



CGDD

Procédure d'Assurance Qualité des projets

INTALIO

intalio

Références

Type du document	Procédure de travail
Centre de Production	Intalio France
Client	Commissariat Général au Développement Durable
Projet	NOVAe
Référence	INTALIO_CGDD_Procédure d’assurance qualité_001

Table des modifications

Historique					
#	Date	Activité	Intervenant	Vérifié par	Validé par
1.0	31/05/2024	Version Initiale	Wady BACCOURI	Marwa JEMLI	Samer NABOULSI

Table des matières

Objectifs du document 4

1. Planification et Analyse 4

2. Types de Tests..... 4

 2.1. Tests de Fumée (Smoke Tests) 5

 2.2. Tests de Système 5

 2.3. Tests de non-Régression 6

 2.4. Tests End-to-End..... 7

3. Collecte des Informations Fonctionnelles 7

 3.1. Analyse des Spécifications 8

 3.2. Documentation Existante 8

 3.3. Réunions avec les Parties Prenantes 8

4. Préparation des Cas de Test 8

 4.1. Mise à Jour du Cahier de Recette 8

5. Exécution des Tests..... 9

 5.1. Environnements Utilisés 9

 5.2. Types de Tests par Environnement 10

6. Gestion des Nouvelles Demandes Clients 10

 6.1. Processus de qualification des nouvelles demandes 11

 6.2. Cycle de vie des Évolutions (client) 11

 6.3. Cycle de vie des anomalies remontées (CLIENT) 12

 6.4. Cycle de vie d’une anomalie remontée par l’équipe QA 12

7. Spécificités d’assurance qualité pour le projet NOVAe..... 13

 7.1. Tests de fumée Automatiques 13

 7.2. Tests de fumée Manuels 14

Conclusion 14

OBJECTIFS DU DOCUMENT

Le but de ce document est de décrire la procédure de gestion de l'assurance qualité (QA) dans notre entreprise. Il détaille les différentes étapes, les typologies de tests, les environnements utilisés et les flux de travail QA pour assurer une transition fluide et efficace jusqu'au déploiement en production.

1. PLANIFICATION ET ANALYSE

La planification et l'analyse sont des étapes cruciales pour définir les objectifs, les critères de réussite, et la portée des tests. Cette phase inclut :

- **Définition des objectifs de test** : Déterminer ce que nous voulons accomplir avec les tests, tels que :
 - Vérification des fonctionnalités
 - Mesure des performances
 - Validation de mise à jour de version
- **Analyse des exigences** : Examiner les spécifications et les exigences du projet (fonctionnelles, non-fonctionnelles) pour identifier les scénarios de test nécessaires.
- **Évaluation de l'impact des évolutions sur les fonctionnalités existantes** : Analyser comment les nouvelles fonctionnalités ou modifications pourraient affecter les fonctionnalités déjà en place pour identifier les risques et ajuster les scénarios de test en conséquence.
- **Ressources et planification du temps** : Allouer les ressources nécessaires (humaines et matérielles) et établir un calendrier de test.

2. TYPES DE TESTS

Nous utilisons plusieurs types de tests pour garantir la qualité du logiciel. Chaque type de test est supervisé par des responsables spécifiques pour assurer une exécution rigoureuse et précise. Les tests sont exécutés dans un ordre chronologique permettant de valider le passage d'une étape à une autre dans le processus de test.

2.1. TESTS DE FUMÉE (SMOKE TESTS)

Responsable : Responsable QA

Étapes :

- Sélectionner les fonctionnalités essentielles
- Rédiger des cas de test pour ces fonctionnalités
- Exécuter les tests rapidement après chaque déploiement
- Vérifier que le logiciel fonctionne de base
- Documenter les résultats et signaler les anomalies

Entrées :

- Déploiement réussi et validé par les équipes DevOps

Sorties :

- Applications opérationnelles (Viewer, Extension, IAM, Storage..)
- Configurations essentielles validée
- Erreurs serveurs éliminés (erreurs 500..)
- Serveurs SMTP opérationnels
- Rapport des tests réalisés

2.2. TESTS DE SYSTÈME

Responsable : Responsable QA

Étapes :

- Identifier le périmètre des fonctionnalités à tester (tickets JIRA, modules des SFD,..)
- Définir les scénarios de test couvrant le périmètre fonctionnel à tester.
- Exécuter les tests pour valider le système dans son ensemble pour le périmètre fonctionnel à tester.

