

# Stratégie de test : Concevez une stratégie de test

## Scénarios prévus

### Sprint 1 et 2 – Scénarios prévus

TOM-6 – Création de comptes d'utilisateurs sécurisés (Criticité : Moyenne)

Scénario	Exigence Vérifiée	Objectif
SC-1.1	TOM-6	Vérifier que le champ « Nom » accepte du texte jusqu'à 50 caractères et est éditable.
SC-1.2	TOM-6	Vérifier que le champ « Prénom » accepte du texte jusqu'à 50 caractères et est éditable. (Anomalie potentielle : actuellement indiqué comme "numérique")
SC-1.3	TOM-6	Vérifier que le champ « Date de naissance » est au format JJ/MM/AAAA et bloque les dates postérieures à 2007.
SC-1.4	TOM-6	Vérifier que le champ « Mot de passe » accepte 50 caractères, avec les contraintes : majuscule, chiffre, caractère spécial, ≠ date de naissance.
SC-1.5	TOM-6	Vérifier qu'en cliquant sur "Suivant", l'utilisateur est redirigé vers la page de téléversement des documents.
SC-1.6	TOM-6	Vérifier que les données saisies valides sont conservées après clic sur « Suivant ».

TOM-7 – Téléversement des pièces justificatives (Criticité : Moyenne)

Scénario	Exigence vérifiée	Objectif
SC-2.1	TOM-7	Vérifier que la liste des documents fournir s'affiche et qu'il est non éditable et pour les en cours de téléversement
SC-2.2	TOM-7	Vérifier l'ajout d'un document via le bouton « Parcourir » et son affichage en cours de téléversement
SC-2.3	TOM-7	Vérifier que les documents téléversés sont affichés avec le bon statut.
SC-2.4	TOM-7	Vérifier que le bouton « Enregistrer » sauvegarde les infos en Base de données
SC-2.5	TOM-7	Vérifier l'apparition de la pop-up après clic sur «

		<b>Annuler » avec redirection correcte selon le choix :</b> <ul style="list-style-type: none"> <li>• Au clic sur “Oui”, rediriger vers l'accueil</li> <li>• Au clic sur “Non” : rester sur la page</li> </ul>
SC-2.6	TOM-7	Vérifier que si l'utilisateur clique sur “Annuler”, le fichier ajouté est supprimé et l'utilisateur est redirigé.

#### TOM-8 – Authentification double facteur (Criticité : Critique)

Scénario	Exigence Vérifiée	Objectif
SC-3.1	TOM-8	Vérifier que le système interroge la brique SECU via l'URL (GET) à la connexion.
SC-3.2	TOM-8	Vérifier que si connexion = OK, l'utilisateur est redirigé vers le tableau de bord.
SC-3.3	TOM-8	Vérifier que si connexion = KO, le système demande une authentification 2FA double facteur.
SC-3.4	TOM-8	Vérifier que si 2FA validé, alors redirection vers le tableau de bord.
SC-3.5	TOM-8	Vérifier que le système envoie ensuite un POST à la brique SECU avec les nouvelles données de connexion.

#### TOM-9 – Authentification biométrique (Criticité : Critique)

Scénario	Exigence vérifiée	Objectif
SC-4.1	TOM-9	Vérifier que le bouton biométrie est affiché sur la page de connexion.
SC-4.2	TOM-9	Vérifier que le système appelle bien la fonction biométrique de l'OS du périphérique.
SC-4.3	TOM-9	Vérifier que l'utilisateur est redirigé vers son tableau de bord si l'authentification biométrique est correcte.

### Sprint 3– Scénarios prévus

#### TOM-10 – Interconnexions bancaires (Criticité : Critique)

Scénario	Exigence vérifiée	Objectif
SC-5.1	TOM-10	Vérifier que l'appel GET vers l'API /banques/infos est correctement exécuté à la demande.
SC-5.2	TOM-10	Vérifier que les données reçues contiennent tous les champs attendus : nom banque, type de compte, montant, identifiant client, appareil.

