



Stratégie globale

Confidentiel © [20XX] CGI inc.



Table des matières

1	Introduction	4
1.1	Présentation	4
1.2	Contexte projet	4
1.3	Termes et abréviations	4
1.4	Références	5
2	Organisation des tests	6
2.1	Domaines de test	6
2.2	Phases de tests	6
2.3	Enchaînement des phases de test	7
2.4	Identification des parties prenantes	7
2.4.1	Équipe projet XXXX (XXXX)	7
2.4.2	Équipe PQHP (XXXX)	7
2.4.3	Équipe CD spécialistes (XXXX)	7
2.4.4	Équipe Design Système (EY)	7
2.4.5	Équipe AMOA (PwC)	8
2.4.6	Équipe interne XXXX (Autres SI)	8
2.4.7	Équipe partenaire des SI nationaux	8
2.4.8	Équipe de développement externe XXXX (MOE XXXX)	8
2.4.9	Équipe de recette TRA (CGI)	8
2.4.10	Équipe migration	8
2.4.11	Équipe accessibilité (ATALAN)	8
2.5	Organisation	9
2.6	Rôles et responsabilités	9
2.7	Méthode de test	10
2.8	Organisation des itérations/incréments	10
2.8.1	Les itérations	10
2.8.2	Les incréments	10
2.8.3	Le sprint planning	10
2.8.4	Definition of Ready (DoR)*	10
2.8.5	Definition of Done (DoD)*	10
2.8.6	Planning type	12
2.9	Gestion des environnements	13
2.10	Gestion des anomalies	13
2.10.1	Sévérité	13
2.10.2	Criticité	14
2.10.3	Priorité	14

2.10.4	Workflow	14
2.10.5	Statuts JIRA	14
2.10.6	Suivi des anomalies	15
2.10.7	Bonnes pratiques	16
2.11	Outillage de test	16
2.11.1	Outil de gestion des tests TRA : Squash TM	16
2.11.2	Outil de gestion des anomalies : JIRA	16
2.11.3	Outils de gestion des tests hors TRA/ hors XXXX	16
3	Gestion des risques	17
3.1	Méthodologie de test RRBT	17
3.2	Les risques projet	18
3.3	Les risques produit	18
4	Périmètre et phases de test	19
4.1	Tests Unitaires	19
4.2	Tests Intégration usine	19
4.3	Recette fonctionnelle	20
4.4	Tests d'interface techniques	22
4.5	Tests d'interface fonctionnels	23
4.6	Tests Usine Migration	24
4.7	Certification statique	24
4.8	Certification dynamique	25
4.9	Tests de performance	26
4.10	Tests d'accessibilité	27
4.11	Tests de validation métier	27
5	Gouvernance des tests	28
5.1	Comitologie	28
5.2	Indicateurs	28
5.2.1	Indicateurs de suivi	28
5.2.2	Indicateurs de performance	29

Introduction

Présentation

Ce document est établi dans le cadre du projet XXXX. Ce projet a pour but la mise en place d’un système d’information national pour la gestion de l’allocation personnalisée d’autonomie au sein de la Caisse Nationale de Solidarité pour l’Autonomie.

Ce document présente la stratégie de test globale définie au niveau du programme : elle est donc transverse à l’ensemble des différents chantiers. Elle va permettre de cadrer les périmètres d’intervention de chaque équipe et d’assurer une complémentarité des niveaux de tests adressés.

L’objectif de la stratégie globale est d’assurer une couverture de tests maximale et non redondante.

Contexte projet

L’objectif du projet XXXX est de mettre en place un système d’information national pour la gestion de l’allocation personnalisée d’autonomie domicile.

Actuellement, cette allocation est gérée de manière départementale à l’intérieur du SI Aides sociales.

La solution cible consiste à obtenir une gestion uniformisée de cette allocation et de conserver les flux actuels qu’ils soient avec des interfaces départementales ou nationales.

Termes et abréviations

Terme/abréviation	Définition
Anomalies	Un fonctionnement (fonctionnel, technique ou documentaire) non conforme au résultat attendu, constaté lors de l’exécution des tests.
Batch	Enchaînement automatique de commandes sans intervention d'un opérateur.
Parties prenantes	Elles sont définies ici comme les entités concernées par le projet et potentiellement émettrices d’exigences ou risques.
PQHP	Pôle Qualité et Harmonisation des Pratiques
RRBT	Risk and Requirements Based Testing. Il s’agit de la matrice des risques et des exigences à partir de laquelle on va définir la stratégie de tests.
SFD	Spécification Fonctionnelle Détaillée
STD	Spécifications Techniques Détaillée
TU	Tests Unitaires
TI	Tests d’Intégration
TS	Tests Système
PO	Product Owner
US	User stories
DoR	Definition of Ready
DoD	Definition of Done

Références

Titre	Date	Version

Organisation des tests

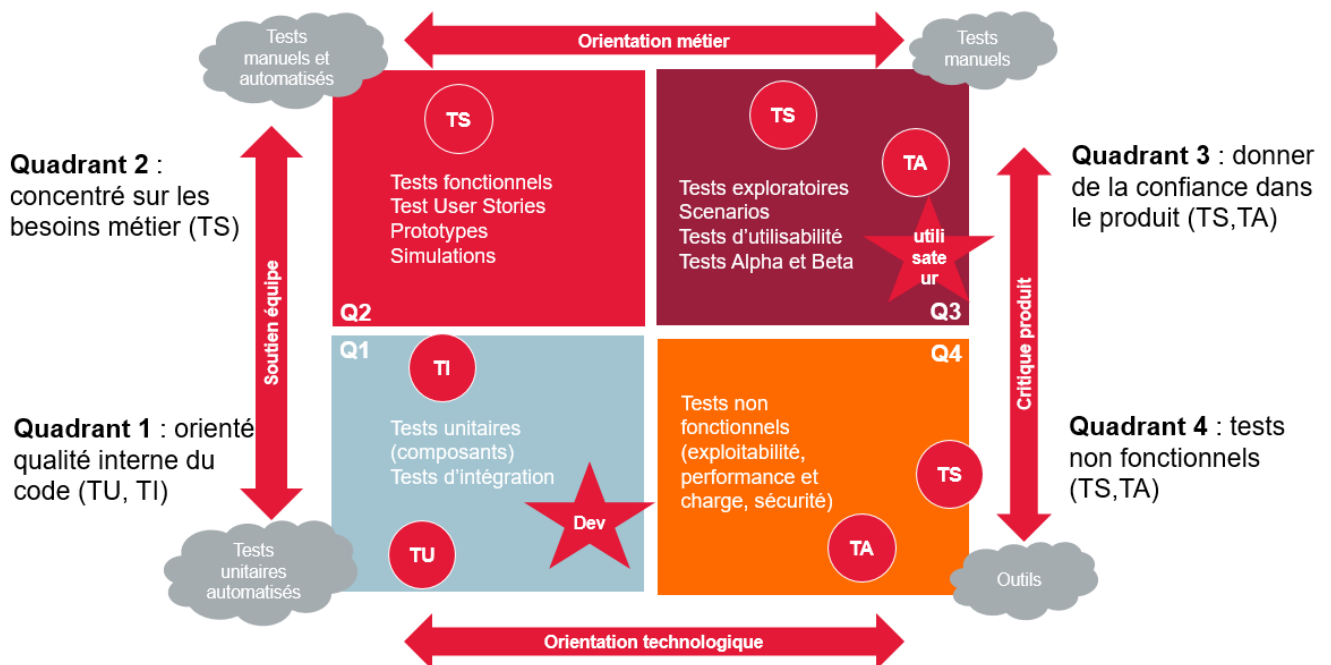
Domaines de test

Le projet se déroulera en plusieurs phases de tests successives. Pour chaque phase, il conviendra de définir l'organisation, les conditions d'entrées/sorties, les pré-requis, les entrants, les livrables et les outils nécessaires.

