

JOURNÉE  
FRANCAISE DE  
L'INGÉNIERIE DES  
EXIGENCES



5 | 6 | 7  
NOVEMBRE 2024

11H30 - 13H30



**Gilles NOUAIS**

**IA POUR LA GÉNÉRATION DES TESTS À PARTIR DES EXIGENCES**



ÉVÈNEMENT SPONSORISÉ PAR



# Agenda

1. **Introduction**
2. **Préparation des Exigences Optimisation des Exigences pour usage dans l'IA**
3. **Génération de Cas de Test**
4. **Boucle de Rétroaction et Amélioration**
5. **Conclusion**

# Introduction

## Constat

---

**Les exigences peuvent être ambiguës, mal définies ou incomplètes.**

---

**Les projets complexes comportent souvent des exigences conflictuelles ou incohérentes.**

---

**La gestion d'un grand nombre de cas de test manuellement peut être extrêmement laborieuse et sujette aux erreurs.**

---

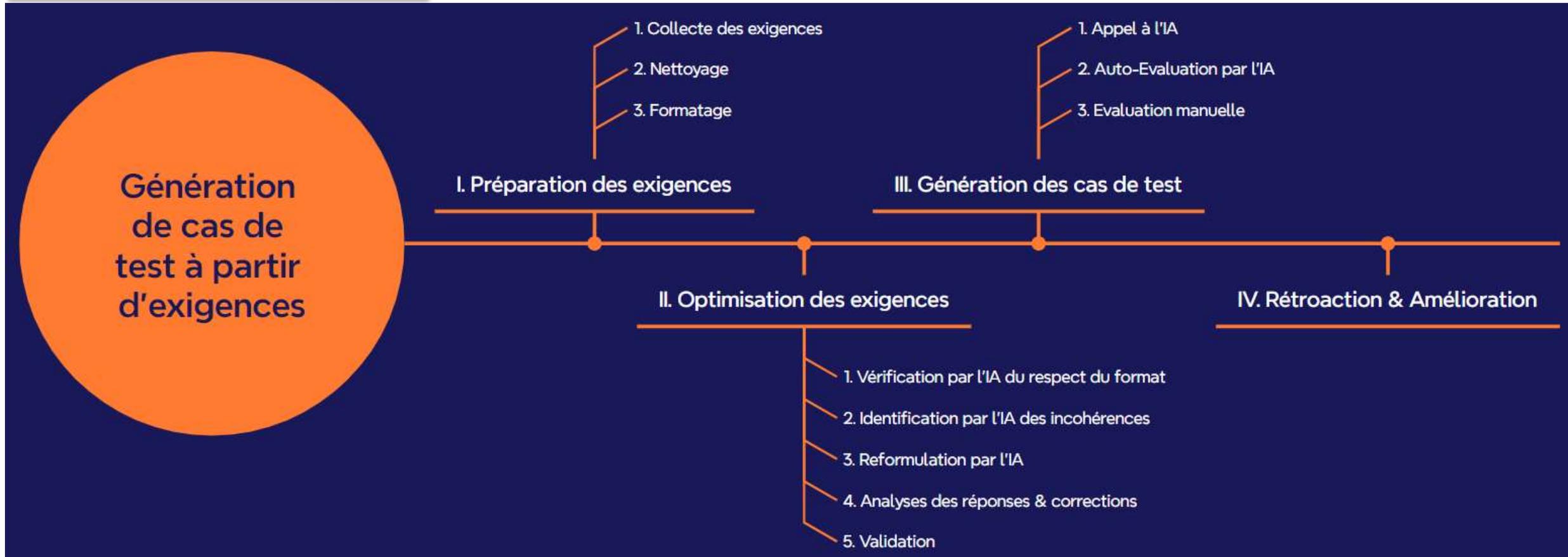
**Le boom de l'IA générative semble pouvoir apporter des réponses.**

