts**
- **La consommation CO2/Wh varie d'un facteur 2 selon LLM/prompt**
- **Aucun prompt n'est parfait**

Discussion - Echanges

AGENDA

1. Introduction

Groupe de travail IA du CFTL - Agenda du tutoriel

2. Cas d'usage et mise en pratique

Fiches cas d'usage et workflow IA - Présentation des ateliers

3. Atelier 1 - Analyse des user stories

4. Atelier 2 - Génération de cas de test et optimisation de la couverture



5. Atelier 3 - Génération de tests automatisés

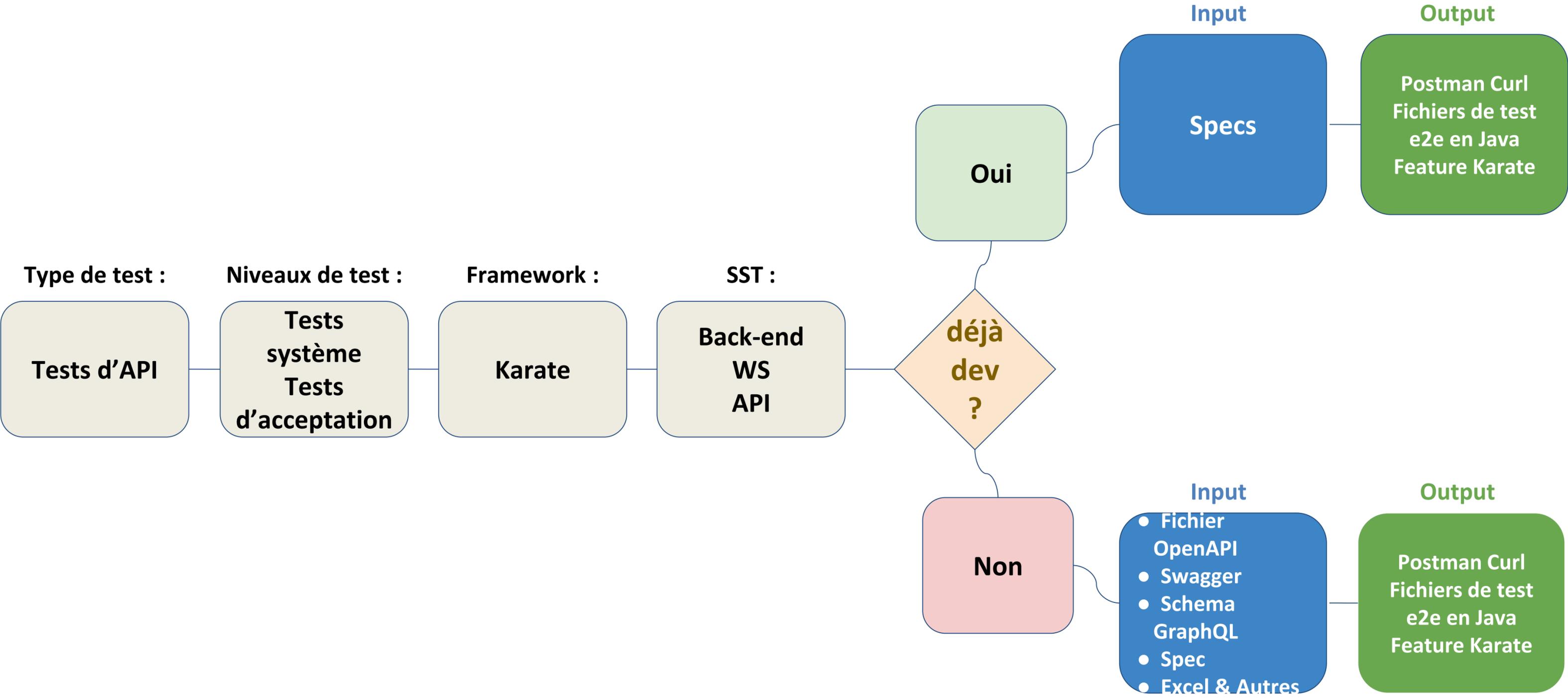
6. Discussion - Echanges

Atelier 3. Génération de tests automatisés

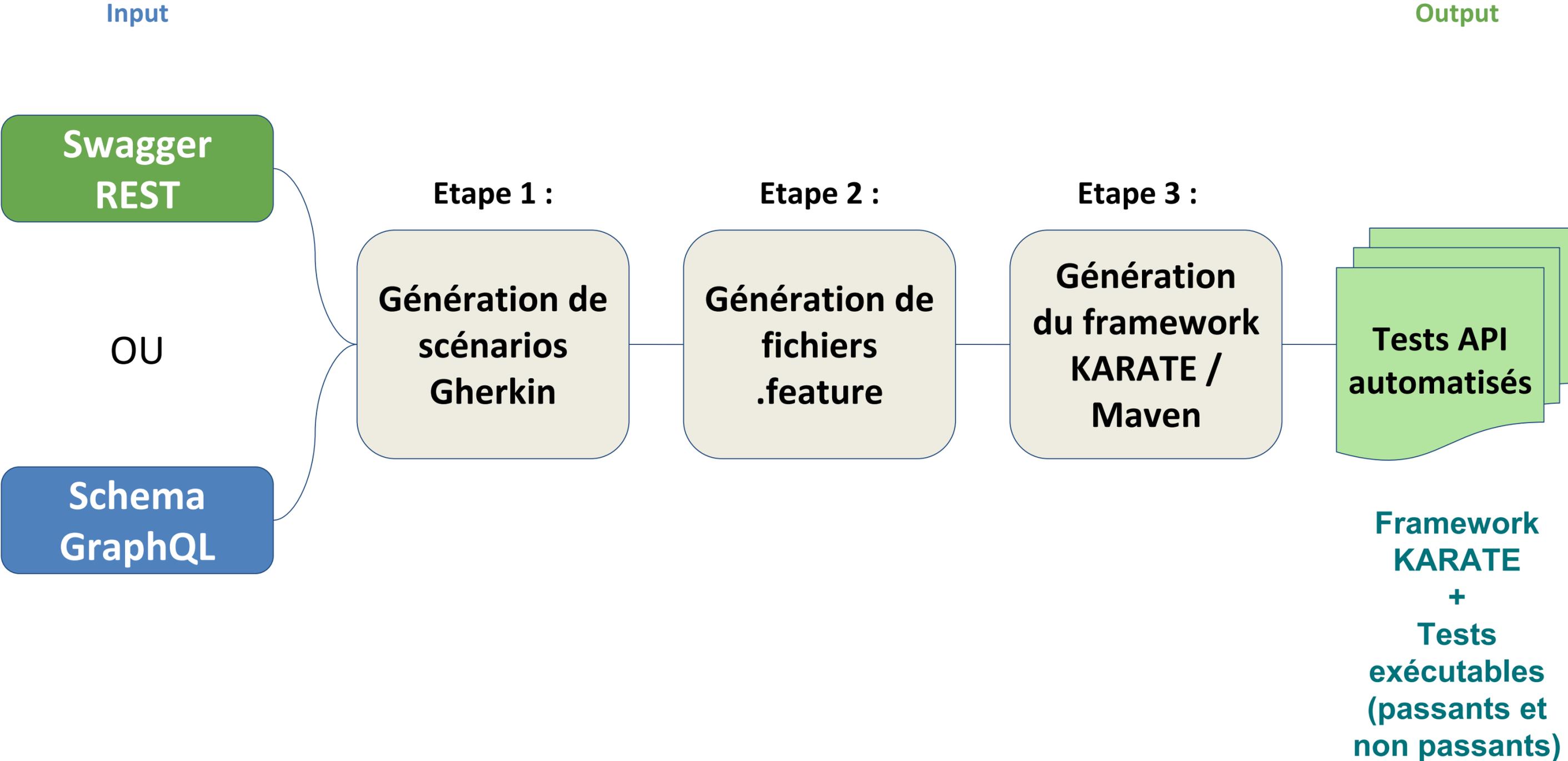
Pierick Njila

Nabil Idhammou

La tâche : Génération de tests automatisés



La tâche : Génération de tests automatisés



Les APIs à tester (inputs)

<https://petstore.swagger.io/#/>

{ REST }

The screenshot shows the Swagger UI for the Petstore API. At the top, the Swagger logo is on the left, and the URL `https://petstore.swagger.io/v2/swagger.json` is in the center, with an `Explore` button on the right. Below the URL, the title `Swagger Petstore` is displayed with version tags `1.0.7` and `OAS 2.0`. A sub-header indicates the base URL: `[Base URL: petstore.swagger.io/v2]` and provides the Swagger JSON link: `https://petstore.swagger.io/v2/swagger.json`. A paragraph of text explains that this is a sample server and provides instructions on how to use the API key `special-key` for authorization. Below this, there are links for `Terms of service`, `Contact the developer`, `Apache 2.0`, and `Find out more about Swagger`. A `Schemes` dropdown menu is set to `HTTPS`, and an `Authorize` button is visible. The main section is titled `pet` with the subtitle `Everything about your Pets` and a `Find out more` link. Three API endpoints are listed: `POST /pet/{petId}/uploadImage` (uploads an image), `POST /pet` (Add a new pet to the store), and `PUT /pet` (Update an existing pet). Each endpoint has a lock icon and a dropdown arrow.