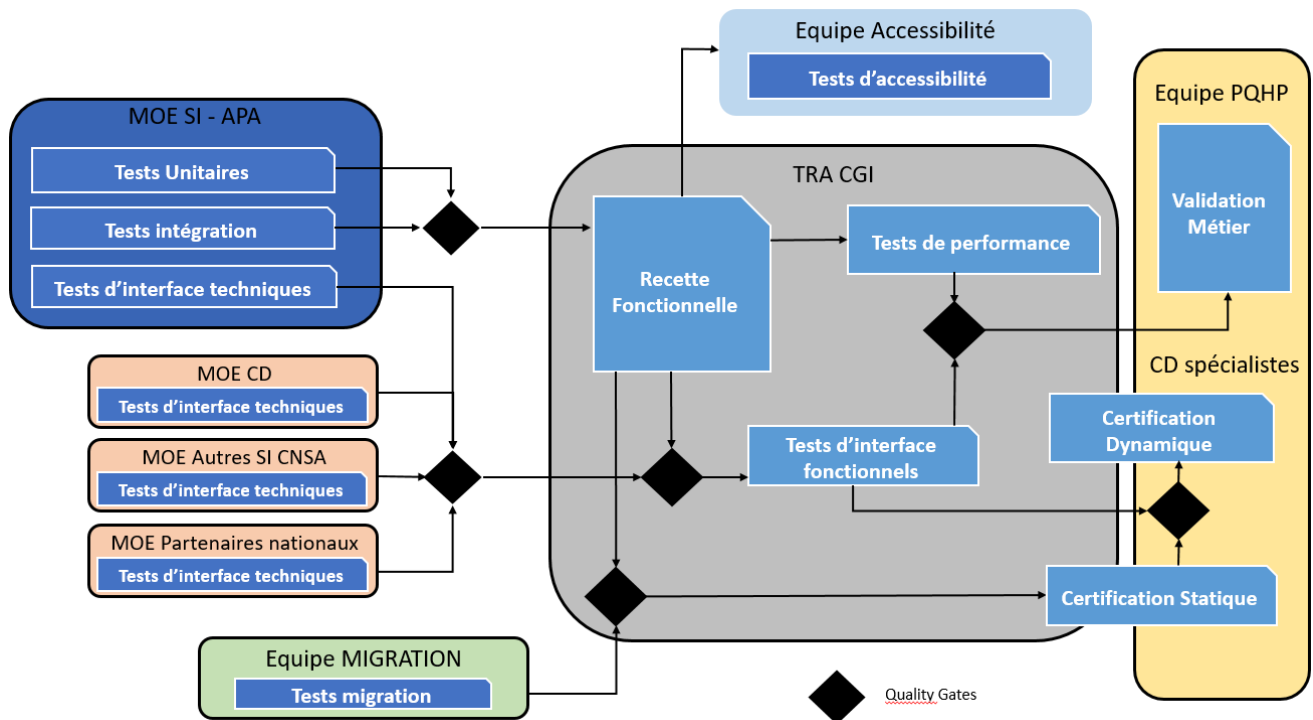
Les phases ci-dessous ont été planifiées pour ce projet :

- Tests Unitaires
- Tests Intégration usine
- Recette fonctionnelle
- Tests d'interface (techniques et fonctionnels)
- Tests d'interface partenaire (départementaux ou nationaux)
- Tests Usine Migration
- Certification statique & dynamique
- Tests performance
- Tests d'accessibilité
- Tests de validation métier

Phases de tests



Enchainement des phases de test



Identification des parties prenantes

Équipe projet XXXX (XXXX)

- Direction de programme
- Responsable des opérations IT
- 2 chefs de projet
- Un référent pour chaque phase de test (référent fonctionnel, métier, interface...)

Équipe PQHP (XXXX)

- Experts métier nationaux
- Utilisateurs finaux

Équipe Conseils Départementaux

- CD spécialistes (expertise métier, key users...)
- MOE interne
- MOE externe

Équipe Design Système (EY)

- Un chef de projet
- Une équipe de développement

- Une équipe de test

Équipe AMOA (PwC)

- Un chef de projet
- Des PO
- Des référents fonctionnels

Équipe interne XXXX (Autres SI)

- Un référent pour chaque SI
- Des référents techniques

Équipe partenaire des SI nationaux

- Référent technique
- Référent métier
- Équipe de développement
- Équipe de test

Équipe de développement externe XXXX (MOE XXXX)

- Direction de projet et chef de projet
- Tests manager
- Référents technique (référent solution, interface, performance...)
- Équipe de Développeurs/testeurs

Équipe de recette TRA (CGI)

- Direction de projet
- Chef de projet testing et pilote de recette
- Tests manager
- Référents fonctionnels & interface
- Référents techniques (référent performance...)
- Équipe de testeurs

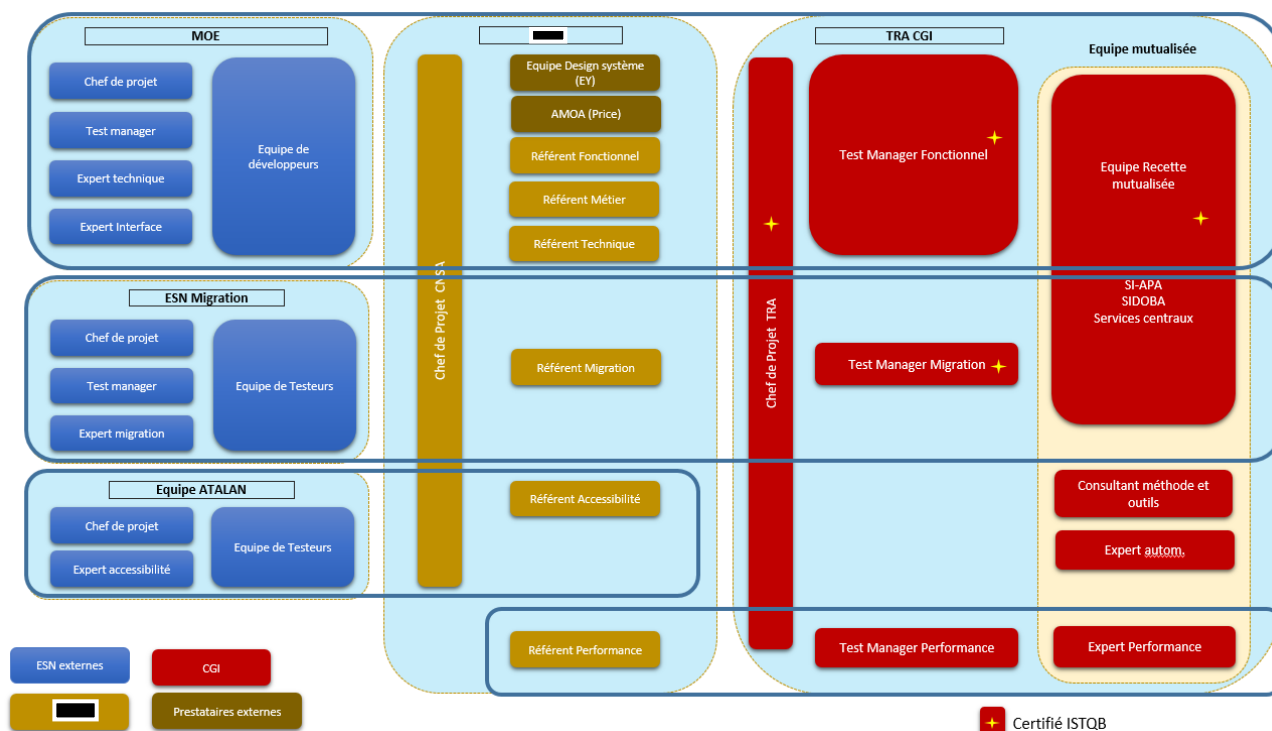
Équipe migration

- Direction de projet et chef de projet
- Test manager
- Expert migration
- Équipe de testeurs

Équipe accessibilité (ATALAN)

- Chef de projet
- Experts accessibilité
- Équipe de Testeurs

Organisation



Rôles et responsabilités

Le tableau ci-dessous présente les rôles et responsabilités associés à chaque étape de tests :

	XXXX	MOE	CGI	CD spécialistes	ESN migration	Équipe PQHP	ATALAN
Tests Unitaires	C	RA	I			I	
Tests Intégration usine	C	RA	I	R		I	
Tests Usine Migration	C	I	I	C	RA	I	
Recette fonctionnelle	C	I	RA	I		I	
Certification statique & dynamique	I	C	RA	R	C	R	
Tests d'interface technique	C	RA	I	R		I	
Recette d'interface fonctionnelle	C	C	RA	R		R	
Tests performance	I	C	RA			I	
Tests d'accessibilité	I	C	I	I		I	RA
Tests de validation métier	R	I	AC	R		R	

- **R : Réalise**; en charge d'effectuer la tâche - plusieurs R possibles
- **A : Approuve**; en charge de superviser la tâche et de référer à la hiérarchie - 1 seul A
- **C : Consulté**; en charge de contribuer, d'apporter des conseils - plusieurs C possibles
- **I : Informé**; doit être informé - plusieurs I possibles

Méthode de test

La méthode de test utilisée sera la mise en place de cycles en V itérés.