## Question

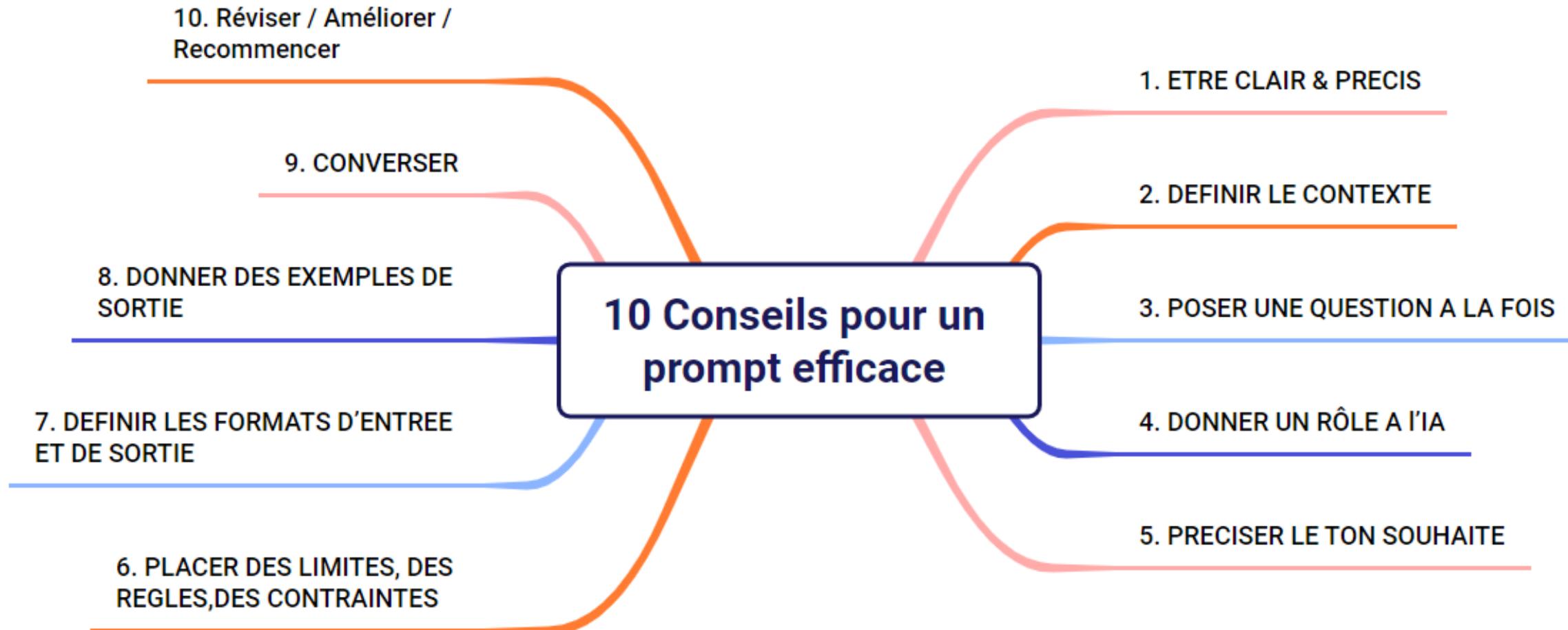
**Comment l'IA peut être intégrée dans le processus d'ingénierie des exigences pour générer des cas de test à partir d'exigences ? Et comment garantir une couverture exhaustive des exigences et une génération de cas de test de haute qualité, tout en optimisant le temps et les ressources ?**

# Introduction

## Notre process



# Introduction



# I. Préparation des exigences

Génération  
de cas de  
test à partir  
d'exigences

1. Collecte des exigences
  2. Nettoyage
  3. Formatage
- I. Préparation des exigences

1. Appel à l'IA
  2. Auto-Evaluation par l'IA
  3. Evaluation manuelle
- III. Génération des cas de test

II. Optimisation des exigences

1. Vérification par l'IA du respect du format
2. Identification par l'IA des incohérences
3. Reformulation par l'IA
4. Analyses des réponses & corrections
5. Validation

IV. Rétroaction & Amélioration

# I. Préparation des exigences



## Collecte des exigences

- α **Sélection** dans l'outil de gestion des exigences (IBM Doors, Jama Connect, Codebeamer, Xebrio...)
- α **Export** sous forme de fichier (word, pdf, excel, xml...)