SC-5.3	TOM-10	Vérifier que les données récupérées sont bien stockées en base de données.
SC-5.4	TOM-10	Vérifier qu'une seule demande toutes les cinq minutes peut être faite .
SC-5.5	TOM-10	Vérifier que si l'API prend plus de 30 secondes à répondre, un message d'erreur clair s'affiche à l'utilisateur.

#### TOM-11 – IHM Tableau de bord (Criticité : Critique)

Scénario	Exigence vérifiée	Objectif
SC-6.1	TOM-11	Vérifier que chaque banque connectée affiche un bloc distinct sur le tableau de bord.
SC-6.2	TOM-11	Vérifier que le bouton "Rafraîchir" relance un appel au service de la BDF banques/infos
SC-6.3	TOM-11	Vérifier que l'heure de la dernière mise à jour est affichée .
SC-6.4	TOM-11	Vérifier que l'accès au tableau de bord n'est possible que si l'utilisateur est connecté.
SC-6.5	TOM-11	Vérifier le comportement du tableau de bord en cas de non-réponse de l'API : message, vide.

#### TOM-12 – Téléchargement des relevés (Criticité : Moyenne)

Scénario	Exigence Vérifiée	Objectif
SC-7.1	TOM-12	Vérifier que le bouton "Télécharger mes relevés" s'affiche bien dans le tableau de bord.
SC-7.2	TOM-12	Vérifier que le clic sur ce bouton déclenche bien un appel au service de la BDF <a href="http://banquedefrance.com/api/v2/banques/riv">http://banquedefrance.com/api/v2/banques/riv</a> .
SC-7.3	TOM-12	Vérifier que le fichier PDF généré contient les données du bon utilisateur.
SC-7.4	TOM-12	Vérifier que le document se télécharge correctement via le système d'exploitation de l'utilisateur (OS).
SC-7.5	TOM-12	Vérifier l'affichage d'un message d'erreur en cas d'échec de téléchargement

## Sprint 4 – Scénarios prévu

**TOM-13 – Interconnexion et transactions** (Criticité : Critique)

Scénario	Exigence vérifiée	Objectif
<b>SC-8.1</b>	TOM-13	Vérifier que l'appel à l'API /banques/transactions renvoie les bonnes données (destinataire, montant, date, compte).
<b>SC-8.2</b>	TOM-13	Vérifier que les transactions sont bien affichées dans l'ordre chronologique décroissant.
<b>SC-8.3</b>	TOM-13	Vérifier le comportement si l'API ne répond pas ou est trop lente (>30 sec).
<b>SC-8.4</b>	TOM-13	Vérifier que l'utilisateur est notifié en cas d'erreur ou d'indisponibilité des données.
<b>SC-8.5</b>	TOM-13	Vérifier qu'un utilisateur avec plusieurs comptes voit bien les transactions réparties par compte et par banque.
<b>SC-8.6</b>	<b>TOM-13</b>	Vérifier que les transactions ne sont pas stockées en BDD.

## Sprint 5 – Scénarios prévus

**TOM-14 – Affichage graphique de consommation** (Criticité : Moyenne)

Scénario	Exigence vérifiée	Objectif
<b>SC-9.1</b>	TOM-14	Vérifier que le graphique type camembert s'affiche bien sur la page dédiée "Suivi consommation"
<b>SC-9.2</b>	TOM-14	Vérifier que les dépenses sont bien catégorisées (alimentation, logement, énergie...).
<b>SC-9.3</b>	TOM-14	Vérifier que les données utilisées pour le graphique proviennent bien de toutes les banques connectées utilisateur.
<b>SC-9.4</b>	TOM-14	Vérifier que la plage de dates sélectionnée modifie bien les données du graphique.
<b>SC-9.5</b>	<b>TOM-14</b>	Vérifier que le bouton "Retour au tableau de bord" redirige correctement.