Dans cette méthode, les besoins sont spécifiés en amont comme pour une méthode cycle en V classique. Par contre, les développements sont faits de manière itérative.

De la même façon, les différentes phases de recette sont découpées en itérations.

Cette approche itérative permet de passer d'un mode traditionnel prédictif en V à une approche répondant mieux aux changements fréquents auxquels les projets sont soumis.

Organisation des itérations/incréments

Les itérations

Les itérations ont une durée de 3 semaines.

Les incréments

Chaque incrément est composé de 4 itérations.

Voir chapitre 2.8.6 pour le planning et l'enchaînement des phases de tests entre les itérations et les incréments.

Le sprint planning

Le sprint planning a lieu le premier jour d'une nouvelle itération et dure environ 2h. Il sert à définir le contenu de l'itération et la manière dont le travail va être réalisé. Les participants sont le PO et des experts techniques MOE.

Cet atelier se déroule en deux temps :

- Définir l'objectif et le contenu de l'itération. Quelles US sont concernées.
- Décomposer ces US en tâches.

En fin de réunion, l'équipe dispose d'un plan de travail précis pour l'itération qui démarre.

Definition of Ready (DoR)*

L'objectif est d'avoir toutes les informations nécessaires au développement d'une US.

A minima, les informations attendues sont :

- Un titre explicite
- Une description (ou proposition de valeur)
- Une estimation
- Un test d'acceptation

Les US étant au statut Ready sont éligibles à être intégrées dans une itération

* Concept à définir par les équipes Agile en dehors des phases de recette. La TRA se propose de participer à la DoR.

Definition of Done (DoD)*

L'objectif est de définir les critères essentiels pour qu'une US atteigne le niveau d'exigence souhaité. Cela permet d'aligner les équipes techniques et métiers sur les résultats attendus et acceptables à la fin d'une itération.

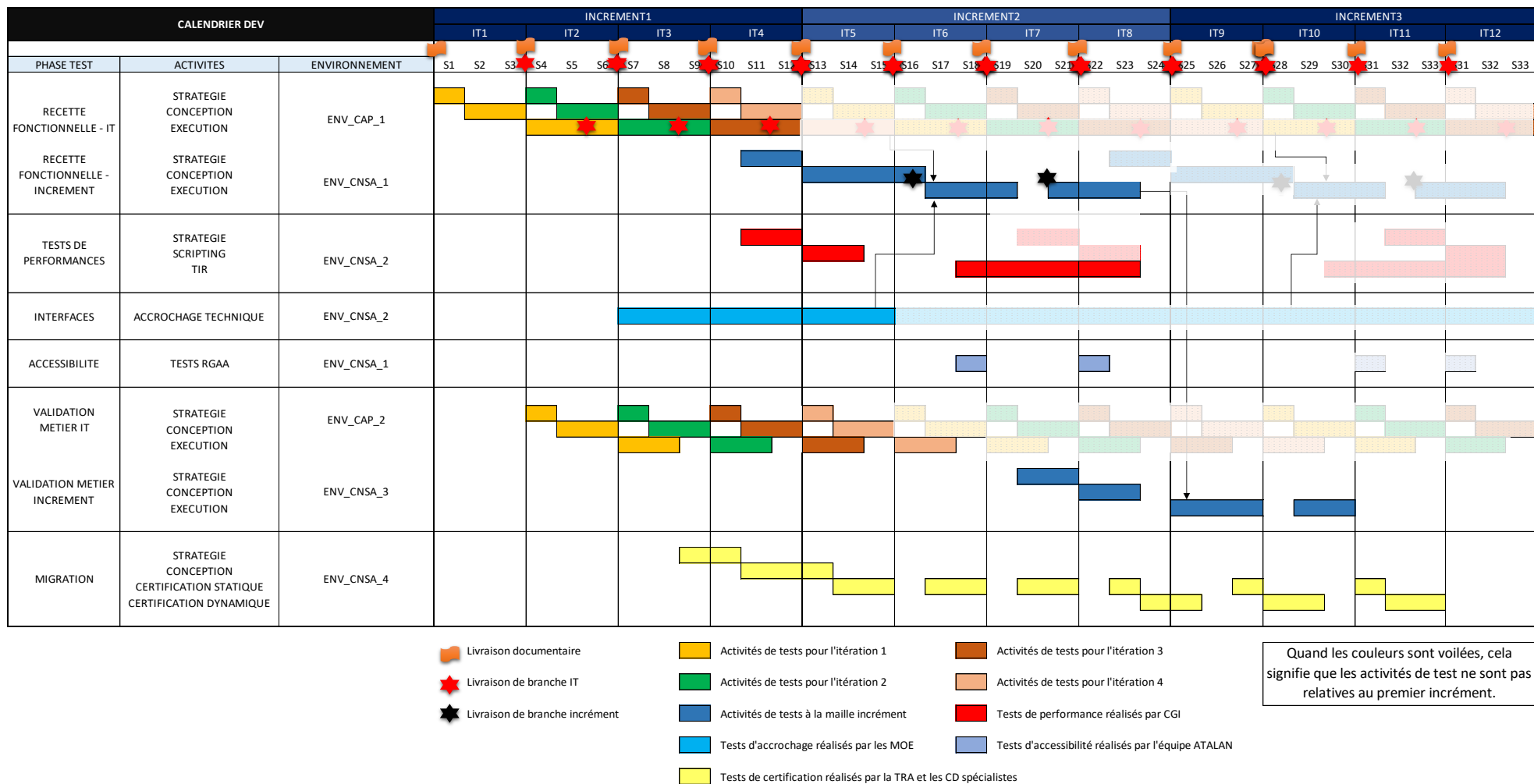
A minima, les critères présents dans la DoD doivent être :

- Revue du code effectuée
- Les critères d'acceptation de l'US sont tous validés
- Les différents tests ont été finalisés et acceptés par les acteurs de la qualité du produit
- La documentation a été mise à jour
- Les maquettes graphiques sont respectées

Les US étant au statut Done sont considérées comme correctes d'un point de vue technique. Elles peuvent être testées d'un point de vue fonctionnel et métier.

*Concept à définir par les équipes Agile en dehors des phases de recette. La TRA se propose de participer à la DoD.

Planning type



Gestion des environnements

Tout au long des différentes phases de test, de nombreux environnements seront utilisés.

	Installation et maintien	Données	Interface
Tests Unitaires	ESN dev	Fabriquées	Bouchonné
Tests Intégration usine	ESN dev	Fabriquées	Bouchonné
Recette fonctionnelle	XXXX	Fabriquées	Intégré
Tests Usine Migration	ESN dev	Migrées	Bouchonné
Certification statique & dynamique	XXXX	Migrées	intégré
Tests d'interface	XXXX	Migrées	Intégré
Tests performance	XXXX	Migrées	Intégré
Tests d'accessibilité	XXXX	Migrées	Intégré
Tests de validation métier	XXXX	Migrées	Intégré

Gestion des anomalies

Le suivi des anomalies se fait dans l'outil de test JIRA.

Les anomalies suivent le workflow qui sera défini lors de la phase de cadrage dans la stratégie de test. Les anomalies sont définies par un couple Sévérité et Criticité qui permet de définir la priorité de correction.

Sévérité

La sévérité est définie au moment de la déclaration de l'anomalie :

- Bloquante : La fonctionnalité ne répond pas aux spécifications et il n'y a pas de solution de contournement. Dans le principe, une anomalie bloquante ne permet pas le passage du projet à l'étape suivante.
- Majeure : La fonctionnalité ne répond pas de manière importante aux spécifications et/ou une solution de contournement est identifiée pour poursuivre les tests. Une anomalie Majeure ne permet normalement pas le passage du projet à l'étape suivante.
- Mineure : Par extension, anomalie ni bloquante ni majeure. L'anomalie mineure ne bloque pas le passage du projet à l'étape suivante, mais peut faire l'objet d'un suivi pour être prise en compte dans les versions suivantes.

Criticité

La criticité est définie à partir de la criticité de la fonctionnalité couverte par le cas de tests en défaut.