Les APIs à tester (inputs)

<https://countries.trevorblades.com/>



Docs Q ⌘ K

A GraphQL schema provides a root type for each kind of operation.

Root Types

query: **Query**

All Schema Types

- Boolean
- Continent
- ContinentFilterInput
- Country
- CountryFilterInput
- Float
- ID
- Int
- Language
- LanguageFilterInput
- State
- String
- StringQueryOperatorInput
- Subdivision

```
1 query {
2   countries { code name emoji }
3 }
4
```

countries.trevorblades.com

```
{
  "data": {
    "countries": [
      {
        "code": "AD",
        "name": "Andorra",
        "emoji": "AD"
      },
      {
        "code": "AE",
        "name": "United Arab Emirates",
        "emoji": "AE"
      },
      {
        "code": "AF",
        "name": "Afghanistan",
        "emoji": "AF"
      },
      {
        "code": "AG",
        "name": "Antigua and Barbuda",
        "emoji": "AG"
      }
    ]
  }
}
```

HIT 144ms RequestID: 17520101963463782430

IP limit 48 (out of 50) requests remaining (refills in 51s)

Variables Headers

Prompt étape 1

<https://petstore.swagger.io/#/>

{ REST }

Ignore all previous instructions.

You are a world-class test automation engineer with over 25 years of hands-on experience.

Your task is to write comprehensive API test cases in **Gherkin syntax**, based on the Swagger specification located at:

<https://petstore.swagger.io/v2/swagger.json>

Requirements:

- Be precise, thorough, and technically accurate.
- Every test case must have a clear and descriptive **title**.
- Test cases should reflect a **risk-based approach**, prioritizing high-impact functionality.
- Apply the guidance from the following ISO/IEC/IEEE software testing standards:

<https://countries.trevorblades.com/>



Ignore all previous instructions.

You are a world-class test automation engineer with over 25 years of hands-on experience.

Your task is to write comprehensive API test cases in **Gherkin syntax**, based on the GraphQL specifications located at:

<https://countries.trevorblades.com/>

Standard	Focus	Relevance to Automation
29119-1	Concepts	Establishes foundational testing principles
29119-2	Processes	Supports integration of automation into workflows
29119-3	Documentation	Encourages standardized, reusable test artifacts
29119-4	Techniques	Guides selection and implementation of automation
29119-5	Keyword-Driven Testing	Promotes maintainable, framework-driven design

Do not include any introductory or concluding comments. Just return the Gherkin test cases with titles.

**A vous de jouer !
Vous avez 10 minutes**

***Parcourez les 2 prompts
Choisissez celui que vous voulez utiliser
Accédez au Chatbot et lancez le prompt
Analysez et évaluez les résultats***

***Qu'auriez vous fait différemment ? Que peut-on améliorer ?
Préparez l'échange final***

Prompt étape 2

Your task now is to generate **comprehensive Karate feature files** based on the previously defined **Gherkin-style API test cases**.

Requirements:

- Output must be **precise, technically accurate, and production-ready**.
- Each **feature file** must have a **clear, descriptive title** and logically structured scenarios.
- Implement a **risk-based testing approach**, prioritizing **high-impact API functionality**.
- Follow best practices from the **ISO/IEC/IEEE 29119** standards as guidance for quality, structure, and automation design:

Standard	Focus	Relevance to Automation
29119-1	Concepts	Establishes foundational testing principles
29119-2	Processes	Supports integration of automation into workflows
29119-3	Documentation	Encourages standardized, reusable test artifacts
29119-4	Techniques	Guides selection and implementation of automation
29119-5	Keyword-Driven Testing	Promotes maintainable, framework-driven design

Instructions:

- Do not include any commentary, explanation, or formatting outside of the **Karate feature syntax**.

Only return the **Karate feature files**, starting directly with the **Feature:** line.

**A vous de jouer !
Vous avez 10 minutes**

***Parcourez le prompt
Accédez au Chatbot et lancez le prompt
Analysez et évaluez les résultats***

***Qu'auriez vous fait différemment ? Que peut-on améliorer ?
Préparez l'échange final***

Prompt étape 3

Your task now is to generate a **complete test automation project using Karate**, based on the previously defined **Karate feature files**.