## Sprint 6 – Scénarios prévus

**TOM-15 – Interface du conseiller virtuel** (Criticité : Moyenne)

Scénario	Exigence vérifiée	Objectif
<b>SC-10.1</b>	TOM-15	Vérifier que l'icône d'accès au conseiller est accessible depuis une bulle présente sur toutes les pages de l'application.
<b>SC-10.2</b>	TOM-15	Vérifier que le clic sur la bulle ouvre une fenêtre de discussion s'ouvre proposant 2 options - Conseil sur les consommations - Conseil sur l'épargne
<b>SC-10.3</b>	TOM-15	Vérifier que la sélection d'une des options redirige vers la page de prise de rendez-vous avec un conseiller.
<b>SC-10.4</b>	<b>TOM-15</b>	Vérifier que le système affiche un message clair en cas d'erreur de redirection

## Méthodes de test adaptées

Scénario	Méthode de test	Justification
<b>SC-1 : Création de compte - Sprint 2</b>	Tests manuels	<b>Vérifier les champs (nom, prénom, date, mot de passe), vérifier les règles de validation et la redirection correcte.</b>
	Test Exploratoire	<b>Tester des saisies inattendues (ex : champ trop long, injection de code, champs inversés, etc.).</b>
	Tests User Interface	<b>Contrôler l'accessibilité, la lisibilité, la cohérence des champs et des boutons dans différents navigateurs et résolutions.</b>
	Tests automatisés	<b>Automatiser les cas de saisies valides et invalides.</b>
<b>SC-2 : Téléversement de documents -Sprint 2</b>	Tests manuels	<b>Vérifier le comportement de l'interface lorsqu'on ajoute, annule ou remplace un fichier (formats autorisés, messages).</b>
	Test Exploratoire	<b>Tester des cas limites : fichier trop lourd, mauvaise extension, clics répétés, annulation.</b>
	Tests automatisés	<b>Automatiser les flux standards du téléchargement pour assurer la stabilité du processus.</b>
<b>SC-3 : Authentification (2FA) -Sprint 1</b>	Tests automatisés	<b>Vérifier les étapes de connexion, d'appel à la brique SECU, et la redirection selon la réponse API.</b>

	Test manuel	<b>Vérifier les cas où l'utilisateur reçoit le code et peut le saisir, ou ne le reçoit pas, ou le code est invalide.</b>
	Tests exploratoires	<b>Identifier d'éventuelles failles dans la logique de double authentification ou mauvaise gestion</b>
<b>SC-4 : Authentification biométrique -Sprint 1</b>	Tests Manuels	<b>Vérifier la détection de la biométrie sur différents équipements et la redirection en cas de succès ou d'échec.</b>
	Tests exploratoires	<b>Tester les cas où la biométrie échoue (empreinte inconnue, capteur bloqué), ou n'est pas disponible sur le terminal.</b>
<b>SC-5 : Récupération de données bancaires -Sprint 3</b>	Tests d'intégration de l'API	<b>Vérifier l'appel à l'API Banque de France, la structure des données reçues, leur enregistrement.</b>
	Tests exploratoires	<b>Vérifier le comportement de l'application lorsque la réponse API contient des champs absents ou inattendus</b>
	Tests de charge et performance	<b>Tester les limites (timeouts, surcharge) et messages d'erreur liés au dépassement des délais ou quotas.</b>
<b>SC-6 : Tableau de bord - Sprint 3</b>	Tests UI	<b>Contrôler l'affichage des blocs par banque, la lisibilité des informations et la cohérence graphique.</b>
	Tests automatisés	<b>Vérifier que le bouton 'Rafraîchir' recharge bien les données dans les bonnes conditions.</b>
<b>SC-7 : Téléchargement des relevés - Sprint 3</b>	Tests manuels	<b>Vérifier que le clic sur "Télécharger" génère bien un fichier PDF avec un nom lisible et un contenu valide.</b>