Priorité

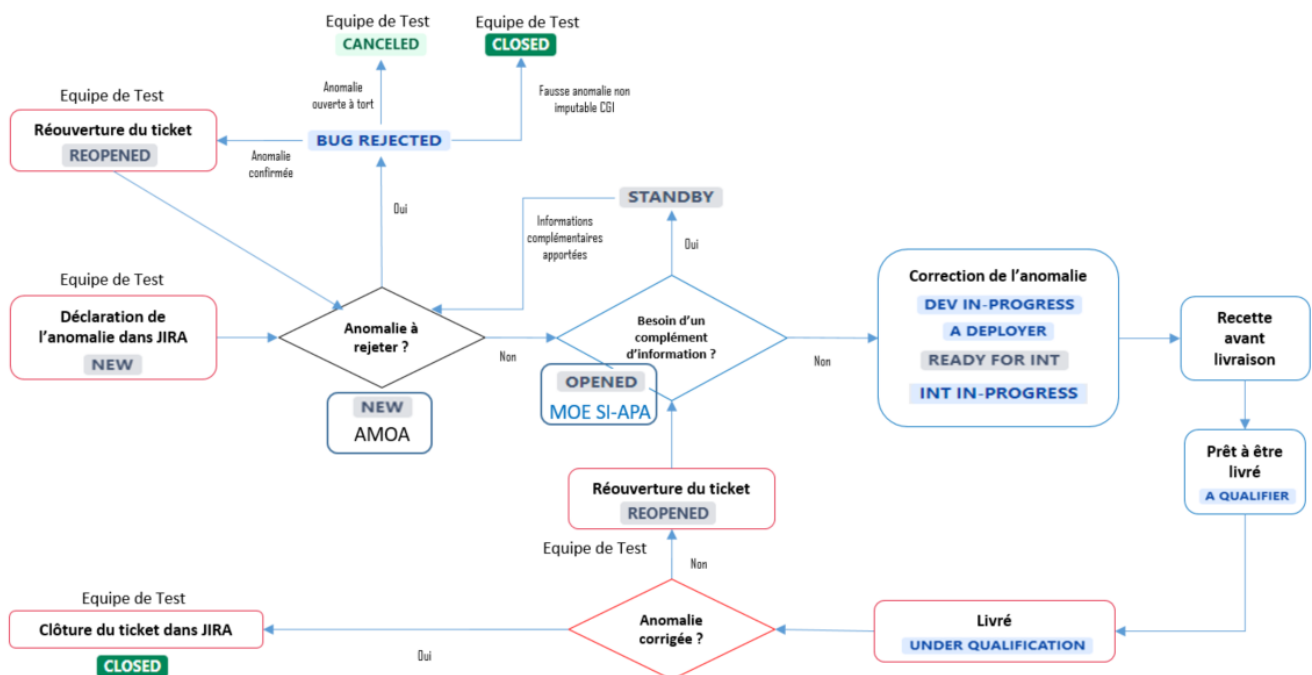
Les anomalies sont systématiquement priorisées : Important, Moyen, Faible

La priorité est la rapidité avec laquelle un bug doit être corrigé par rapport aux autres anomalies déclarées ayant le même degré de sévérité.

La priorisation se fait à partir du tableau ci-dessous :

Sévérité des anomalies			
Criticité		Bloquante	Majeure
	03-Impératif	Important	Important
	02-Important	Important	Moyen
	01-Interessant	Moyen	Moyen
	00-Ignoré	Moyen	Faible

Workflow



Statuts JIRA

Statut des tickets concernés par les équipes de test :

- NEW : Statut initial à la création de l'anomalie. Suite à la création, le ticket est affecté à un référent AMOA pour validation.
- STANDBY : Ticket renvoyé à l'équipe de test par l'équipe de développement pour des informations complémentaires.
- BUG REJECTED : L'anomalie est refusée par l'équipe MOE. Suite à ce rejet l'anomalie peut être :
 - Rouverte (REOPENED) si on estime que l'anomalie est bien présente
 - Annulée (CANCELED) si l'anomalie a été ouverte à tort
 - Fermée (CLOSED) si l'anomalie n'est pas confirmée mais que l'équipe de test n'est pas responsable (mauvaise documentation fournie, problème d'environnement...)
- UNDER QUALIFICATION : Anomalie livrée. A qualifier.
- REOPENED :
 - Ticket rouvert suite à un premier rejet (statut précédent : BUG REJECTED).
 - Ticket rouvert suite à la qualification du correctif qui n'est pas conforme (statut précédent : UNDER QUALIFICATION)
- CANCELED : Anomalie annulée
- CLOSED : État final de l'anomalie.
 - La qualification du correctif est satisfaisante.
 - L'anomalie n'est pas avérée mais n'a pas été ouverte à tort

Statut des tickets concernés par l'équipe AMOA :

- NEW : Un ticket est affecté à l'AMOA suite à sa création. Si l'anomalie est confirmée, elle est affectée à l'équipe MOE XXXX au statut OPENED.

Statut des tickets concernés par l'équipe de développement (MOE XXXX):

- OPENED : Le ticket est soumis à l'équipe de développement pour analyse.
- EVOLUTION : L'équipe de développement considère qu'il ne s'agit pas d'un simple correctif mais d'une évolution
- DUPLICATE : Ticket en doublon
- DEV-IN-PROGRESS : La correction est en cours
- READY FOR INT : Correctif prêt à être qualifié sur l'environnement de développement.
- INT IN PROGRESS : Recette en cours
- A QUALIFIER : Recette concluante sur l'environnement de dev. Le correctif est prêt à être déployé sur l'environnement de recette.

Suivi des anomalies

Voir comitologie (paragraphe 5.1)

Bonnes pratiques

Lors de la création d'une anomalie

- Ajouter (si pertinent) une copie d'écran (outil capture disponible sur les postes)
- En pièces jointes, ajouter également tous les éléments liés au cas de test en erreur : fichier de paramétrage, fichier de ciblage/échantillonnage, flux, logs...
- Dans la description de l'anomalie, indiquer systématiquement : le profil utilisé pour faire le geste, et la référence de l'objet manipulé.
- Un Template sera proposé pendant la phase de cadrage

Gestion des anomalies

- Seul le créateur d'une anomalie est autorisé à clore ou abandonner cette anomalie.
- Les états « Canceled » et « Closed » sont les seuls états finaux dans le workflow de gestion des anomalies. Notamment, l'état « Bug rejected » est soumis à validation par le créateur de l'anomalie.
- Un ticket fermé (closed ou canceled) ne doit jamais être rouvert.
- Afin d'avoir un meilleur suivi de l'évolution d'une anomalie, il est utile de laisser un commentaire à chaque modification de la fiche anomalie (statut, niveau d'importance, niveau de priorité, ajout d'une pièce jointe...).
- Toute modification d'une anomalie entraîne une notification mail au créateur et aux observateurs.

Outillage de test

Afin d'assurer le succès du projet, nous préconisons l'utilisation des outils suivants :

Outil de gestion des tests TRA : Squash TM

- Gestion des exigences, afin de consolider le fonctionnement attendu de l'application sous tests sans avoir à consulter plusieurs US issues d'itérations différents
- Gestion des scripts de tests (étapes à dérouler et résultats attendus)
- Trace de l'exécution des tests (étapes déroulées et résultats obtenus vs résultats attendus)
- Traçabilité bidirectionnelle entre ces objets et avec les défauts
- Suivi des indicateurs d'avancement des tests



Outil de gestion des anomalies : JIRA

- Mis en place par l'intégrateur
- Documentation des écarts détectés lors de l'exécution et identification des défauts
- Gestion du workflow de résolution des défauts
- Traçabilité avec les exécutions, les tests et les exigences