- Assurer que toutes les fonctionnalités fonctionnent de manière cohérente et interagissent correctement entre elles.
- Documenter les résultats et signaler les anomalies.

Entrées :

- Scénarios d'utilisation réelle de l'application

Sorties :

- Périmètre fonctionnel identifié testé et validé
- Système fonctionnel dans des conditions réelles
- Rapport d'assurance qualité système couvrant le périmètre fonctionnel défini

2.3. TESTS DE NON-RÉGRESSION

Les tests de non-régressions permettent de valider que les nouvelles fonctionnalités et correctifs déployés n'ont pas eu d'effet de bord sur d'autres fonctionnalités avec un impact potentiel.

Responsable : Responsable QA

Étapes :

- Identifier les fonctionnalités susceptibles d'être impactées et les scénarios de test associés.
- Vérifier que les modifications récentes n'ont pas introduit de régressions sur les fonctionnalités susceptibles d'être impactées.
- Documenter les résultats et signaler les anomalies.

Entrées :

- Cas de test de non-régression

Sorties :

- Validation d'absence de régressions après les modifications
- Rapport de tests de non-régression

2.4. TESTS END-TO-END

Responsable : Responsable QA

Étapes :

- Définir les flux de travail complets à tester.
- Définir les conditions de test : situations réelles d'utilisation de l'outil.
- Rédiger des scénarios de test couvrant ces flux.
- Exécuter les tests pour valider les flux de travail.
- Vérifier que l'ensemble du système fonctionne correctement dans des situations réelles.

Entrées :

- Cahier de recette globale

Sorties :

- Validation de bout en bout de la version pour passage en production
- Rapport de tests de bout en bout

Chaque responsable doit documenter les résultats de leurs tests respectifs et signaler toute anomalie ou déviation par rapport aux critères d'acceptation.

3. COLLECTE DES INFORMATIONS FONCTIONNELLES

Avant de commencer à élaborer les cas de test et à effectuer l'assurance qualité, il est essentiel de collecter les informations fonctionnelles nécessaires pour comprendre les exigences et les fonctionnalités du système. Ci-dessous les étapes à suivre pour obtenir ces informations :

3.1. ANALYSE DES SPÉCIFICATIONS

Examiner attentivement les spécifications du projet ou tout document de conception fourni pour comprendre les fonctionnalités attendues, les exigences fonctionnelles et les cas d'utilisation du système.

3.2. DOCUMENTATION EXISTANTE

Toute documentation existante sur le système est consultée, tels que des manuels utilisateur, des guides fonctionnels et des diagrammes de flux, pour obtenir des informations détaillées sur les fonctionnalités et les exigences du système.

3.3. RÉUNIONS AVEC LES PARTIES PRENANTES

Des ateliers fonctionnels sont organisés, en phase de clarification fonctionnelle, avec les parties prenantes du projet, y compris les clients, et les chefs de projet, pour clarifier les exigences fonctionnelles et recueillir des informations supplémentaires sur les fonctionnalités attendues.

4. PRÉPARATION DES CAS DE TEST

Les cas de test sont conçus pour couvrir différents aspects de l'application, y compris les fonctionnalités principales, et les scénarios d'utilisation. Chaque cas de test doit être clair, complet et reproductible.

4.1. MISE À JOUR DU CAHIER DE RECETTE

Pour chaque nouvelle spécification ou modification, le cahier de recette doit être mis à jour pour inclure les nouveaux scénarios de test. Cette étape est cruciale pour s'assurer que tous les aspects du produit sont testés de manière adéquate et que les nouvelles fonctionnalités sont couvertes.



5. EXÉCUTION DES TESTS

5.1. ENVIRONNEMENTS UTILISÉS

Nous avons quatre environnements VM distincts pour différents types de tests :

- **Environnement Interne**

Description : L'environnement interne est identique à l'environnement client de recette pour garantir une reproduction précise des erreurs ou des bogues rencontrés lors des tests de recette.