	Tests exploratoires	<b>Tester les cas limites : clics multiples, téléchargement vide ou corrompu, lien inactif, absence de réponse API.</b>
	Tests de non-régression	<b>Vérifier que l'ajout de cette fonction ne perturbe pas l'affichage du tableau de bord.</b>
<b>SC-8 : Interconnexions bancaires et transactions - Sprint 4</b>	Tests d'intégration API	<b>Vérifier que l'appel GET fonctionne, que les données renvoyées sont bien mappées, et gérées en cas d'erreur</b>
	Tests automatisés	<b>Vérifier que le bon nombre de lignes est affiché en fonction des données API. Réutilisé pour non-régression</b>
	Tests exploratoires	<b>Simuler des cas d'erreur de réponse, d'utilisateur avec plusieurs banques...</b>
<b>SC-9 : Suivi de consommation -Sprint 5</b>	Tests manuels	<b>Vérifier l'affichage du graphique (bonne répartition des catégories, montants visibles, légende correcte).</b>
	Tests UI	<b>S'assurer du bon affichage du graphique camembert sur desktop et mobile (responsive).</b>
<b>SC-10 : Conseiller virtuel (Sprint 6</b>	Tests manuels	<b>Vérifier que la bulle est accessible depuis toutes les pages et mène bien à la prise de rendez-vous.</b>
	Tests exploratoires	<b>Tester les comportements si l'utilisateur change de page, ou si le service est momentanément indisponible.</b>

## Estimation de la charge de production des scénarios

La charge de production des cas de test a été estimée à **70 heures**, soit environ **9 jours ouvrés** à temps plein pour un testeur fonctionnel.

Cette estimation repose sur un abaque de complexité, où :

- **Un scénario complexe nécessite en moyenne 8 heures de travail**
- **Un scénario de complexité moyenne requiert environ 6 heures**

Les 10 scénarios planifiés dans le cadre du projet (SC-1 à SC-10) ont été évalués en fonction de leur complexité, répartis sur les 6 sprints QA.

La production est réalisée de manière progressive, en parallèle du développement agile.

Scénario	Complexité	Conception	Exécution	Total (en h)
SC-1	Complex	2	6	8
SC-2	Moyen	1,5	4,5	6
SC-3	Complex	2	6	8
SC-4	Complex	2	6	8
SC-5	Complex	2	6	8
SC-6	Moyen	1,5	4,5	6
SC-7	Moyen	1,5	4,5	6
SC-8	Complex	2	6	8
SC-9	Moyen	1,5	4,5	6
SC-10	Moyen	1,5	4,5	6

Total (en h)	70,00
Total (en jours)	9

# Ressources nécessaires

Afin de mener à bien la campagne de tests dans de bonnes conditions, les ressources suivantes sont requises.

## 1. Compétences requises pour la campagne de tests

- **Testeur fonctionnel**  
Chargé de l'exécution des tests manuels, de la documentation des anomalies, ainsi que de la gestion et de la préparation des jeux de données.
- **Testeur en automatisation (QA Automation)**  
Responsable de la conception, de la maintenance et de l'exécution des tests automatisés sur les parcours critiques (authentification, tableau de bord, téléversement).
- **Développeur**  
En charge de la correction rapide des anomalies détectées et de la collaboration étroite avec l'équipe QA pour assurer la qualité du produit.
- **Product Owner / Business Analyst**  
Garant de la validation des cas d'usage et de la conformité fonctionnelle et métier des scénarios de test.
- **Responsable sécurité**  
Sollicité pour les tests impliquant des données sensibles, la gestion des accès ou les mécanismes d'authentification avancés.

## 2. Éléments nécessaires pour l'environnement de test

- **Environnement de préproduction strictement identique à la production** (base de données, APIs actives, mécanismes d'authentification, configuration réseau).
- **Instance stable de l'application Web**, accessible à l'ensemble de l'équipe QA.
- **Accès aux services tiers et APIs externes :**
  - API Banque de France
  - API SECU pour la gestion du 2FA
- **Outils de test :**
  - Navigateurs à jour (Chrome, Firefox)
  - Postman pour les tests d'API
  - Outils d'automatisation (Cypress ou Selenium)
- **Accès aux logs applicatifs et à une base de données de test dédiée.**
- **Outil de gestion des tests**  
(outil spécialisé ou, à défaut, solution structurée telle que Notion ou un tableau partagé).