Outils de gestion des tests hors TRA/ hors XXXX

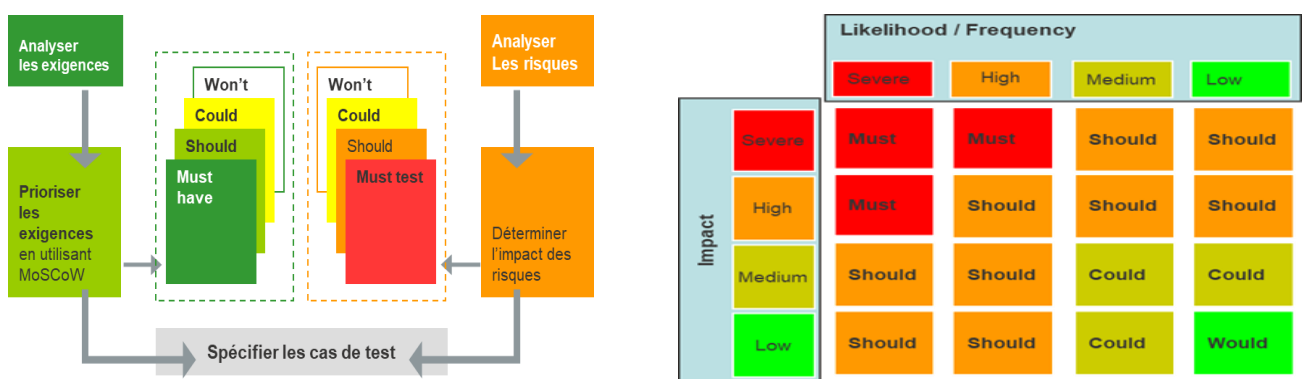
- A discrétion des ESN

Gestion des risques

Méthodologie de test RRBT

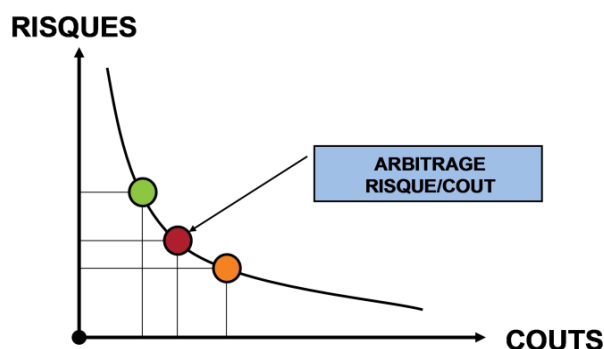
La méthodologie RRBT s'applique à gérer les risques produit et les exigences métiers en intégrant les différents modes possibles de couverture des risques et leur niveau de criticité, pour répondre au mieux aux exigences des différentes parties prenantes concernées. L'objectif principal est d'orienter les activités de test et recette vers ce qui est essentiel pour le client et les fonctionnalités critiques avant l'accessoire.

La phase d'analyse RRBT permet de produire une « Matrice RRBT ou effort de tests », qui sert de base de travail pour appliquer la meilleure stratégie de test en fonction des contraintes de planning ou de budget. Le niveau de criticité est déterminé pour chaque couple risque / besoin métier en fonction de la criticité du risque et de l'importance du besoin



Les apports de cette méthode sont multiples :

- Faire porter l'effort de test sur les fonctions les plus importantes
- Prioriser les scénarios dès la phase de conception des cas de test afin d'identifier les anomalies importantes au plus tôt
- Déterminer les tests de non régression et prioriser leur automatisation
- Piloter la recette et les décisions avec une vision métier complémentaire à la vision applicative
- Maîtriser le planning et le budget en définissant la meilleure stratégie de test par rapport à ces contraintes



Les risques projet

Ce sont les risques qui menacent la capacité du projet à atteindre ses objectifs. Ce sont les risques techniques, les risques humains, les risques planning...

Niveau	Libellé	Conséquence	Actions préventives / correctives
Critique	Gestion de la documentation	Des travaux de conception et/ou d'exécution réalisés à partir de documents obsolètes	Communication de toutes les modifications dans les livrables. Dépôt de la documentation dans un répertoire accessible à toutes les parties prenantes.
Critique	Phase de test d'interface technique trop tardive	Impossibilité pour l'équipe fonctionnelle de réaliser la recette d'interface en respectant les jalons.	Planifier des tests d'accrochage technique dès les premières itérations.
Critique	Responsabilité des éditeurs dans la gestion des interfaces	Interfaces non opérationnelles	Définir au plus tôt les responsabilités entre les éditeurs des SI legacy et l'éditeur du XXXX.
Majeur	Problème d'installation ou de mise à disposition des environnements de test.	Environnements de test non disponibles pour la recette → risque de non-respect du jalon de fin	Suivi de l'avancement et de la préparation des environnements.
Mineur	Absences non prévue dans l'équipe	Retard dans la conception et/ou l'exécution	Prévoir un plan de backupage

Les risques produit

Ce sont les risques liés à la qualité du produit lui-même.

Niveau	Libellé	Conséquence	Actions préventives / correctives
Critique	Migration des données compliquée entre la solution source et la solution cible	Certaines données ne sont pas reprises ou sont reprises de manière incomplète.	Impliquer l'équipe migration au plus tôt dans le processus de test. Multiplier les tirs de certification statique pour trouver les anomalies au plus tôt.
Critique	Récupération des données dans les SI legacy	Difficulté à comparer les données entre les SI source et les SI cible.	Définir au plus tôt avec les éditeurs départementaux les données attendues dans le SI cible.
Critique	Développer les US de manière trop isolées	Les US ne s'imbriquent pas correctement entre elles. Certains débranchement d'écran ne fonctionnent pas.	Le PO doit toujours avoir une vision globale de la solution cible. Commencer les tests d'intégration usine au plus tôt pour détecter rapidement les problèmes.
Majeur	La solution cible ne communique pas avec les autres applications	Les impacts inter-application sont inexistant.	Procéder rapidement aux tests d'interface technique pour vérifier que les branchements avec les autres applications fonctionnent.
Majeur	Multiplicité des interfaces pour un même flux	Difficulté de développer une solution compatible avec les différentes solution Legacy.	Prévoir des échanges avec les éditeurs départementaux pour comprendre les spécificités de chaque solution.
Majeur	Mauvaise compréhension du besoin	La solution ne correspond pas aux besoin du métier	Intégrer au plus tôt la MOA pour valider que leur besoin a bien été compris.

Périmètre et phases de test

Tests Unitaires

	Description
Objectif	<p>Cette étape de test a pour but de s'assurer unitairement que l'ensemble des critères d'acceptation de chaque US soit validé.</p> <p>Passage des US validées à l'état Done (voir voir DoD)</p>
Périmètre	<p>Le périmètre de test est défini pendant le sprint planning.</p> <p>Tous les critères d'acceptation des US du périmètre de l'itération devront être testées avant le passage de relais à l'équipe de test suivante.</p> <p>L'équipe évaluera la pertinence d'automatiser une partie des tests. Les tests qui auront été automatisés feront partie des livrables dans une logique de non régression.</p>
Entrants	<ul style="list-style-type: none">➤ Le cahier des charges➤ Une expression de besoin➤ Des SFD validées➤ Une stratégie de recette
Prérequis	<ul style="list-style-type: none">➤ Un ou plusieurs environnements de développement bouchonnés.➤ Les composants développés par l'équipe design système ont été mis à disposition de l'équipe MOE
Livrables	<ul style="list-style-type: none">➤ Un plan de test➤ Un bilan d'exécution
Profils contributeurs	<ul style="list-style-type: none">➤ XXXX : PO et référents fonctionnels & techniques➤ Design système : Développeurs et testeurs➤ MOE XXXX : Test manager, experts techniques, développeurs et testeurs
Gestion des anomalies	<p>Les anomalies seront déclarées au fil de l'eau par l'équipe de test dans l'outils de gestion des anomalies de l'intégrateur.</p>
Critères de sortie	<ul style="list-style-type: none">➤ 100% des test déroulés sur les critères d'acceptation des US prioritaires➤ Aucune anomalie bloquante

Tests Intégration usine

	Description
--	-------------

Objectif	L'objectif de cette recette est de vérifier que les différents composants d'une application travaillent correctement ensemble. Elle permet également de vérifier que les performances de la solution sont correctes.
Périmètre	Idem TU
Entrants	Idem TU
Prérequis	➤ Un ou plusieurs environnement d'intégration bouchonnés.
Livrables	<ul style="list-style-type: none"> ➤ Un plan de test ➤ Une démo ➤ Un bilan d'exécution
Profils contributeurs	<ul style="list-style-type: none"> ➤ XXXX : PO et référents fonctionnels & techniques ➤ Design système : Développeurs et testeurs ➤ MOE XXXX : Test manager, experts techniques, développeurs et testeurs
Gestion des anomalies	Les anomalies seront déclarées au fil de l'eau par l'équipe de test dans l'outil de gestion des anomalies de l'intégrateur.
Critères de sortie	<ul style="list-style-type: none"> ➤ 100% des test déroulés sur les critères d'acceptation des US prioritaires ➤ Aucune anomalie bloquante