## Nettoyage

Élimination des éléments non pertinents (commentaires, images...)



## Formatage

Conversion en texte brut

# I. Préparation des exigences

## PROMPT NETTOYAGE

Bonjour ChatGPT, Je travaille sur un projet d'ingénierie des exigences et j'ai besoin de nettoyer une liste d'exigences en éliminant les éléments non pertinents tels que les commentaires, les images référencées et les annotations superflues.

**Contexte :** Je vais utiliser ces exigences propres pour générer des cas de test par la suite. Donc, j'ai besoin que le texte soit clair et exempt de toute information superflue.

**Tâche :** Tu es un assistant d'ingénierie des exigences. Peux-tu s'il te plaît nettoyer la liste d'exigences ci-dessous ?

- **Ton souhaité :** Professionnel et précis
- **Règles et contraintes :**
  - o Supprimer tous les commentaires (indiqués par ).
  - o Supprimer toutes les références d'image (indiquées par (Image ref: image\_url)).
  - o Supprimer toutes les annotations (indiquées par // Annotation superflue).
- **Format d'entrée et de sortie :** Texte brut Exigences :

1. [Texte de l'exigence]
2. [Texte de l'exigence avec image] (Image ref: image\_url)
3. [Texte de l'exigence] // Annotation superflue ... Merci de m'aider à obtenir une version nettoyée de cette liste.



Nettoyage

### Nettoyages réalisés :

1. **EX-004** : Suppression de « (link: ../image.png) ».
2. **EX-007** : Suppression de « (link: ../image.png) ».
3. **EX-004** : Suppression de « // Annotation superflue ».
4. **EX-004** : Suppression de « <!-- Autre commentaire insolite --> ».

Si vous avez des exigences supplémentaires ou des clarifications, n'hésitez pas à me le faire savoir!



Message GPT-4

# I. Préparation des exigences

**Gilles NOUAIS**

 ExigencesKO\_doublon\_in...  
File

**\*\*PROMPT FORMATAGE\*\***

Bonjour ChatGPT,  
J'ai nettoyé une liste d'exigences et j'ai maintenant besoin de la convertir en texte brut pour la rendre compatible avec l'analyse par l'IA.

**\*\*Contexte :\*\***  
Ces exigences nettoyées seront utilisées pour générer des cas de test. Je souhaite que le format final soit en texte brut, bien structuré et exempt de tout formatage compliqué.

**\*\*Tâche :\*\***  
Tu es un assistant d'ingénierie des exigences. Peux-tu s'il te plaît convertir la liste suivante en texte brut ?

- **\*\*Ton souhaité :\*\*** Professionnel et précis
- **\*\*Format d'entrée :\*\*** Exigences nettoyées
- **\*\*Format de sortie :\*\*** Texte brut, bien structuré

Merci de me fournir cette liste d'exigences en texte brut.



**GPT-4**  
Bien sûr, je peux vous aider avec cela. Voici les exigences nettoyées, présentées en texte brut et bien structurées :

**EX-001**

Type d'exigence : Fonctionnelle

Description : Le système doit permettre aux utilisateurs de paramétrer les heures de rendez-vous pour un appel différé dans le Back-Office (BO), avec une plage horaire allant de 8h00 à 19h30, ajustée toutes les 30 minutes.

US liée : FDSO-43112

Priorité : Moyenne

Justification : Cette fonctionnalité permet d'élargir les heures d'ouverture du service client, augmentant ainsi le nombre de clients ou prospects que le service peut traiter.

Critère de test : Vérifier que les heures de rendez-vous peuvent être paramétrées dans le BO toutes les 30 minutes entre 8h00 et 19h30 et qu'elles sont visibles et sélectionnables par l'utilisateur.

Dépendances : EX-002 (intégration avec Genesys pour la gestion des rendez-vous téléphoniques).

Critères d'acceptance :

- L'utilisateur peut sélectionner des créneaux horaires de rendez-vous toutes les 30 minutes entre 8h00 et 19h30.
- Un message de confirmation est affiché pour chaque sélection de rendez-vous.
- Des SMS de confirmation sont envoyés pour chaque rendez-vous pris.
- Aucune erreur n'est générée lors du paramétrage ou de la modification des horaires dans le BO.



