Qualité Logiciels Racine Carrées

Version <1.0>

2023-03-02

Loïc DUMAY

*[Note: Le texte inséré entre crochets et affiché en italique bleu sert à fournir un guide à l’auteur et doit être effacé avant la publication du document. Le paragraphe suivant ce texte est en style Normal.]*

[Pour personnaliser les champs automatiques avec Microsoft Word, qui affichent un arrière plan lorsqu’on les sélectionne activer Fichier/Propriétés et modifier les champs Sujet, Auteur, Compagnie et catégorie de l’onglet Résumé, et le champ Version de l’onglet Personnalisation Ces champs peuvent être mis à jour dans le document en les sélectionnant tous ou individuellement et en activant la touche F9. Les champs des en-têtes et des pieds de page doivent être mise à jour séparément.]

Historique des modifications du document

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Auteur** |
| <aaaa-mm-jj> | <x.x> | <détails> | <nom> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table des matières

1. Introduction 4

1.1 Objectif du document 4

1.2 Portée du document 4

1.3 Historique 4

1.4 Définitions, acronymes et abréviations 4

1.5 Identification des artéfacts 4

1.6 Vue d’ensemble 5

2. Exigences à tester 6

3. Stratégie de tests 6

3.1 Types de tests 6

3.1.1 Tests fonctionnels 6

Tableau récapitulatif des tests à faire pour chaque type !!! 7

3.1.2 Tests d’interface utilisateur 7

3.1.3 Tests de données et d’intégrité de base de données 7

3.1.4 Profilage de performance 8

3.1.5 Tests de configuration 9

3.1.6 Tests d’installation 10

3.2 Outils 11

4. Ressources 11

4.1 Travailleurs 11

4.2 Système 13

5. Jalons du projet 13

6. Biens livrables 13

6.1 Modèle de test 13

6.2 Journaux de test 14

6.3 Rapports d’anomalies 14

7. Annexe A: Tâches du projet 15

Plan de tests

# Introduction

[L’introduction donne une vue d’ensemble de tout le document. On y présente toute information que le lecteur a besoin pour comprendre le document. Elle comprend l’objectif du document, sa portée, les définitions, acronymes et abréviations, les références et une vue d’ensemble du document.]

## Objectif du document

*[Préciser les objectifs de ce document comme, par exemple :*

* Identifier les informations existantes du projet et les composants qui doivent être testés.
* Énumérer les exigences d’évaluation à haut niveau
* Recommander et décrire les stratégies de test à employer
* Identifier les ressources nécessaires et fournir un estimé de l’effort de test
* Identifier les biens livrables pour les tests.]

## Portée du document

[Une brève description de la portée de ce document, notamment les étapes d’évaluation, comme par exemple, les tests unitaires, les tests d’intégration, les tests système et les types de test qui sont traités par ce plan comme les tests fonctionnels et les tests de performance.

Énumérer brièvement les caractéristiques ciblées pour test et les fonctions qui ne seront pas testées.

Énumérer toutes les hypothèses faites durant le développement de ce document qui ont un impact sur la conception, le développement ou l’implémentation des tests.

Énumérer les risques et contingences qui peuvent affecter la conception, le développement ou l’implémentation des tests.

Énumérer toutes les contraintes qui peuvent affecter la conception, le développement ou l’implémentation des tests.]

## Historique

[Décrire brièvement, 3 ou 5 paragraphes, des cibles des tests, composants, application, système, etc., et leurs buts. Inclure l’information sur les caractéristiques principales, l’architecture et un bref historique du projet.]

## Définitions, acronymes et abréviations

[Énumérer les définitions de tous les termes, acronymes et abréviations nécessaires à la compréhension du document d’architecture logicielle. Cette information peut renvoyer à l’artéfact Glossaire du projet.]

## Identification des artéfacts

Le tableau suivant identifie les artéfacts et leur disponibilité pour évaluation :

[NOTE: Ajouter ou supprimer les items appropriés.]

| **Document  (version / date)** | **Disponibilité** | **Revu** | **Ressource** | **Notes** |
| --- | --- | --- | --- | --- |
| *Spécification d’exigences logicielles* | 🞏 Non | 🞏 Non |  |  |
| *Spécifications des cas d’utilisation* | 🞏 Non | 🞏 Non |  |  |
| *Réalisation des cas d'utilisation* | 🞏 Non | 🞏 Non |  |  |
| *Prototype d’interface utilisateur* | 🞏 Non | 🞏 Non |  |  |

## Vue d’ensemble

*[Cette section décrit le contenu du reste du document et explique comment le document est organisé.]*

# Exigences à tester

La liste suivante identifie les items, cas d’utilisation, exigences fonctionnelles et exigences non-fonctionnelles, qui ont été désignés comme cibles de test. Cette liste représente ce qui sera testé.

[Énumérer les principales exigences à tester]

# Stratégie de tests

[La stratégie de tests présente l’approche recommandée d’évaluation des cibles des tests. La précédente section, Exigences à tester, décrit de qui sera testé et cette section comment cela sera testé.

Fournir, pour chaque type de test, une description du test et comment il sera implémenté et exécuté.

Lorsque les tests ne sont pas implémentés et exécutés, expliquer en une phrase pourquoi.

La stratégie de tests consiste d’abord à identifier les techniques de tests et à identifier les critères de complétion des tests.

Les tests doivent être exécutés dans des environnements sécurisés avec des données connues et contrôlées.]