Recette fonctionnelle

	Description
--	-------------

Objectif	<p>La recette fonctionnelle permet de détecter les défauts de l'application et de remonter des anomalies.</p> <p>L'objectif est de vérifier que les US sont conformes aux attentes du PO.</p>
Périmètre	<p>Il s'agit de réaliser des tests fonctionnels orientés métier. L'objectif est d'avoir un maximum de tests de bout en bout et non de tester une à une les règles de gestion (ce qui est fait en phase de TU).</p> <p>Plusieurs niveaux de test :</p> <ul style="list-style-type: none"> ➤ Tests à la maille itération. Test au niveau des US sur des environnements de recette bouchonnés. <p>Prévoir également une livraison et un déploiement des itérations sur des environnements intégrés XXXX pour les accrochages techniques des interfaces.</p> <ul style="list-style-type: none"> ➤ Tests à la maille incrément. Tests de bout en bout (axés cas d'usages) avec les interfaces sur des environnements d'intégration. <p>Plusieurs spécificités à prendre en compte :</p> <ul style="list-style-type: none"> ➤ Tests de non régression : Les campagnes de tests de non régression auront lieu à chaque fin d'incrément, à partir de l'incrément 2. Elles couvriront les principales fonctionnalités de l'incrément précédent. Dans le cadre de l'incrément 4, l'ensemble du périmètre de TNR sera repris. ➤ Tests de navigateurs : Nous proposons de mettre en place une alternance du navigateur utilisé entre chaque itération. Pour la recette d'incrément, les différents navigateurs devront être testés. ➤ . Tests support mobile : Outils et version à définir
Entrants	<ul style="list-style-type: none"> ➤ Liste des Cas d'usage : 1 semaine avant le début de l'incrément. ➤ SFD validées : 4 semaines avant le début de la recette fonctionnelle <p>Si les entrants devaient évoluer pendant la phase de recette, il faut que les documents soient partagés avec le test manager CGI. Le test manager doit être informé dès qu'une nouvelle version est validée.</p>
Prérequis	<ul style="list-style-type: none"> ➤ Un outil de gestion des tests (Squash TM). ➤ Les conditions de sortie des phases de TU et TI doivent avoir été atteint (voir DoD). ➤ Les objectifs de recette doivent avoir été définis pendant le sprint planning. ➤ Les développements doivent avoir été livrés par la MOE XXXX. ➤ Les développements doivent avoir été installés sur le ou les environnements de recette XXXX à J+2 après la livraison. ➤ Plan de test TU et TI fourni par la MOE. ➤ Démo concluante réalisée par la MOE. ➤ Feu vert obtenu du PO
Livrables	<ul style="list-style-type: none"> ➤ Stratégie d'itération/incrément ➤ Cahiers de tests ➤ Fichier hebdomadaire de suivi d'exécution et des anomalies ➤ Bilan de qualification
Profils contributeurs	<ul style="list-style-type: none"> ➤ XXXX : PO et référents fonctionnels. Equipe technique pour installation des développements. ➤ CGI : Test manager et équipe de test ➤ MOE XXXX : Test manager
Gestion des anomalies	<p>Les anomalies seront déclarées au fil de l'eau par l'équipe de test dans l'outils de gestion des anomalies (JIRA).</p>

	<p>Les anomalies seront priorisées selon leur criticité et leur sévérité. La priorisation pourra être revue lors du point hebdomadaire de suivi des anomalies.</p> <p>Les anomalies bloquantes devront être corrigées très rapidement (durée à définir dans le sprint planning) et livrées dans un patch correctif pour être qualifiées par l'équipe de recette avant la fin de l'itération. Les anomalies majeures devront être corrigées et livrées dans un patch correctif pour être qualifiées avant la fin de l'incrément.</p>
Critères de sortie	<ul style="list-style-type: none"> ➤ 100% des test déroulés sur les US prioritaires ➤ 90% des test déroulés sur l'ensemble du périmètre ➤ Aucune anomalie bloquante résiduelle ➤ Moins de 5 anomalies majeures ou mineures ouvertes

Tests d'interface techniques

	Description
Objectif	<p>L'objectif est de valider les accrochages techniques entre le SI cible et les différentes applications interfacées avec celui-ci.</p> <p>Il s'agit de confirmer avant la phase de test d'interface fonctionnel que les différents outils communiquent bien entre eux et que les formats de fichiers sont conformes à ce qui est attendu.</p>
Périmètre	<ul style="list-style-type: none"> ➤ Des interfaces départementales. Attention, ces interfaces peuvent être différentes selon les éditeurs départementaux. Prévoir de tester les différentes options. ➤ Des interfaces nationales.
Prérequis	<ul style="list-style-type: none"> ➤ Un ou plusieurs environnements de recette ➤ Des fichiers des différents flux départementaux ➤ Des fichiers des différents flux nationaux
Livrables	<ul style="list-style-type: none"> ➤ Plan de test détaillé ➤ Bilan de qualification
Profils contributeurs	<ul style="list-style-type: none"> ➤ XXXX : PO et référent interface du XXXX et des SI legacy ➤ MOE : Experts techniques et experts interfaces ➤ Partenaires : Référents métier et éditeur
Gestion des anomalies	<p>Les anomalies seront déclarées au fil de l'eau par l'équipe de test dans l'outil de gestion des anomalies (JIRA).</p>
Critères de sortie	<ul style="list-style-type: none"> ➤ Toutes les interfaces ont été testées. ➤ Aucune anomalie bloquante ouverte.

Tests d'interface fonctionnels

	Description
Objectif	<p>L'objectif est de valider les branchements entre le SI cible et les différentes applications interfacées avec celui-ci.</p> <p>Il s'agit de confirmer après la phase de test d'interface technique que les différents flux fonctionnent, et que les données échangées sont correctement enregistrées dans l'application cible.</p> <p>Pour les interfaces nationales, prévoir des échanges avec des responsables des applications concernées.</p>
Périmètre	<ul style="list-style-type: none">➤ Des interfaces départementales. Attention, ces interfaces peuvent être différentes selon les éditeurs départementaux. Prévoir de tester les différentes options.➤ Des interfaces nationales.
Prérequis	<ul style="list-style-type: none">➤ Un ou plusieurs environnements de recette➤ Des fichiers des différents flux départementaux➤ Des fichiers des différents flux nationaux
Livrables	<ul style="list-style-type: none">➤ Plan de test détaillé➤ Bilan de qualification
Profils contributeurs	<ul style="list-style-type: none">➤ XXXX : PO et référent interface du XXXX et des SI legacy➤ CGI : Test manager et équipe de test➤ MOE : Experts techniques et experts interfaces➤ Partenaires : Experts des applications nationales et départementales
Gestion des anomalies	<p>Les anomalies seront déclarées au fil de l'eau par l'équipe de test dans l'outil de gestion des anomalies (JIRA).</p>
Critères de sortie	<ul style="list-style-type: none">➤ Toutes les interfaces ont été testées.➤ Aucune anomalie bloquante ouverte.

Tests Usine Migration

	Description
Objectif	L'objectif est de vérifier suite à une migration des données du système Legacy vers la solution cible, on retrouve bien l'exhaustivité des données.
Périmètre	Les tests sont faits par comparaison. On doit pouvoir retrouver le même volume de données entre l'environnement source et l'environnement migré. Après chaque livraison, un tir complet sera lancé. On estime qu'il faudra entre 3 et 5 tirs pour valider la migration.
Prérequis	<ul style="list-style-type: none"> ➤ Des données Legacy (issues de fichiers fournis par l'éditeur responsable de l'export des données d'un SI legacy) ➤ Mise à disposition d'un environnement cible pour pouvoir migrer les données.
Profils contributeurs	<ul style="list-style-type: none"> ➤ XXXX : PO et référent migration ➤ CGI : Test manager ➤ MOE : Experts techniques ➤ Équipe migration : Test manager et Testeurs ➤ Partenaires : Référents métier et éditeur
Gestion des anomalies	Les anomalies seront déclarées au fil de l'eau par l'équipe de test dans l'outil de gestion des anomalies (JIRA).
Critères de sortie	Aucun écart de volume entre les données du Legacy et les données migrées.

