

De l'IA générative aux tests "augmentés" : Les 7 leçons apprises pour industrialiser l'Intelligence Artificielle

Anthony Labarre

Gilles Nouais



BEFFROI DE MONTROUGE



Sommaire

1. Contexte

2. La stratégie et le dispositif

3. Défis Techniques et Leçons Apprises

4. L'Impact Mesuré et la Diversification des Usages

43 ans d'expérience

Editeur de logiciels
ADN 100% Secteur Public

230 Salariés

Actionnariat salarié

20 millions de CA

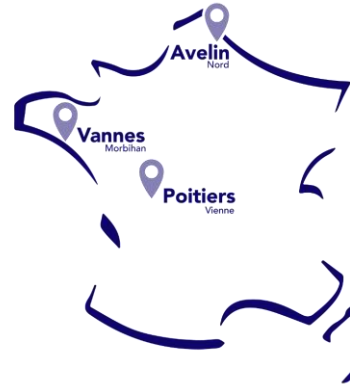
850 clients

Collectivités & établissements de santé

12 produits

+10 000 jours de R&D / an

100% française



Un numérique porteur de sens

- Aides & Subventions versées
- Gestion des Opérations & Financements
- Ecosystème Numérique
- Pilotage financier & stratégique

25 ans d'expertises

+ 1400 talents

+ 110 experts sur la qualité logiciel

12 sites dont Espagne et Maroc



ACCÉLÉRATEUR
DE TRANSFORMATION

- Conseil
- Créatifs
- Technologies
- IA + DATA
- Cybersécurité

Contexte

Contexte & enjeux



- ➔ Equipe de production : 120 personnes
- ➔ 20 testeurs
- ➔ 4 à 6 versions majeures/an
- ➔ 1 patch hebdomadaire

- Ressources précieuses difficile à recruter
- Budgets toujours en tension
- Une production logicielle qui s'accélère
- Temps de conception de test contraint
- Nécessité de multiplier les typologies de test
 - Sécurité
 - Accessibilité
 - Ecoconception numérique...

Demande de la Direction de rechercher des leviers d'optimisation de production

Le début des questions

Par quoi je commence ?
Quel outil ?
Ai-je encore besoin de testeurs ?
C'est quoi une IA ?
C'est quoi vraiment l'état de l'art ?...

L'état de l'art

Journaux, Séminaires, Conférences....
Tous parlent IA, c'est le buzzword !
Des promesses! Des expérimentations !
Sont-elles toutes réelles?

Sans oublié le passé!

« L'automatisation va tuer le test manuel »
« Avec l'agilité, plus besoin de testeur »

Objectifs

Objectifs identifiés

- **Standardiser** les pratiques et valider la **conformité** aux standards existants
- **Augmenter la vélocité** de la conception des tests
- Renforcer la **couverture de test** (tests non fonctionnels, cas spécifiques, usages)
- **Diversifier** nos tests (sécu, charge, RGAA...)
- Augmenter la vélocité de production des **scripts automatisés**
- Alimenter la gestion **des jeux de données**
- Mieux **cibler les défauts** et optimiser les efforts de test

Le rêve !

- Profiter d'un LLM pour générer le texte « **verbeux** »
- Profiter de l'IA pour **imaginer, structurer** nos cas de test
- Eviter le syndrome de la « **page blanche** »

Contraintes

- Outil **simple**, accessible et évolutif
- Outil **branché** sur nos outils de production (API)
 - Avec notre contexte
 - Avec nos méthodes
 - Avec nos logiciels
- Respect de la **Sécurité (ISO 27001)** ,
la **souveraineté des données** ,
la **résilience numérique** ,
l'**Ecoconception numérique** ...

Stratégie et dispositif



Découverte IA

- ✓ Veille et Recherche d'outils
 - Ca marche en labo
 - Effet Wahou !
 - Pas de produit fini
- ✓ Accompagnement NIJI
 - Du mythe à la réalité
 - Définition des usages
 - POC
- ➔ Lancement du chantier Home-made

Avril - mai

Plateforme Technique

- ✓ Une IA conforme à la politique de sécurité
 - Choix Architecture (Saas / Paas / Onpremise)
- ✓ Une IA Bon marché et Simple (maîtrise des couts -Capex/Opex)
- ✓ Maturité des LLM
- ✓ Intégration IA dans Ecosystème MGDIS (API, Jira, Squash)
 - ➔ L'acteur Clé:
1 alternante Master dédiée IA
Estelle TOURASSE

Juillet - Septembre

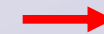
Déploiement Progressif

- ✓ Phase 1: Preuve de Concept
 - 1 référent activité
 - Conception de cas de test
- ✓ Phase 2: Ouverture Cellule de conception (4)
- ✓ Phase 3 : Et les autres
 - Chaque cellule définit ses usages
- ➔ Tester, prouver, ajuster, capitaliser
- ➔ « L'usage appelle l'usage »