### 3. Usage des tests automatisés

Les tests automatisés sont fortement recommandés pour les parcours critiques et récurrents, notamment :

- **Authentification classique et 2FA (SC-3)**
- **Rafraîchissement du tableau de bord (SC-6)**
- **Téléversement de fichiers (SC-2)**

Ces scénarios sont :

- Fréquemment exécutés,
- Essentiels au bon fonctionnement du produit,
- Simples à maintenir sur la durée.

Les tests automatisés permettent :

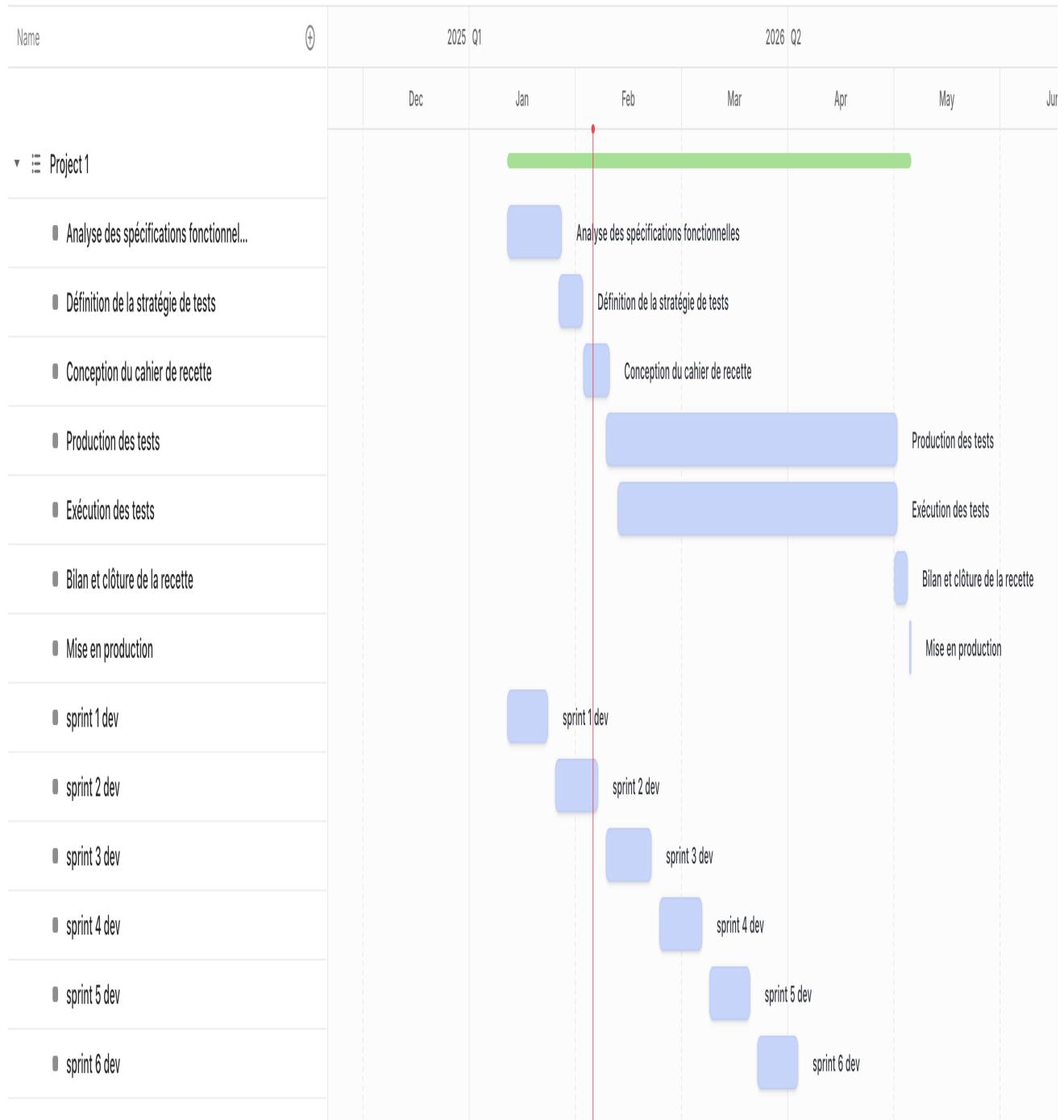
- Un gain de temps significatif lors des phases de recette,
- Une réduction du risque de régression,
- Une couverture de test fiable et reproductible.