Certification statique

	Description
Objectif	L'objectif est de vérifier que suite à une migration des données du système Legacy vers la solution cible, les données sont conformes.
Périmètre	Tests de conformité des données par échantillonnage. Les valeurs des données migrées doivent être intégrées convenablement dans le SI cible. On doit pouvoir les retrouver dans les bons champs et sur les bons écrans conformément aux maquettes définies par les référents XXXX. Après chaque livraison, un tir complet sera lancé. On estime qu'il faudra entre 3 et 5 tirs pour valider la migration.
Prérequis	<ul style="list-style-type: none"> ➤ Des données Legacy (issues de fichiers fournis par l'éditeur responsable de l'export des données d'un SI legacy) ➤ Mise à disposition d'un environnement cible pour pouvoir migrer les données. ➤ La phase de migration technique s'est terminée sans erreur.
Livrables	<ul style="list-style-type: none"> ➤ Stratégie de recette ➤ Plan de test détaillé ➤ Bilan de qualification

Profils contributeurs	<ul style="list-style-type: none"> ➤ XXXX : PO et référent migration ➤ CGI : Test manager et testeurs ➤ MOE : Experts techniques ➤ Équipe migration : Test manager
Gestion des anomalies	Les anomalies seront déclarées au fil de l'eau par l'équipe de test dans l'outil de gestion des anomalies (JIRA).
Critères de sortie	Après 3 tirs, on peut basculer sur des tirs avec Certification statique + certification dynamique.

Certification dynamique

	Description
Objectif	L'objectif est de valider les processus métier et les différentes fonctionnalités en utilisant des données migrées.
Périmètre	<p>Tests comparatifs par échantillonnage. Les fonctionnalités doivent avoir un comportement identique à celui constaté lors de la recette fonctionnelle en réalisant des tests avec des données migrées.</p> <p>Dans un processus de bout en bout, il faut pouvoir continuer une fonctionnalité commencée sur l'application source dans l'application cible</p>
Prérequis	<ul style="list-style-type: none"> ➤ Des données Legacy (issues de fichiers fournis par l'éditeur responsable de l'export des données d'un SI legacy) ➤ Mise à disposition d'un environnement cible avec des données migrées. ➤ Au moins 3 tirs de certification statique ont été déroulés
Livrables	<ul style="list-style-type: none"> ➤ Stratégie de recette ➤ Plan de test détaillé ➤ Bilan de qualification
Profils contributeurs	<ul style="list-style-type: none"> ➤ XXXX : PO et référent migration ➤ CGI : Test manager et testeurs ➤ MOE : Experts techniques ➤ Équipe migration : Test manager
Gestion des anomalies	<p>Les anomalies seront déclarées au fil de l'eau par l'équipe de test dans l'outil de gestion des anomalies (JIRA).</p> <p>Après chaque livraison, un tir complet sera lancé.</p>
Critères de sortie	<ul style="list-style-type: none"> ➤ Toutes les fonctionnalités critiques ont été déroulées ➤ Aucune anomalie bloquante ouverte.

Tests de performance

	Description
Objectif	<p>L'objectif est de s'assurer de la stabilité et de la robustesse de la solution en fonction de la charge.</p> <p>Identifier les potentiels problèmes de performance (goulot d'étranglement, sous dimensionnement de l'infrastructure, ...)</p>
Périmètre	<ul style="list-style-type: none"> ➤ Tests de charge ➤ Tests d'endurance ➤ Test de robustesse ➤ Tests de résistance ➤ Tests de pointe ➤ Tests de volume
Entrants	<ul style="list-style-type: none"> ➤ Des fichiers volumineux pour les tests de robustesse ➤ Points de mesures qui seront à mettre en place pour superviser l'infrastructure ➤ Plan de navigation dans l'application par un utilisateur
Prérequis	<ul style="list-style-type: none"> ➤ Mise à disposition des environnements cibles ➤ Identification et fourniture des outils à utiliser (Neoload, Dynatrace...) ➤ Fourniture des jeux de données (ou méthode de récupération)
Livrables	<ul style="list-style-type: none"> ➤ Document de stratégie de tests techniques ➤ Scripts développés ➤ Bilans et comptes rendus des tirs
Profils contributeurs	<ul style="list-style-type: none"> ➤ XXXX : Référents techniques & fonctionnels ➤ CGI : Test manager, experts Techniques et architectes ➤ ESN dev : Experts techniques
Gestion des anomalies	<p>Les anomalies seront déclarées au fil de l'eau par l'équipe de test dans l'outils de gestion des anomalies (JIRA).</p>
Critères de sortie	<p>100% des points de mesure validés</p>

Tests d'accessibilité

	Description
Objectif	Les tests d'accessibilité et ergonomie permettent de vérifier que l'application est bien accessible selon la réglementation RGAA.
Périmètre	Toutes les US testées lors des phases précédentes
Prérequis	Mise à disposition d'un environnement intégré
Profils contributeurs	<ul style="list-style-type: none"> ➤ XXXX : Référents techniques & fonctionnels ➤ CGI : Test manager ➤ ESN dev : Experts techniques ➤ Partenaire : Expert norme RGAA
Gestion des anomalies	Les anomalies seront déclarées au fil de l'eau par l'équipe de test dans l'outils de gestion des anomalies (JIRA).
Critères de sortie	Tous les développement réalisés répondent à la norme RGAA

Tests de validation métier

	Description
Objectif	L'objectif est de valider que le produit répond aux besoins des utilisateurs finaux
Périmètre	<p>Tous les processus métier dont les US sont au statut Done.</p> <p>La validation métier interviendra au niveau des itérations et des incréments.</p>
Entrants	<ul style="list-style-type: none"> ➤ Bilan de qualification fonctionnelle ➤ Bilan de certification statique et dynamique ➤ Des cahiers de recette
Prérequis	<ul style="list-style-type: none"> ➤ Critère de sortie de recette fonctionnelle atteint ➤ Critère de sortie de certification statique et dynamique atteint ➤ Mise à disposition d'environnements intégrés avec des données migrées représentatives.
Livrables	<ul style="list-style-type: none"> ➤ Bilan de qualification
Parties prenantes	<ul style="list-style-type: none"> ➤ XXXX : Équipe mixte PQHP et CD spécialistes ➤ CGI : Test manager ➤ MOE : Test manager
Gestion des anomalies	Les anomalies seront déclarées au fil de l'eau par l'équipe de test dans l'outils de gestion des anomalies (JIRA).
Critères de sortie	<ul style="list-style-type: none"> ➤ Toutes les fonctionnalités critiques ont été déroulées ➤ Aucune anomalie bloquante ouverte.

Gouvernance des tests

Comitologie

	Objectif	Participants	Fréquence
Comité Anomalies	Suivi des anomalies de recette et priorisation des nouvelles anomalies	Les tests managers et un responsable des développements.	Hebdomadaire 30 min
Comité opérationnel	Suivi de l'avancement des différents travaux Partage des risques et des alertes	Les tests managers, le PO et le scrum master	Hebdomadaire 1 h
Comité de pilotage	Suivi d'avancement Partage des informations financières	Les chefs de projet	Bi mensuel 1h
Bilan d'incrément	Présentation du bilan de recette	Les tests managers, le PO et le scrum master	Fin d'incrément 2h

Comité de suivi

Indicateurs

On distinguera deux types d'indicateurs :

- Indicateurs de suivi permettant de mesurer l'avancement des activités de recette
- Indicateurs de performance permettant de mesurer l'efficacité des activités de recette