 Message GPT-4 



Formatage

# II. Optimisation des exigences

Génération  
de cas de  
test à partir  
d'exigences

I. Préparation des exigences

1. Collecte des exigences
2. Nettoyage
3. Formatage

III. Génération des cas de test

1. Appel à l'IA
2. Auto-Evaluation par l'IA
3. Evaluation manuelle

II. Optimisation des exigences

1. Vérification par l'IA du respect du format
2. Identification par l'IA des incohérences
3. Reformulation par l'IA
4. Analyses des réponses & corrections
5. Validation

IV. Rétroaction & Amélioration

# II. Optimisation des exigences

## Analyse sémantique

- Vérification par l'IA du respect du format
  - Utiliser des phrases complètes.
  - Inclure des conditions et des actions claires.
  - Éviter l'ambiguïté et la subjectivité.
  - Une exigence par ligne.
  - Utiliser un format structuré (identifiant unique, assertion, critères de validation, justification (optionnelle), priorité, auteur).
- Identification des Incohérences et Conflits
- Reformulation pour Clarté et Précision

## Analyse des réponses et Corrections

- Evaluation des Suggestions : Pertinences, faisabilité
- Implémentation des Modifications : Prise en compte dans le fichier texte brut

## Vérification Finale & Validation

- Nouvelle analyse après modifications : Effectuer une nouvelle analyse des exigences après implémentation des modifications pour s'assurer qu'elles restent cohérentes et précises.
- Validation avec les Parties Prenantes : Obtenir l'approbation des parties prenantes pour les exigences modifiées. Prendre en compte les modifications dans l'outil source.

# II. Optimisation des exigences

Bonjour ChatGPT, Je travaille sur des exigences pour un projet logiciel et j'ai besoin de vérifier chaque exigence avant de les utiliser pour générer des cas de test.

## Contexte:

- Les exigences doivent respecter le format prescrit par l'IREB et inclure un identifiant unique, une description claire, des critères de validation, une justification (optionnelle), et une priorité
- Les exigences doivent être formulées selon les critères suivant : claires, complètes, consistantes et sans ambiguïté
- Les exigences doivent être cohérentes et exemptes de conflits

**Ton souhaité :** Professionnel et précis

## Règles et contraintes :

- Toutes les exigences fournis doivent être analysé.
- Identifier les contradictions et les duplications.
- Utiliser des phrases complètes et éviter les ambiguïtés.
- Inclure des conditions et des actions claires.
- Chaque exigence doit inclure :
  - Un identifiant unique
  - Une assertion claire et concise
  - Des critères de validation spécifiques
  - Une justification (optionnelle)
  - Une priorité (Haut, Moyen, Bas)
  - L'auteur de l'exigence
- Proposer des suggestions de résolution.

**Format d'entrée :** Liste des exigences donner sous forme d'un fichier texte contenant 15 exigences

**Format de sortie :** Liste sous forme de tableau des exigences. Le tableau contient les colonne suivantes :

- Identifiant exigence
- Conformité
- Problèmes
- Proposition de correction

## Exemple de résultat :

Id	Conformité	Problème	Résolution
EX-001	OK		
EX-002	KO	Absence de critère d'acceptation	Les commentaires sont enregistrés et les URLs ou balises HTML dans les commentaires sont neutralisées L'utilisateur doit pouvoir cliquer sur un tag de sous-thématique à la fin d'un article pour afficher d'autres contenus liés à cette sous-thématique sur la page "Conseils et Actus".
EX-003	KO	Description incomplète	
EX-004	KO	Incohérence : La plage horaire d'ouverture défini dans l'exigence 004 et en contradiction avec la plage de l'exigence EX-001	La plage d'ouverture est 8h30 - 12h00 et 14H - 17h30

# II. Optimisation des exigences

Voici un tableau qui résume l'analyse des exigences en fonction des critères demandés :

Id	Conformité	Problèmes	Proposition de correction
EX-001	OK		
EX-002	OK		
EX-003	OK		
EX-004	OK		
EX-005	OK		
EX-006	OK		
EX-007	OK		