Septembre - Novembre

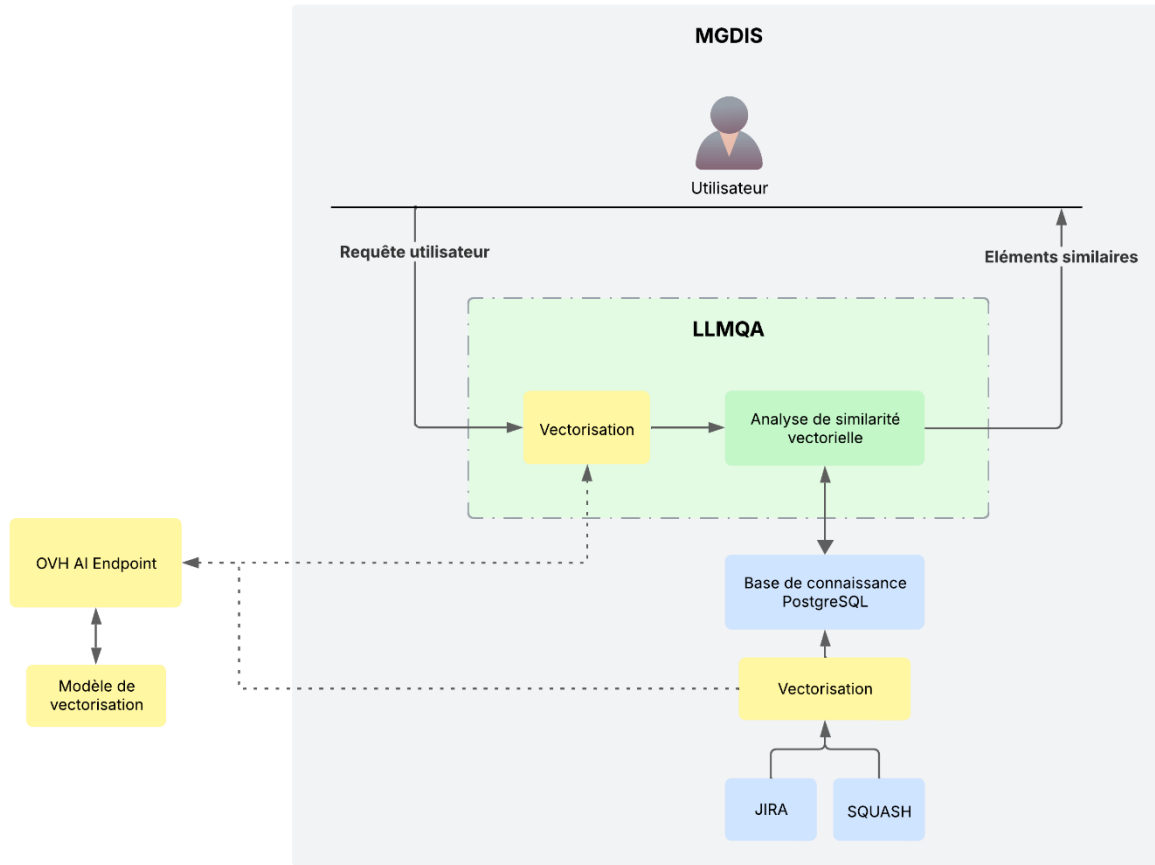
L'étude des cas d'usages (Roadmap)

ID	Type	Cas d'usage	Valeur	Complexité
1	Cas de test	Génération de cas de tests à partir de l'US	1	0
2	Cas de test	Compléter les cas de tests existants avec les points de contrôle	1	2
3	Cas de test	Proposer des tests différents : techniques, sécurité, performance, utilisabilité, etc.	1	1
4	Cas de test	Retrouver les cas de test similaires, identifier les doublons	1	3
5	US	Analyser la conformité d'une US	1	0
6	Automatisation	Conception des scripts Robot Framework à partir des cas de tests	1	2
7	JDD	Création d'un jeu de données pour un cas de test	1	3
8	Cas de test	Tests d'API pour des cas métiers pour des TNR	1	2
9	Cas de test	Tests RGA avec l'IA sur de l'interprétation	2	4
10	Campagne	Homologation : générer une campagne à partir des personas et du contexte des lots	2	3
11	LOGS	Analyse des logs suivant une nomenclature/conformité	2	1
12	US	À partir d'une US, détecter les cas de tests existants impactés devant être modifiés	2	3
13	JDD	Génération d'un script MongoDB pour alimenter la base à partir d'un JDD	2	1
14	US	Création de critères d'acceptation à partir du cas d'usage (exportable dans Jira)	2	2
15	BUG	Détection des zones de faiblesses	2	4
16	Cas de test	Reformulation d'un cas de test en respectant un template	3	1
17	JDD	Pouvoir retrouver une liste des scripts de JDD	3	3
18	US	Analyse sémantique / donner la compréhension de l'US	3	1
19	US	Reformulation sous la forme d'un template standard	3	1
20	US	Analyse et représentation graphique	3	1
21	LOGS	Identification des logs problématiques	3	2
22	Persona	Création d'une liste de personas	4	2
23	LOGS	Explication des logs	4	1