En revanche, les scénarios impliquant le **conseiller virtuel** (SC-10) ou **l'authentification biométrique** (SC-4) seront privilégiés en **tests manuels**, en raison de leur complexité et de leur dépendance aux interactions système et matérielles.

## Étapes clés de la stratégie

La campagne de tests est structurée en plusieurs phases clés, réparties sur une période de quatre mois, de janvier à Mai 2026. Elle s'inscrit pleinement dans la méthodologie Agile adoptée par l'équipe projet.

Chaque phase correspond à un jalon majeur du cycle de validation et est alignée avec le plan de sprints défini par le Product Owner, garantissant ainsi une intégration continue des activités de test au rythme des développements.



## 1. Analyse des spécifications fonctionnelles

### Date estimée : 12 janvier 2026

Cette phase consiste à analyser en détail les exigences fonctionnelles du produit afin de comprendre les besoins métier, identifier les fonctionnalités critiques et définir les scénarios de test prioritaires. Elle permet d'anticiper les risques et d'assurer une couverture de test pertinente dès le démarrage.

## 2. Définition de la stratégie de tests

### **Date estimée : 27 janvier 2026**

La stratégie de tests vise à définir l'approche globale de la campagne : types de tests à réaliser (fonctionnels, automatisés, API), outils utilisés, critères de priorisation et organisation des activités par sprint. Cette étape garantit une vision claire et partagée par toutes les parties prenantes.

## 3. Conception du cahier de recette

### **Date estimée : 3 février 2026**

Cette étape permet de formaliser l'ensemble des cas de test en précisant pour chacun les objectifs, les prérequis, les étapes d'exécution et les critères de validation. Le cahier de recette constitue la référence principale pour l'exécution des tests.

## 4. Production des tests

### **Période : du 9 février au 1 mai 2026**

Les scénarios de test sont rédigés progressivement au fil des sprints, en tenant compte de la complexité fonctionnelle et des évolutions du produit. Cette phase inclut la préparation des jeux de données et, le cas échéant, l'écriture des tests automatisés.

## 5. Exécution des tests

### **Période : du 12 février au 1 mai 2026**

Les tests sont exécutés en continu, en parallèle des développements. Cette phase comprend les tests manuels, automatisés et API, ainsi que le suivi des résultats, la remontée des anomalies et leur revalidation après correction.

## 6. Bilan et clôture de la recette

### **Date estimée : 1 mai 2026**

Un bilan global de la campagne de tests est réalisé afin d'analyser les résultats, évaluer la qualité du produit et traiter les anomalies restantes. Cette étape inclut la rédaction du procès-verbal de recette et la validation finale côté métier.

## 7. Mise en production

### Date estimée : 5 mai 2026

Après validation complète de la recette fonctionnelle, le produit est déployé en production. Cette phase marque la fin de la campagne de tests et le passage en exploitation.

## Préconisations

### 1. Sur le planning serré et le découpage Agile

#### Contexte et justification

Le planning du projet repose sur une organisation en sprints, avec une exécution des tests réalisée en parallèle des développements. Cette approche Agile favorise la réactivité, mais rend également les activités de test fortement dépendantes de l'avancement des développements.

#### Risque identifié

Tout retard sur un sprint peut avoir un impact direct sur la préparation, la rédaction ou l'exécution des tests (manuels et automatisés). Cela peut entraîner une réduction de la couverture de test, une exécution précipitée de certains scénarios ou un report des validations critiques.