EX-008	OK		
EX-009	OK		
EX-010	OK		
EX-011	KO	Duplication avec EX-001	Retirer l'exigence EX-011 car elle est identique à EX-001.
EX-012	KO	Absence de description	Ajouter une description claire pour EX-012.
EX-013	KO	Contradiction : les SMS de rappel sont envoyés aux numéros commençant par 08 ou 09 au lieu de 06 ou 07 comme stipulé dans les critères d'acceptance.	Corriger pour indiquer que les SMS de rappel doivent être envoyés aux numéros 06 ou 07 conformément aux critères d'acceptance.
EX-014	KO	Description incohérente, manque de structure et absence de justification, priorité, etc.	Reformuler entièrement cette exigence pour qu'elle soit claire, complète, et conforme au format requis.

Cela permet de voir les problèmes d'ambiguïté, de duplication et de contradictions dans les exigences. N'hésitez pas à ajuster si vous avez besoin de plus de détails sur une exigence en particulier.

🔊 🗨️ 📄 🗑️ ↺



📄 Message ChatGPT

# III. Génération des cas de test

Génération  
de cas de  
test à partir  
d'exigences

## I. Préparation des exigences

1. Collecte des exigences
2. Nettoyage
3. Formatage

## III. Génération des cas de test

1. Appel à l'IA
2. Auto-Evaluation par l'IA
3. Evaluation manuelle

## II. Optimisation des exigences

1. Vérification par l'IA du respect du format
2. Identification par l'IA des incohérences
3. Reformulation par l'IA
4. Analyses des réponses & corrections
5. Validation

## IV. Rétroaction & Amélioration

# III. Génération des cas de test



## Appel à l'IA

Utiliser un LLM pour générer des cas de test détaillés à partir des exigences.

Processus : L'IA analyse chaque exigence et en déduit des scénarios de test spécifiques.

Entrée : Liste des exigences formalisées et optimisé pour l'IA.

Sortie : Liste de cas de test détaillés couvrant chaque exigence.



## Auto-évaluation

L'IA analyse les cas de test pour s'assurer qu'ils couvrent toutes les exigences et qu'ils ne présentent pas d'incohérences.

Critères : Couverture exhaustive des exigences, Absence d'ambiguïté. Clarté des étapes.



## Évaluation Manuelle des Cas de Test

Les experts QA passent en revue chaque cas de test, valident les scénarios et apportent des ajustements si nécessaires.

# III. Génération des cas de test

Bonjour ChatGPT,

Je travaille sur des exigences pour un projet logiciel et j'ai besoin de générer des cas de test détaillés à partir de ces exigences.

**Contexte :** Chaque exigence doit être transformée en un ou plusieurs cas de test pour vérifier que les fonctionnalités sont correctement implémentées.

**Tâche :** Tu es un assistant d'ingénierie des exigences. Peux-tu générer des cas de test détaillés pour les exigences suivantes ? Il faut des scénarios distincts pour: Les cas passants, les cas d'erreurs et les cas aux limites (Borne inférieure, borne supérieure, valeur vide, ...). Pour chaque type, ils faut plusieurs scénarios

- **Ton souhaité :** Clair et détaillé
- **Règles et contraintes :**
  - Chaque cas de test doit inclure :
    - ◆ Un identifiant unique, Un titre, Un objectif, Les préconditions
    - ◆ Les étapes à suivre et Les résultats attendus, Les données de test (si applicables)
- **Format d'entrée :** Liste des exigences formalisées
- **Format de sortie :** Liste des cas de test détaillés.

## Exemple Cas de Test : Recherche d'une chambre

### Cas Passant : Recherche standard

**Titre du cas de test :** Recherche de disponibilité standard

**Objectif :** Vérifier que la recherche de disponibilité fonctionne correctement avec des critères valides.

**Pré-requis :** Le client est sur la page d'accueil du site de réservation d'hôtel.

Étape	Action	Résultat attendu
1	Cliquer sur l'option de recherche de disponibilité	L'interface de recherche s'affiche avec des champs pour spécifier les dates de séjour, le nombre d'adultes/enfants, etc.
2	Entrer les dates de séjour <date_début> à <date_fin>	Les dates sont acceptées et les champs sont remplis correctement.
3	Spécifier le nombre d'adultes <nombre_adultes> et d'enfants <nombre_enfants>	Les informations sont acceptées et les champs sont remplis correctement.
4	Cliquer sur le bouton "Rechercher"	Une liste de chambres disponibles s'affiche avec les détails tels que le prix, les équipements, les photos, etc.