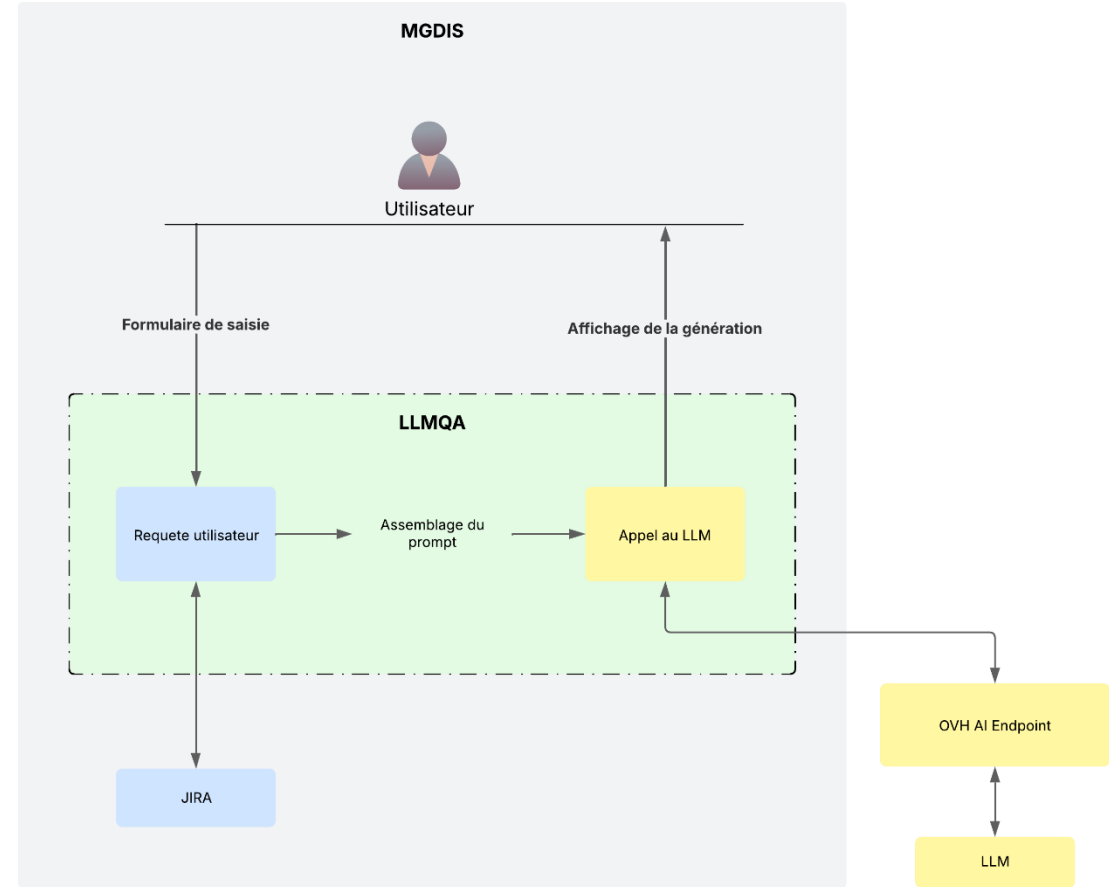


Traité par IDE, mise à disposition de licence Github Copilot

Schéma fonctionnel



UseCase: Détection des similarités



UseCase: Génération du cas de test

Nos nouveaux assistants virtuels

Conformité de l'exigence

« Dis moi si l'exigence entrante est bien écrite et si je peux écrire mes cas de test »



LOFN

Sentinelle mystique, elle tisse les fils entre les exigences, veille sur chaque exigence et prodigue les recommandations de qualité

Similarité des cas de test

« Est-ce que mon patrimoine a déjà des cas de test qui parle de ton exigence »



LEIF

Explorateur intrépide, il parcourt sans relâche notre savoir faire collectif et traque les similitudes

Génération des cas de test

« Génère moi les cas de test de cette exigence au format gherkin »




ORNN

Maitre Forgeron, concepteur infatigable, il consacre son énergie à façonner les cas de test

mgdis-ops-llm.node.ovh/llmqa/

PO / PM
Test QA
Support N2
ChatBot



LOFN

Votre assistante pour l'analyse de la conformité des exigences.