- **Configuration Serveur :** Les serveurs utilisés dans l'environnement interne ont la même configuration matérielle et logicielle que ceux de l'environnement client de recette.
- **Logiciels et Versions :** Les mêmes logiciels, ainsi que les mêmes versions et mises à jour, sont installés dans les deux environnements.
- **Données de Test :** Les données utilisées pour les tests dans l'environnement interne sont une réplique exacte des données utilisées dans l'environnement de recette.



- **Environnement Recette**

Description : L'environnement client de recette est utilisé pour tester par l'équipe QA les fonctionnalités et valider les corrections avant le déploiement final. Il est également accessible au client pour tester et valider les fonctionnalités.

- **Environnement de Formation (CGDD uniquement)**

Description : Environnement spécifique à CGDD, utilisé pour former les utilisateurs administrateurs et les utilisateurs finaux lors des sessions de formation pratiques.

- **Environnement de Production**

Description : Environnement final pour les utilisateurs finaux.

5.2. TYPES DE TESTS PAR ENVIRONNEMENT

Nous avons quatre environnements distincts pour différents types de tests :

Type de Test	Environnement Interne	Environnement Recette	Environnement de Formation	Environnement de Production
Tests de Système	✓	✓	✓	✓
Tests End-to-End		✓		✓
Tests Automatisés	✓	✓	✓	
Tests de non- Régression	✓	✓	✓	
Tests de Fumée	✓	✓	✓	✓

6. GESTION DES NOUVELLES DEMANDES CLIENTS

Cette section détaille le processus de traitement des nouvelles demandes soumises par le client, ainsi que le cycle de vie des demandes.

6.1. PROCESSUS DE QUALIFICATION DES NOUVELLES DEMANDES

1. **Le Client crée une demande via un ticket JIRA** : Ce processus commence lorsque le client signale un problème ou une demande via un ticket dans l'outil de gestion de projet JIRA. 
2. **Le PM / QA analyse la demande** : Le chef de projet (PM) ou l'équipe d'assurance qualité (QA) examine la demande pour comprendre sa nature et son impact.
3. **Anomalie ou évolution ?** : À ce stade, une décision est prise pour déterminer si la demande est liée à un défaut (anomalie) ou à une évolution (nouvelle fonctionnalité ou amélioration).
4. **Si c'est une anomalie** : Cycle de vie des anomalies
5. **Si c'est une évolution** : Cycle de vie des évolutions

6.2. CYCLE DE VIE DES ÉVOLUTIONS (CLIENT)

1. **Nouvelle demande d'évolution du client** : Une nouvelle demande est initiée par le client, nécessitant des changements ou des ajouts au projet.
2. **Analyse des exigences par le PM** : Le Project Manager (PM) analyse les exigences pour s'assurer que la demande est claire et validée par le client.
3. **Étude de faisabilité par le PM** : Le PM évalue la faisabilité de la demande en déterminant si l'équipe peut la mettre en œuvre en tenant compte des contraintes du projet (temps, budget, ressources, périmètre du projet). Les critères de décisions se basent sur :
 - La nature de la demande : produit ou personnalisation
 - La faisabilité fonctionnelle et technique
 - La complexité
4. **Approbation des exigences** : Les exigences sont approuvées par les parties prenantes après vérification de leur clarté et de leur faisabilité.

5. **Planification** : Une phase de planification est entamée pour organiser les tâches nécessaires à la réalisation de la demande, y compris la définition des délais et l'allocation des ressources.
6. **Développement** : L'équipe de développement travaille sur la mise en œuvre des exigences approuvées en suivant le plan établi.
7. **Tests** : Une série de tests est réalisée pour s'assurer que les développements répondent aux exigences et fonctionnent correctement sans bugs.
8. **Retour du client** Le client fournit un retour sur le travail effectué, permettant de valider la satisfaction ou d'identifier des ajustements nécessaires.

6.3. CYCLE DE VIE DES ANOMALIES REMONTÉES (CLIENT)

1. **Bug découvert par le client** : Ce processus commence lorsque le client signale un bug dans le logiciel.
2. **Analyse du bug par le QA** : L'équipe d'assurance qualité (QA) analyse le bug pour comprendre sa nature et sa gravité.
3. **Planification du bug par le PM** : Le chef de projet (PM) planifie la résolution du bug en tenant compte de sa priorité.
4. **Bug assigné à l'équipe de développement** : L'équipe de développement reçoit l'affectation du bug et commence à travailler sur sa résolution.
5. **Bug résolu par l'équipe de développement** : L'équipe de développement corrige le bug.
6. **Bug retesté par le QA** : Le QA reteste le logiciel pour vérifier que le bug a été correctement résolu. 
7. **Bug vérifié par le client** : Le client vérifie que le bug a bien été corrigé.
8. **Bug fermé par le PM** : Le PM clôture le bug une fois qu'il est confirmé qu'il a été résolu.