### Cas d'Erreur : Dates non valides

**Titre du cas de test :** Recherche avec dates non valides

**Objectif :** Vérifier la gestion des erreurs lorsque des dates non valides sont saisies.

**Pré-requis :** Le client est sur la page d'accueil du site de réservation d'hôtel.

Étape	Action	Résultat attendu
-------	--------	------------------

# III. Génération des cas de test

Je vais générer des cas de test détaillés pour certaines des exigences que vous avez fournies dans le document. Commençons par quelques exemples concrets.

## Exigence EX-001 : Paramétrage des heures de rendez-vous pour un appel différé

Cas Passant : Paramétrage correct des heures

Identifiant : CT-EX001-01

Titre : Paramétrage d'un rendez-vous entre 8h00 et 19h30

Objectif : Vérifier que les créneaux horaires peuvent être paramétrés toutes les 30 minutes entre 8h00 et 19h30.

Préconditions : L'utilisateur est connecté au Back-Office et accède à la section de gestion des rendez-vous.

Données de test : Aucun

Étape	Action	Résultat attendu
1	Accéder à la section de paramétrage des heures de rendez-vous	L'interface affiche les créneaux horaires de 8h00 à 19h30, ajustés toutes les 30 minutes.
2	Sélectionner un créneau à 10h00	Le créneau est sélectionné correctement, et un message de confirmation s'affiche.
3	Valider le paramétrage	Le système sauvegarde les paramètres sans générer d'erreurs.

Cas d'Erreur : Créneau horaire invalide

Cas d'Erreur : Créneau horaire invalide

Identifiant : CT-EX001-02

Titre : Paramétrage d'un rendez-vous hors des heures permises

Objectif : Vérifier que le système ne permet pas de sélectionner des créneaux hors des heures permises.

Préconditions : L'utilisateur est connecté au Back-Office et accède à la section de gestion des rendez-vous.

Données de test : Aucun

Étape	Action	Résultat attendu
1	Essayer de paramétrer un créneau à 7h30	Le système affiche un message d'erreur indiquant que l'heure est hors de la plage permise.
2	Valider le paramétrage	Le système bloque l'action et ne sauvegarde pas le créneau.

Cas Limite : Sélection du premier et dernier créneau

Identifiant : CT-EX001-03

Titre : Paramétrage du premier et dernier créneau

Objectif : Vérifier que le système accepte le premier créneau (8h00) et le dernier (19h30).