Numéro de l'Exigence :

Zone de saisie libre :

> Prompts système

Meta-Llama 3.3 70B Mistral small

Type d'analyse :

MUST SC RG

Consommation tokens
Heure : 8685 / 500000
Jour : 51684 / 4000000

Détection de similarité

mgdis-ops-llm.node.ovh/llmqa/Détection_de_similarités

> PO / PM

▼ Test QA

Conformité des exigences

► Détection similarités

Générateur de Tests


ChatBot Incidentologie

Assistant Test Exploratoire

Générateur de test V2

> Support N2

ChatBot



LEIF

Votre assistant pour la détection d'éléments similaires à un PLAID d'exigences fonctionnelles.

Numéro de l'Exigence :

Zone de saisie libre :


Aiden Geronimmo Multicité Numens Sofi

Générateur de tests

mgdis-ops-llm.node.ovh/llmqa/Générateur_de_Tests

- > PO / PM
- > Test QA
- > Support N2

ChatBot




Consommation tokens

Heure : 26043 / 500000

Jour : 37859 / 4000000

Votre assistant pour la génération de tests.



Numéro de l'Exigence :

?

+ Ajouter un ticket

Numéro du contexte (épopée) :

?

Zone de saisie libre :

Enrichir la génération avec la documentation Aiden (RAG) ?

> Prompts système

Meta-Llama 3.3 70B Mistral small

Générer

Génération

Données d'entrée



Défis Techniques et Leçons Apprises

Leçon 1 : IA souveraine, risques maîtrisés

- ⊕ **La problématique** : Fournir un outillage pour éviter le Shadow IA
 - Trouver une solution souveraine
 - Maîtriser l'installation et l'exploitation
 - Sécuriser les données, les applications, le savoir-faire

- ⊕ **Risque** : Monter une usine à Gaz , non maintenable, obsolète et coûteuse

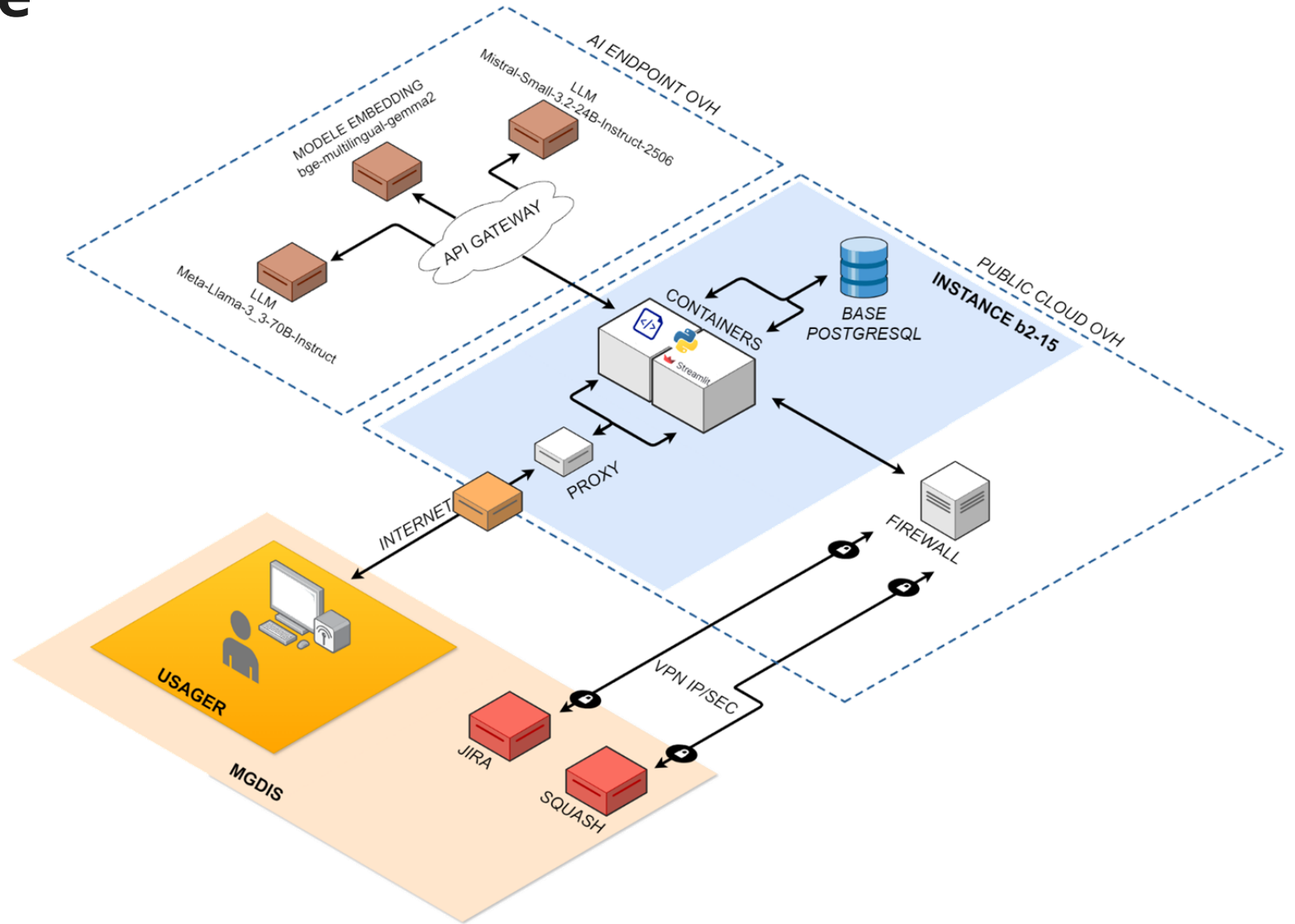
- ⊕ **Les études**:
 - ⊕ Solution « sur étagère » : N'existe pas
 - ⊕ Solutions intégrées : « ne permettent pas d'interconnexion avec notre Référentiel d'exigences, référentiel de cas de test, Documentation
 - ⊕ LLM en accès Libre : Interdit : perte de maîtrise des données, peu d'homogénéisation des résultats et des pratiques
 - ⊕ LLM sur Infrastructure interne: Extrêmement coûteux , obsolescence , exploitation , compétences de sécurisation ...