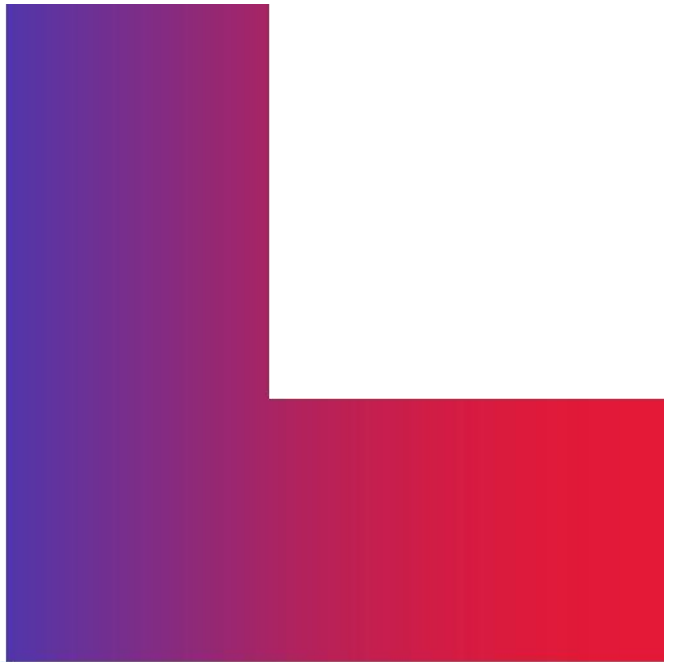
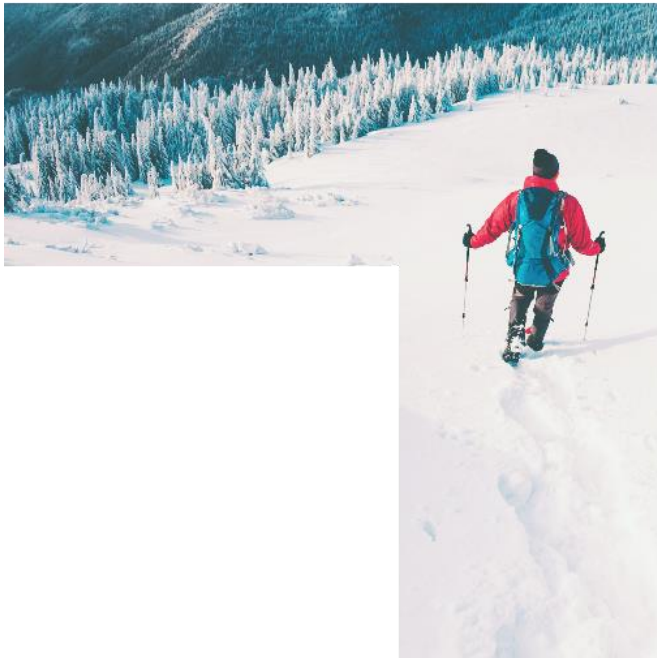
Indicateurs de suivi

Avancement de la qualification : Par itération et par incrément

- Suivi de conception (% avancement, statut des tests conçus)
- Suivi d'exécution
 - % d'avancement
 - Taux de tests non exécutés
 - Taux de succès
 - Taux d'échec
 - Taux de test bloqués
- Suivi de la couverture d'exigence
- Suivi du stock d'anomalies
 - Statut des anomalies
 - Priorité des anomalies
 - Equipe responsable

Indicateurs de performance

- Taux d'anomalies par sévérité
- Taux d'anomalies ouvertes à tort
- Taux d'anomalies par rapport au nombre de tests



US & Épics

Nom de projet : Gestion de Projet

Fichier des Épics et User Stories pour une Application de Gestion de Projet

Épic 1 : Gestion des Projets

****Description**** : Cette épic regroupe toutes les fonctionnalités permettant de créer, modifier, et suivre des projets dans l'application.

US 1.1 : Création d'un projet

- ****En tant que**** chef de projet,
- ****je veux**** créer un nouveau projet,
- ****afin de**** pouvoir organiser les tâches et les ressources associées.
- ****Critères d'acceptation**** :
 - Le chef de projet peut entrer un nom, une description, une date de début, et une date de fin pour le projet.
 - Le système valide que le nom du projet est unique.
 - Une fois créé, le projet est visible dans la liste des projets.
- ****Priorité**** : Haute

US 1.2 : Modification d'un projet

- ****En tant que**** chef de projet,
- ****je veux**** modifier les détails d'un projet existant,
- ****afin de**** mettre à jour les informations si nécessaire.
- ****Critères d'acceptation**** :
 - Le chef de projet peut modifier le nom, la description, les dates, et les membres de l'équipe.
 - Les modifications sont enregistrées et visibles immédiatement.
- ****Priorité**** : Moyenne

Épic 2 : Gestion des Tâches

****Description**** : Cette épic couvre la création, l'assignation, et le suivi des tâches au sein d'un projet.

US 2.1 : Ajout d'une tâche à un projet

- ****En tant que**** chef de projet,
- ****je veux**** ajouter une tâche à un projet,
- ****afin de**** décomposer le travail en étapes gérables.
- ****Critères d'acceptation**** :

- La tâche doit inclure un titre, une description, une priorité (Haute, Moyenne, Basse), et une date d'échéance.
- La tâche est ajoutée à la liste des tâches du projet.
- **Priorité** : Haute

US 2.2 : Assignment d'une tâche à un membre de l'équipe

- **En tant que** chef de projet,
- **je veux** assigner une tâche à un membre de l'équipe,
- **afin de** clarifier les responsabilités.
- **Critères d'acceptation** :
 - Le chef de projet peut sélectionner un membre de l'équipe dans une liste.
 - Le membre assigné reçoit une notification par email.
 - La tâche affiche le nom du membre assigné.
- **Priorité** : Moyenne

US 2.3 : Suivi de l'état d'une tâche

- **En tant que** membre de l'équipe,
- **je veux** mettre à jour l'état de ma tâche (À faire, En cours, Terminé),
- **afin de** informer l'équipe de mon progrès.
- **Critères d'acceptation** :
 - Le membre peut changer l'état de la tâche via un menu déroulant.
 - L'état mis à jour est visible par tous les membres de l'équipe.
 - Une notification est envoyée au chef de projet lorsque l'état passe à "Terminé".
- **Priorité** : Moyenne

Épic 3 : Collaboration et Communication

Description : Cette épic regroupe les fonctionnalités permettant aux membres de l'équipe de collaborer et de communiquer efficacement.

US 3.1 : Commenter une tâche

- **En tant que** membre de l'équipe,
- **je veux** ajouter un commentaire sur une tâche,
- **afin de** partager des informations ou poser des questions.
- **Critères d'acceptation** :

- Le membre peut écrire un commentaire et l'envoyer.
- Le commentaire est visible par tous les membres de l'équipe.
- Une notification est envoyée aux membres assignés à la tâche.
- **Priorité** : Moyenne

US 3.2 : Recevoir des notifications en temps réel

- **En tant que** membre de l'équipe,
- **je veux** recevoir des notifications en temps réel,
- **afin de** rester informé des mises à jour importantes.
- **Critères d'acceptation** :
 - Les notifications sont envoyées pour les assignations de tâches, les commentaires, et les changements d'état.
 - Les notifications apparaissent dans l'application et par email.
- **Priorité** : Moyenne

Épic 4 : Rapports et Suivi

Description : Cette épic inclut les fonctionnalités permettant de générer des rapports et de suivre l'avancement des projets.

US 4.1 : Générer un rapport d'avancement

- **En tant que** chef de projet,
- **je veux** générer un rapport d'avancement pour un projet,
- **afin de** suivre les progrès et les retards.
- **Critères d'acceptation** :
 - Le rapport inclut le pourcentage de tâches terminées, les tâches en retard, et les tâches à venir.
 - Le rapport peut être exporté en PDF.
- **Priorité** : Haute

US 4.2 : Visualiser un tableau de bord

- **En tant que** chef de projet,
- **je veux** visualiser un tableau de bord,
- **afin de** avoir une vue d'ensemble de tous mes projets.
- **Critères d'acceptation** :
 - Le tableau de bord affiche le nombre de projets actifs, les tâches en retard, et les tâches terminées.

- Les données sont mises à jour en temps réel.
- **Priorité** : Haute

Épic 5 : Gestion des Utilisateurs et Rôles

Description : Cette épic couvre la gestion des utilisateurs, leurs rôles, et leurs permissions.

US 5.1 : Ajout d'un membre à l'équipe

- **En tant que** chef de projet,
- **je veux** ajouter un membre à l'équipe d'un projet,
- **afin de** collaborer avec lui.
- **Critères d'acceptation** :
 - Le chef de projet peut inviter un utilisateur par email.
 - L'utilisateur reçoit une invitation et doit l'accepter pour rejoindre l'équipe.
- **Priorité** : Moyenne

US 5.2 : Gestion des rôles (chef de projet vs membre)

- **En tant que** administrateur,
- **je veux** définir les rôles des utilisateurs (chef de projet ou membre),
- **afin de** contrôler leurs permissions.
- **Critères d'acceptation** :
 - Les chefs de projet peuvent créer/modifier des projets et assigner des tâches.
 - Les membres ne peuvent que consulter et mettre à jour leurs tâches.
- **Priorité** : Haute