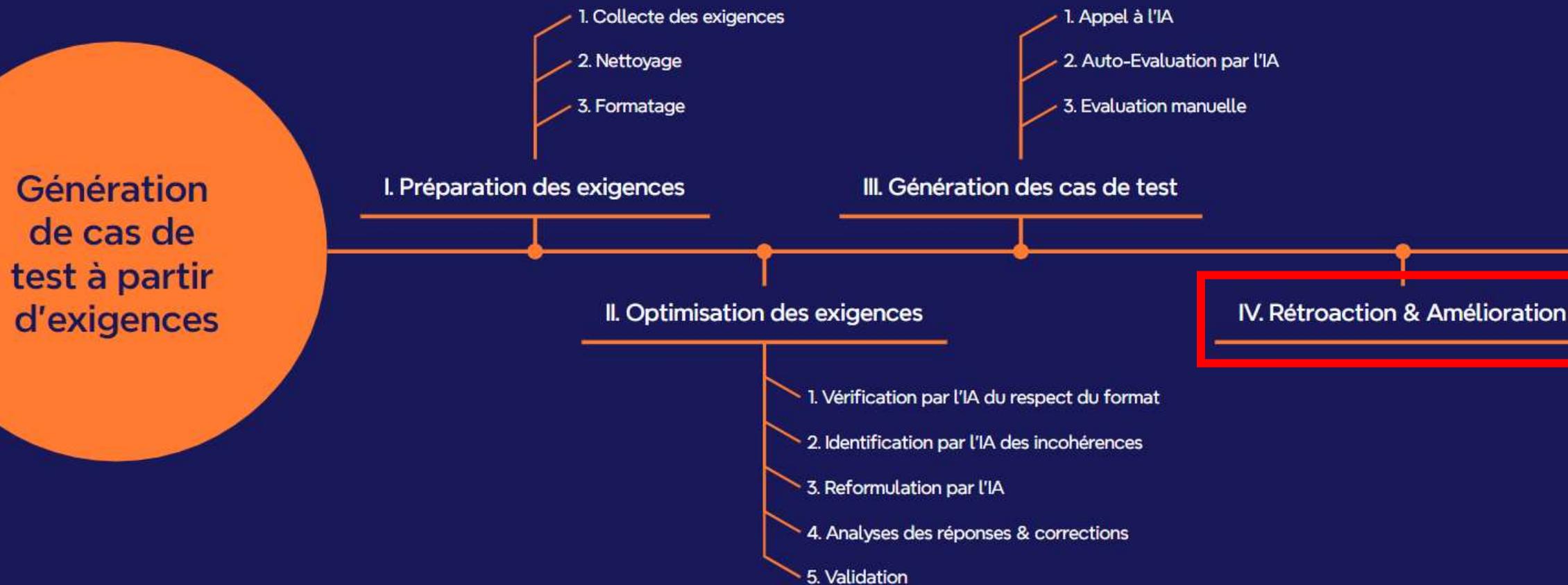
Préconditions : L'utilisateur est connecté au Back-Office et accède à la section de gestion des rendez-vous.

Données de test : Aucun

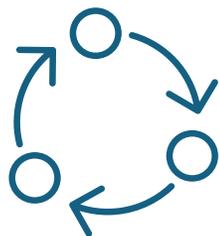
Étape	Action	Résultat attendu
1	Sélectionner le créneau de 8h00	Le créneau est sélectionné correctement, et un message de confirmation s'affiche.
2	Sélectionner le créneau de 19h30	Le créneau est sélectionné correctement, et un message de confirmation s'affiche.

Exigence EX-002 : Like/Dislike dans la FAQ

# IV. Rétroaction & Amélioration



# IV. Rétroaction & Amélioration



## Itération

- ⌘ Si nécessaire, itérez sur les étapes précédentes en ajustant l'US transformée ou en reformulant le prompt pour obtenir des résultats encore plus pertinents.
- ⌘ Utilisez le processus itératif pour optimiser la qualité des cas de test générés.



## Rétroaction

- PLAN** : Collecter du feedback (Retour sur la pertinence et l'efficacité des tests exécutés) , analyser, planifier
- DO** : Déployer les améliorations. Mise à jour des exigences, Mise à jour des cas de test, Mise à jour des prompts
- CHECK** : Mesurer l'impact des améliorations.
- ACT** : Corriger, améliorer

# Conclusion

Génération  
de cas de  
test à partir  
d'exigences

## I. Préparation des exigences

1. Collecte des exigences
2. Nettoyage
3. Formatage

## III. Génération des cas de test

1. Appel à l'IA
2. Auto-Evaluation par l'IA
3. Evaluation manuelle

## II. Optimisation des exigences

1. Vérification par l'IA du respect du format
2. Identification par l'IA des incohérences
3. Reformulation par l'IA
4. Analyses des réponses & corrections
5. Validation

## IV. Rétroaction & Amélioration

# Conclusion

- C'est un Process encore « artisanal »
- Les outils commencent à intégrer quelques fonctionnalités de génération de cas de test

## Avantages clés de l'IA :



**Gain de temps et efficacité**



**Amélioration de la qualité et de la précision**



**Couverture plus exhaustive des Exigences**



**Boucle de rétroaction continue**

## Prochaines étapes :



**Adoption de l'IA dans vos processus**



**Formation et sensibilisation**



**Premières expérimentations**



**Évaluation et ajustements**