- ⊕ **La solution** : déploiement sur les IA endpoints OVHCloud
 - Résilience numérique
 - Souveraineté
 - Analyse de risque/Fournisseur
 - Consommation à la demande dans des tenants / sécurisation par des contrats OVH (*souverain*)
 - Déploiement simple et rapide

Architecture technique

Technologies choisies

- IA Endpoint OvhCloud
- Instance Public Cloud
- Script Python
- FrontEnd: Streamlit
- Stockage : Postgresql



Leçon 2 : L'IA pragmatique: coût et outillage

Objet

€ Cout

Setup

Consulting

LLM



Personnel



Personnel



Infrastructure

- Plateforme IA Endpoints
- POC des LLM
- Interface Streamlit
- Script Python appel API & Cron
- Développement des interfaces Jira, Squash
- Tests et ajustement de la solution

- Participation aux ateliers Niji
- Prompting des 3 cas d'usages
- Tests pertinence des résultats sur différents LLM

- Pilotage de projets

- Accompagnement Niji

- 1 instance VM
- 3 endpoints OVHCloud

Exemple : 1 cas de test = 8000 tokens

- 30 jours (Equivalence alternance)
Alternance Ingénieur IA

- 20 jours – Responsable conception des tests

- 10 jours – Responsable

- 13 jours

- bge-multilingual-gemma2 : 0,01 € HT pour 1M de tokens
- Meta-Llama-3_3-70B-Instruct : 0,67 € HT pour 1M de tokens
- InMistral-Small-3.2-24B-Instruct-2506 : 0,28 € pour 1M de token

1 VM – 15Go – 4CPU – 100Go ssd – 250Mb/s : 46€ /mois

Estimation sur Décembre 2025: Entre 50 et 60 €Ht /mois

#Leçon 3 : Du quick win au résultat sûr

Les constats :

Les résultats sont très intéressants sur des applications « grand public », mais encore perfectibles sur des logiciels spécialisés ou spécifiques

Les résultats sont très probants sur des petites fonctionnalités, mais non exhaustifs lorsque la combinatoire de test est importante

⊕ **Les Risques :**

- ⊗ Limiter l'utilisation à des micro-fonctionnalités
- ⊗ Rejet des testeurs

⊕ **Les solutions :**

Utiliser le RAG (Documentation, Tickets, Code, Exigences, cas de test ...) : Enrichir le contexte
Utiliser des LLM avec fenêtre de contexte importante
Contrôler: les testeurs doivent contrôler le résultat, challenger les réponses
Créer un agent de contrôle (Poser des questions, vérifier)

#Leçon 4 – Le prompt est l'équation magique

Un Mythe : le Prompt Unique

On pensait que « prompter » était rapide et simple = Investissement majeur pour la qualité des résultats

On pensait qu'il fallait un gros prompt = Échec (oubli du contexte, moins clair, moins précis et avec hallucination)