#### Préconisations

- Mettre en place une revue intermédiaire hebdomadaire entre les équipes Développement et QA afin d'anticiper les blocages et d'ajuster les priorités.
- Appliquer une priorisation stricte des scénarios de tests critiques, en lien avec les parcours métier à fort enjeu.
- Intégrer une marge de sécurité minimale de **deux jours** avant le jalon final de mise en production afin d'absorber les aléas de dernière minute.

### 2. Sur les tests automatisés

#### Contexte et justification

L'automatisation est prévue uniquement sur les parcours critiques du produit afin de sécuriser les fonctionnalités les plus sensibles et les plus utilisées.

#### Risque identifié

Si les outils, les environnements ou les ressources dédiées à l'automatisation ne sont pas opérationnels suffisamment tôt, les tests automatisés risquent de ne pas être intégrés à temps dans la campagne, réduisant leur valeur ajoutée.

## Préconisations

- Définir et concevoir les scripts automatisés dès le sprint **2 (semaine 4)** même si leur exécution est planifiée ultérieurement.
- S'appuyer sur un outil maîtrisé par l'équipe QA (Cypress, Selenium ou équivalent) afin de limiter la courbe d'apprentissage.
- Prévoir une phase de validation technique (environnement, données, accès) avant le sprint **5 (semaine 9,)** pour garantir une exécution fluide des tests automatisés.

## 3. Sur la gestion des anomalies

### Contexte et justification

Le suivi rigoureux des anomalies est un élément clé pour piloter la campagne de tests et ajuster en continu le plan de validation.

### Risque identifié

Un volume important d'anomalies non suivies, mal qualifiées ou insuffisamment priorisées peut désorganiser la fin de campagne et retarder la clôture de la recette.

## Préconisations

- Mettre en place un outil de suivi partagé (Notion, Excel ou équivalent) intégrant les statuts, la priorité, le responsable et la date de correction.
- Désigner un référent QA chargé de la validation des corrections et de la cohérence globale du backlog d'anomalies.
- Organiser des points réguliers de revue des anomalies avec l'équipe de développement.

## 4. Sur la campagne de non-régression

### Contexte et justification

Une campagne de non-régression est indispensable pour garantir la stabilité du produit après l'enchaînement de plusieurs sprints et l'évolution continue des fonctionnalités critiques.

### Risque identifié

Sans rejouabilité des scénarios critiques, des régressions fonctionnelles peuvent être introduites sur des fonctionnalités déjà validées, sans être détectées avant la mise en production.

## Préconisations

- Planifier une campagne complète de non-régression en fin de cycle, avant la mise en production.

- Maintenir un socle de scénarios automatisés permettant une exécution régulière et rapide des parcours critiques.
- Compléter l'automatisation par des tests manuels ciblés sur les fonctionnalités sensibles.

## 5. Sur la documentation des tests

### **Contexte et justification**

Une documentation QA claire et à jour est essentielle pour assurer la traçabilité des validations, faciliter les relectures et garantir la continuité en cas de changement de ressources.

### **Risque identifié**

Une documentation incomplète ou obsolète peut entraîner des oubli de tests, des validations non justifiées ou des difficultés lors des bilans de recette.

### **Préconisations**

- Mettre à jour régulièrement le cahier de recette, le tableau de suivi des anomalies et le journal d'exécution des tests.
- Centraliser l'ensemble des documents QA sur un outil collaboratif partagé (Notion, Google Drive ou équivalent).
- Standardiser les formats de documentation pour améliorer la lisibilité et la réutilisabilité.

## 6. Sur la gestion des environnements de test

### **Contexte et justification**

Les écarts entre l'environnement de test (préproduction) et l'environnement de production sont une source fréquente de bugs non détectés en phase de recette.

### **Risque identifié**

Une fonctionnalité validée en test peut échouer en production si les configurations diffèrent (authentification, API, paramètres serveurs, données).

### **Préconisations**

- Garantir un alignement maximal entre l'environnement de test et la production (configuration, services, sécurité).
- Utiliser des jeux de données réalistes et représentatifs des cas métier.
- Valider dès le début de la campagne les accès à l'ensemble des API tierces et services externes.