6.4. CYCLE DE VIE D'UNE ANOMALIE REMONTÉE PAR L'ÉQUIPE QA

1. **Anomalie découverte par l'équipe QA** : Ce processus commence lorsque l'équipe QA signale un bug dans le logiciel.

2. **Analyse du bug par le QA** : L'équipe QA analyse le bug pour comprendre sa nature et sa gravité.
3. **Planification du bug par le PM** : Le chef de projet (PM) planifie la résolution du bug en tenant compte de sa priorité (Low, Medium, Show stopper).
4. **Bug assigné à l'équipe de développement** : L'équipe de développement reçoit l'affectation du bug et commence à travailler sur sa résolution.
5. **Bug résolu par l'équipe de développement** : L'équipe de développement corrige le bug.
6. **Bug retesté par le QA** : Le QA reteste le logiciel pour vérifier que le bug a été correctement résolu.
7. **Bug vérifié par le client** : Le client vérifie que le bug a bien été corrigé.
8. **Bug fermé par le PM** : Le PM clôture le bug une fois qu'il est confirmé qu'il a été résolu.



7. SPÉCIFICITÉS D'ASSURANCE QUALITÉ POUR LE PROJET NOVAE

L'ensemble des tests standard INTALIO sont appliqués pour l'assurance qualité sur le projet NOVAe. Les cahiers de recettes ainsi que les scénarios de tests sont adaptés aux spécificités du projet. La seule particularité NOVAe en termes de types de test réside dans l'exploitation de tests automatiques durant la phase des Smoke Tests, en plus des tests manuels standards. Ci-dessous les détails relatifs aux tests de fumée « Smoke tests » de NOVAe.

Dans le contexte NOVAe, les tests de fumée sont réalisés en combinant des tests automatiques et des tests manuels.

7.1. TESTS DE FUMÉE AUTOMATIQUES

Les tests automatiques de NOVAe reposent sur un ensemble de fonctionnalités jugées critiques pour le projet. Ci-dessous la liste des scénarios testés automatiquement.

Portail pétitionnaire

- Connexion du pétitionnaire

- Création de demande et remplissage du formulaire pour l'ensemble des WF et pour le critère région :
 - Une demande pour une seule région
 - Une demande pour plusieurs régions

CTS

- Connexion sur CTS en tant qu'administrateur
- Vérification du verrouillage d'une demande
- Vérification de la recevabilité
- Vérification de la complétude
- Vérification de la section Instruction (Validation 0, Niveau 1 et Niveau 2)
- Vérification de la publication

7.2. TESTS DE FUMÉE MANUELS

Portail pétitionnaire

- Création du compte pétitionnaire
- Déclenchement d'un recours (wkf 1, 2 et 3)

CTS

- Vérification de la section Instruction (Validation 0 avec transfert automatique, Niveau 1 avec transfert automatique et Niveau 2)
- Vérification de la section Instruction (RG) (Validation 0 auto, Niveau 1 auto et Niveau 2)
- Vérification de la section Déclencher recoure RC (admin)

CONCLUSION

En résumé, la procédure d'assurance qualité mise en place, d'une façon générique pour l'ensemble des projet INTALIO et adaptée au projet NOVAe, permet de garantir une transition fluide et efficace jusqu'au déploiement en production. En structurant les différentes étapes de tests, de la planification initiale à l'exécution des tests spécifiques et à la gestion des anomalies,

nous assurons la robustesse et la fiabilité du logiciel développé. L'utilisation d'une combinaison de tests manuels et automatiques, adaptée aux spécificités du projet, renforce notre capacité à identifier et à corriger les éventuelles défaillances de manière proactive. La rigueur et la précision de cette démarche assurent non seulement la conformité aux exigences du CGDD mais également la satisfaction des utilisateurs finaux.