+ La Réalité :

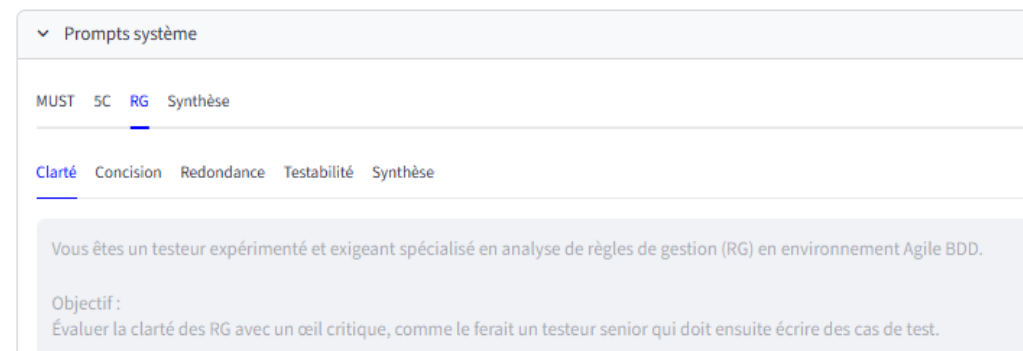
- Découper le prompt est mieux et donne de **meilleurs résultats** et un **meilleur impact**.
- Sécuriser ses méthodes et les résultats dans un prompt system
- Expliquer et fournir des exemples

+ Le résultat :

- Idempotence des résultats
- Évite les hallucinations
- Facilite le travail du testeur
- Homogénéise les sorties

+ La solution : Utilisation de l'IA pour prompter (Méta-prompting)

Adapter les prompts systèmes au besoin des testeurs : atteindre l'idempotence + Homogénéité des résultats



#Leçon 4 – Le prompt est l'équation magique

```
user_prompt_fr:
"
Titre: {id} - {titre}

Détails:
{description}
{criteres_acceptance}
"

system_prompt_fr:
"
Objectif:
Vous allez m'assister en tant qu'expert senior en qualité logicielle, spécialisé dans la validation d'applications web.
Votre objectif sera de générer des scénarios de test pour une user story ainsi que les critères d'acceptation associés, en tenant compte des cas passants.

Exigences:
- IMPORTANT : Générez votre réponse en français quelles que soient la langue d'entrée
- Les tests doivent suivre le format Gherkin : 'Étant donné - Quand - Alors'
- Suivez le format de sortie spécifié
- Générez le nombre nécessaire de tests pour couvrir tous les critères d'acceptation fournis par l'utilisateur

Instructions:
1. Analyser l'US et les critères d'acceptation associés
2. Générer des cas passants
3. Fournissez toujours votre sortie en français, même si l'entrée est dans une autre langue

Détail des étapes:
1. Analyser l'US et les critères d'acceptation associés
  - Lire et comprendre la User Story (US) décrivant une fonctionnalité spécifique de l'application à tester
  - Identifier les critères d'acceptation définis par l'utilisateur dans les détails de l'US
2. Générer des cas passants
  - Créer des scénarios de test au format Gherkin qui valident que la fonctionnalité fonctionne correctement selon les critères d'acceptation
  - Donner un titre au test
  - Fournir une liste d'étapes avec, pour chaque étape, une action et le résultat attendu

Format de sortie:
Votre rôle est de suivre le modèle ci-dessous. TOUT LE TEXTE DOIT ÊTRE EN FRANÇAIS quelle que soit la langue d'entrée. L'output doit être un json valide
{
  'cas_passants': [
    {
      'titre': 'titre du test',
      'description': 'description du test suivant le formalisme Gherkin',
      'exemple': 'phrase donnant plusieurs valeurs d'entrées à tester'
    }
  ]
}

Assurez-vous que vos réponses soient des objets JSON valides, correctement formatés et sans erreurs.
N'incluez aucun texte supplémentaire, aucune explication ou saut de ligne en dehors du JSON.
"
```

Le prompt n'est pas une recherche Google

Il se structure:

#Rôle

#Contexte

#Instruction

#Format

#Contraintes

#Données

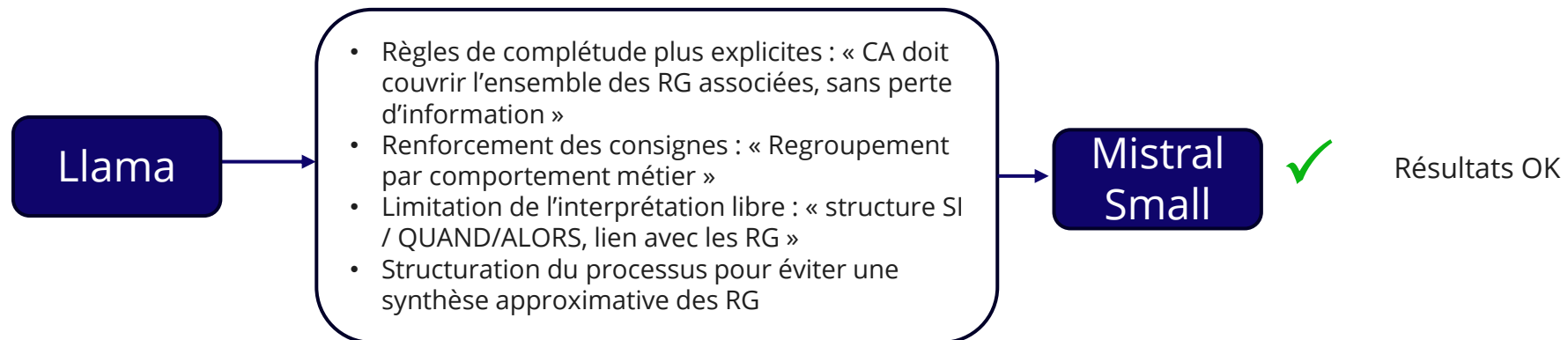
#Leçon 5 : Le trio gagnant : 1 usage, 1 LLM1 prompt

L'Expérimentation : Pourquoi on a **tâtonné** entre LLM (ex : Mistral 7b, Mistral Small et Llama 3.3 70b)

⊕ **Le Problème** : Les **résultats de prompts sont différents** selon le LLM.



⊕ **La Solution** : Il faut sélectionner le LLM en fonction de l'usage et adapter le prompt selon le LLM



#Leçon 6. Skills first, prompts after

- ⊕ **La problématique** : population de testeurs non initiés à l'IA et non techniques
- ⊕ **Erreur**: Perte de temps dans les expérimentations, tâtonnements, veille, mimétisme d'autres usages...

- ⊕ **Leçons !**
 - ↳ Former les personnes clés sur l'IA et le prompt en amont
 - ➔ La certification (*pas disponible au démarrage de notre projet*) ISTQB: Testing with Generative AI est un concentré de bonnes pratiques en 2 jours : c'est indispensable pour concevoir les agents IA

 - ↳ Sensibiliser les testeurs
 - ➔ Rôle du testeur
 - ➔ Les risques (Qualité, facteurs humains)
 - ➔ Le prompt
 - ➔ L'analyse des résultats

#Leçon 7. L'IA copilote, le testeur pilote

- ⊕ **Les problématiques** : Population de testeurs = « non techniques »
Besoin d'homogénéisation des résultats en sorties
Adhésion des testeurs
- ⊕ **Expérimentation** : Choisir entre un Chatbot - API - IHM spécifique

Numéro de l'Exigence : ?

PLAID-107809

+ Ajouter un ticket

Numéro du contexte (épopée) : ?

Zone de saisie libre :

Enrichit le résultat avec un exemple de jeu de données

Enrichir la génération avec la documentation Aiden (RAG) ?

> Prompts système

Meta-Llama 3.3 70B Mistral small

Ticket extrait depuis JIRA :
EA - Traitement de masse de génération des notifications de décision en mode expert (entrée fichier CSV)

Générer

Modifier un CA Fusionner des CA Scinder un CA Supprimer un CA Valider les CAs

La conséquence : création d'une IHM surcouché du LLM et des prompts avec parcours fléchés dédiés au métier du testeur

Interface préparant les éléments pour chaque agent, avec du pas à pas guidé

Enrichissement par l'expérience du testeur

Cadrage des outils

Choix et décision laissés au testeur

L'Impact Mesuré et la Diversification des Usages

L'Impact Mesuré (KPI)

- ⊕ **Lutte contre la « Feuille Blanche »** : l'IA aide à initier la réflexion, à poser une première structure de test, le concepteur peut valider ou challenger la solution

- ⊕ **Réduction du Temps de Conception** : Passage de **4h à 5 - 10 minutes**
 - ⊕ Le temps gagné en écriture est investi dans une meilleure collaboration entre PO et testeurs pour le raffinement de l'exigence
 - ⊕ Coût 1^{er} scénario de test: 120€ → à 10€

- ⊕ **Réorganisation de l'équipe de conception** : passage de 4 à 3 concepteurs de test (-25%)

- ⊕ **Idéation** : arrivée de nouveaux tests – éviter le paradoxe du pesticide

- ⊕ **Amélioration du patrimoine de tests** → Cas de test plus clairs, plus compréhensibles :
 - ⊕ Détection et diminution des doublons,
 - ⊕ cas de tests plus précis,
 - ⊕ standardisé : Même style d'écriture quel que soit le concepteur,
 - ⊕ Homogénéité : Même granularité des tests quelque soit l'exigence