Requirements:

- The output must be **precise, technically accurate**, and **ready for execution in a production-grade environment**.
- Include a fully configured `pom.xml` file with **all necessary dependencies and plugins** required to build and run the Karate tests.
- Apply a **risk-based testing strategy**, prioritizing **critical API functionality** that has the highest business or operational impact.
- Structure the project and test logic in alignment with the following **ISO/IEC/IEEE 29119 software testing standards**:

Standard	Focus	Relevance to Automation
29119-1	Concepts	Establishes foundational testing principles
29119-2	Processes	Supports integration of automation into workflows
29119-3	Documentation	Encourages standardized, reusable test artifacts
29119-4	Techniques	Guides selection and implementation of automation
29119-5	Keyword-Driven Testing	Promotes maintainable, framework-driven design

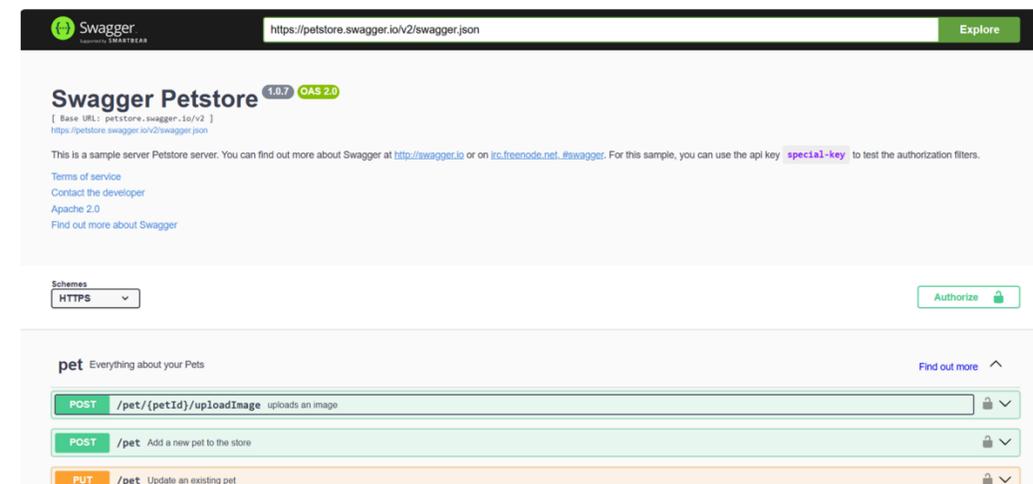
Instructions:

- Do **not** include any commentary, explanation, or descriptive text.
- Only return the **full Karate test automation project**, including folder structure, files, and all required configurations.

Démonstration Prompt Framework Karate

<https://petstore.swagger.io/#/>

{ REST }



Synthèse

Évaluation



Tests autos GraphQL												
	Etape 1				Etape 2				Etape 3			
Noms modèles	Nombre de tests	Cas passants	Cas non passants	Couverture	Nombre de features	Framework fonctionnel ?	Fonctionnel après cc	Nombre de fichier	Cas exécutable:	Erreurs / Commentaires		
GPT-4o	10	8	2	66% (4/6)	10	Oui	Oui	12	4			
Llama 4 Scout	13	9	4	66% (4/6)	1	Oui	Oui	8	13			
Mistral Saba	10	6	4	33% (2/6)	1	Non	Oui	7	10			
DeepSeek V3 (0324)	8	6	2	50% (3/3)	7	Oui	Oui	6	8			
Claude 3.7 Sonnet	14	12	2	66% (4/6)	1	Non	Non	?	0	Framwork incomplet par manque de tokens		
xAI/Grok 3 Mini	14	11	3	83,3%(5/6)	4	Oui	Oui	7	14	Quelques assertions à revoir		
Google/Gemma 3 27B	19	11	8	50% (3/6)	4	Non	Non	6	0	Pas mal de soucis avec la syntaxe des requêtes		

API :	https://countries.trevorblades.com/
Nombre de fonctionnalités	6
Fonctionnalité GraphQL	Description
countries	Liste de tous les pays
country(code: ID!)	Détails d'un pays par code
continents	Liste de tous les continents
continent(code: ID!)	Détails d'un continent par code
languages	Liste de toutes les langues
language(code: ID!)	Détails d'une langue par code

[Résultats - Google Sheets](#)