Diversification des Usages : Aller au-delà de la Génération

« L'usage appelle l'usage »

⊕ **Analyse du Support/Incident : Incidentologie par prompt IA**

- ⊗ Regrouper et corrélérer les incidents issus de l'ITSM à la notion de bug.
- ⊗ Analyser syntaxiquement des demandes clients → Remonter les défauts sur une version, un type de client, une fonctionnalité
- ⊗ Dialoguer avec IA pour proposer l'effort de test et la stratégie de test à appliquer

⊕ **Chiffrage des tests assistés par IA**

- ⊗ Création de squelette de test, estimation des temps de test pour macro-chiffrage des évolutions

⊕ **Analyse des logs en production:** Utiliser l'IA pour définir les **zones de faiblesses** (par séries temporelles) et prédire les nouveaux défauts.

- ⊗ Série Temporelle - Étude des analyses en production basée sur les logs sur les équipes SRE

⊕ **Génération de cas de tests à partir d'images (maquettes)**

⊕ **Exécution des cas de test sur Lynqa** (interaction avec LLMQA)

L'impact : L'IA Transforme nos métiers

Le métier de testeur va-t-il disparaître ?

NON

Oui: ce sont de **nouveaux outils**! Il faut les appréhender

Non, nous avons connu aussi les sirènes de l'agilité, de l'automatisation... et pourtant on est toujours en train de valider une version avec nos testeurs !

Oui, cela nécessite une adaptation pour positionner **nos agents virtuels sur des tâches répétitives, sans valeur ajoutée directe**

Non, comme il y a des adaptations comme sur les autres métiers d'ailleurs, n'oublions pas que l'IA hallucine, mais les testeurs sont les experts !

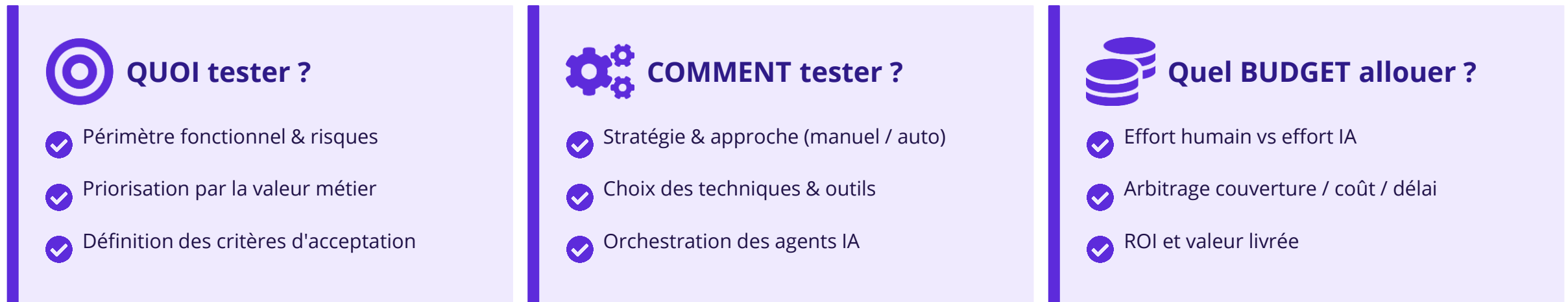
Non ! On peut se concentrer sur des tâches plus complexes, des interactions avec les métiers c'est aussi une **opportunité de rendre notre métier encore plus intéressant**

Notre vision : le QA Strategist, chef d'orchestre de ses agents

Le testeur évolue : il conçoit la stratégie, pilote l'exécution par ses agents, et garantit la valeur



Les 3 décisions stratégiques que seul le QA Strategist prend



→ **Le QA n'est pas remplacé : il monte en stratégie. La valeur du métier se déplace vers la décision et l'orchestration.**

Synthèse des leçons appprises

- **Leçon 1** : L'industrialisation nécessite un socle technique maîtrisé (Architecture, sécurité, API).
- **Leçon 2** : L'IA pragmatique: coût et outillage
- **Leçon 3** : Commencer petit pour déployer large (Déploiement progressif, PoC).
- **Leçon 4** : La qualité du prompt est la clé de la performance (Rôle de l'alternant, système de prompt).
- **Leçon 5** : Le trio gagnant : 1 usage, 1 LLM, 1 prompt
- **Leçon 6** : La technologie ne suffit pas, la formation est le vrai moteur.
- **Leçon 7** : L'IA est un copilote, pas un pilote automatique (Focus sur l'adoption et le rôle du testeur).