AGENDA

1. Introduction

Groupe de travail IA du CFTL - Agenda du tutoriel

2. Cas d'usage et mise en pratique

Fiches cas d'usage et workflow IA - Présentation des ateliers

3. Atelier 1 - Analyse des user stories

4. Atelier 2 - Génération de cas de test et optimisation de la couverture

5. Atelier 3 - Génération de tests automatisés



6. Discussion - Echanges

Discussion - Echanges

Discussion - Echanges

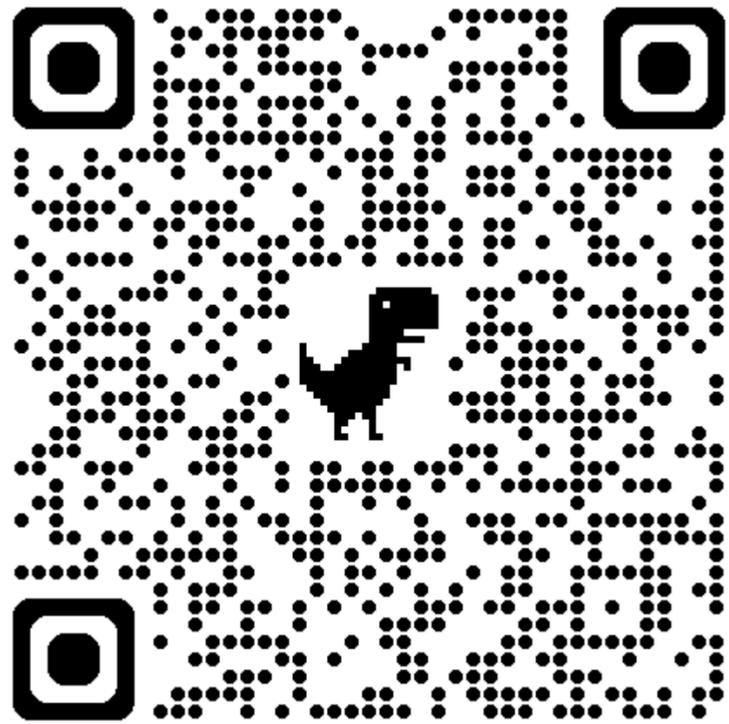
Choix des réponses	Réponses	
Amélioration de la rédaction des user stories, exigences et critères d'acceptation	58,72%	101
Génération de la Stratégie de test / du Plan de test	28,49%	49
Analyse de risques	15,12%	26
Analyse de test / Conditions de test	23,84%	41
Conception de cas de test manuels	56,40%	97
Automatisation des tests (génération / correction du code de test)	60,47%	104
Création / amélioration des rapports de test	20,93%	36
Analyse des anomalies	15,70%	27
Planification des tests	9,30%	16
Nombre total de répondants : 172	298,85%	514

Quels cas d'usage de l'IA dans votre contexte ?

Quels sont vos retours d'expérience ?

Vos retours

CFTL TAIA - Retours sur le tutoriel



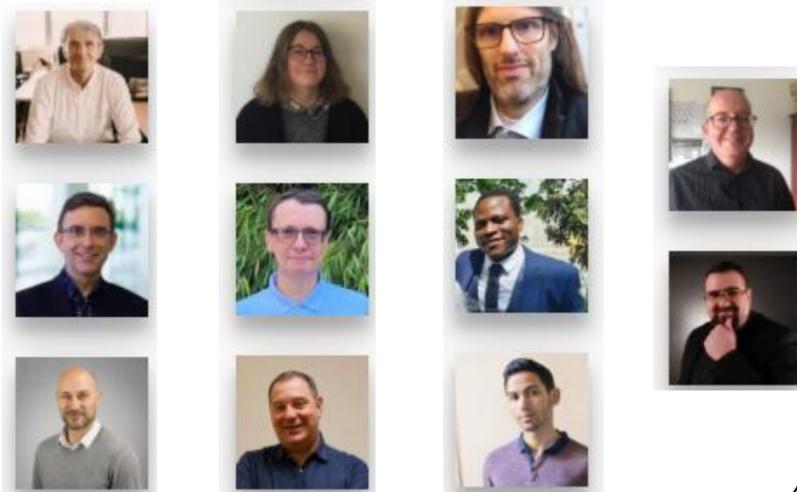
**Merci de répondre à
ce (court) formulaire**



Groupe de Travail sur l'IA du CFTL

Cas d'usage de l'IA pour les tests :
Exploration et mise en pratique

MERCI ET A BIENTÔT



JOURNÉE
TUTORIELS

IETL 2025

Génération de cas de test / Nombre de cas de tests total