## Types de tests

### Tests fonctionnels

[Les tests fonctionnels portent sur les exigences fonctionnelles qui peuvent être retracées dans les cas d’utilisation ou les fonctions d’affaire et dans les règles d’affaire. Le but de ces tests est de vérifier la validité des données, de leur traitement et de leur récupération ainsi que des règles d’affaire. Ce type d’évaluation repose sur la technique de la boîte noire, c’est-à-dire, une vérification de l’application et de ses processus internes en l’utilisant avec l’interface utilisateur et en en analysant les résultats. Voici un exemple de stratégie proposée :]

|  |  |
| --- | --- |
| **Stratégie pour les tests fonctionnels** | |
| Objectif de test: | Vérifier que la fonctionnalité rencontre la bonne cible de test, incluant la navigation, la saisie de données, le traitement et la récupération. |
| Technique: | Exécuter les cas d’utilisation, chaque chemin du cas ou fonction en utilisant des données valides et non valides tout en vérifiant:   * Les résultats attendus avec des données valides. * Les messages d’erreur et d’avertissement lorsque des données non valides sont utilisées. * Que chaque règle d’affaire est appliquée correctement. |
| Critère de complétion: | * Tous les tests prévus ont été exécutés. * Toutes le anomalies identifiées ont été enregistrées. |
| Considérations: particulières | [Décrire les items ou soulever les questions internes ou externes qui ont un impact sur l’implémentation et l’exécution d’un test fonctionnel] |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Récapitulatif des tests**[Trigramme du type FON pour fonctionnel par exemple] | | | | |
| **Numéro** | **Nom du test** | **Couverture Tests** | **Couverture du code** | **Test prévu** |
| 1 | Nom/description de la fonctionnalité à tester (Exemple : login du client à l’application, connexion à la BDD, importation des données….) | X% des fonctionnalités | X% du code | Oui NON  Commentaires |
| 2 |  |  |  |  |
| 3 |  |  |  |  |

### Tableau récapitulatif des tests à faire pour chaque type !!!

### Tests d’interface utilisateur

[Les tests d’interfaces utilisateur évaluent l’interaction de l’utilisateur avec le logiciel. Le but de ces tests est de vérifier que l’interface utilisateur donne l’accès approprié aux fonctions des cibles de test. De plus, ces tests permettent de vérifier que les objets de l’interface utilisateur se comporte tel qu’attendu et qu’ils sont conformes aux normes du client ou de l’industrie.]

|  |  |
| --- | --- |
| **Stratégie pour les tests d’interface utilisateur** | |
| Objectif de test: | Évaluer:   * La navigation des cibles de test représentent adéquatement les fonctions d’affaire et les exigences, incluant d’écran à écran, de champ à champ, et l’utilisation des modes de déplacement tells les clés de tabulation, le mouvement de la souris et les clés d’accès rapide. * Les composants de l’écran et les caractéristiques tels les menus, la taille, la position, la cible de saisie et l’état sont conformes aux normes. |
| Technique: | Produire des tests pour chaque écran afin de vérifier la navigation et les états de chacun des écrans et composants de l’application. |
| Critère de complétion: | La vérification de chaque écran est concluante en se conformant à la référence ou à une norme acceptable. |
| Considérations: particulières | Il est possible que toutes les propriétés personnalisables ou des composants d’un tiers ne soient testées. |

### Tests de données et d’intégrité de base de données

[Les bases de données et les processus de base de données doivent être testés comme sous-systèmes à l’intérieur du projet. Ces sous-systèmes doivent être testés sans utiliser l’interface utilisateur comme interface des données. L’identification des outils et techniques pour effectuer les tests repose sur la nature du système de gestion de base de données (SGBD).]

|  |  |
| --- | --- |
| **Stratégie pour les tests de données et d’intégrité de base de données** | |
| Objectif de test: | Vérifier les méthodes d’accès à la base de données ainsi que les processus fonctionnent correctement et sans corruption de données. |
| Technique: | Appeler chaque méthode et processus d’accès à la base de données en alimentant chacun de données valides, de données invalides et de requêtes de données.  Inspecter la base de données pour vérifier que les données ont été chargées comme prévu, que toutes les opérations de la base de données s’exécutent normalement et réviser les données produites afin de s’assurer que les bonnes données ont été extraites et pour les bonnes raisons. |
| Critère de complétion: | Tous les processus et méthodes d’accès fonctionnent tels qu’ils ont été conçus sans corruption de données. |
| Considérations: particulières | Les tests peuvent nécessiter un environnement de développement du SGBD ou des pilotes pour charger ou modifier directement des données dans la base de données.  Les processus doivent être actives manuellement.  Des bases de données petites ou de taille minimale devraient être utilisées pour accroître la visibilité d’événements non acceptables. |

### Profilage de performance

[Le profilage de la performance est un test de performance où les temps de réponses, les taux de transaction et les autres exigences temporelles sont mesurées et évaluées. Le but du profilage de performance est de vérifier si les exigences de performances ont été atteintes. Il est implémenté et exécuté profiler et ajuster la performance des cibles de test en fonction de conditions comme la charge de travail et les configurations matérielles.

NOTE : Les transactions sont, ci-dessous, des transaction d’affaire logiques. « Ces transactions sont définies comme des cas d’utilisation qu’un acteur du système est présumé exécuter en utilisant une cible de test comme, par exemple, ajouter ou modifier un contrat.

|  |  |
| --- | --- |
| **Stratégie pour le profilage de performance** | |
| Objectif de test: | Vérifier la performance d’exécution de transactions désignées ou de fonctions d’affaires dans les conditions suivantes :  - Charge de travail normale  - Charge de travail dans le pire des cas |
| Technique: | Les procédures de test utilisées sont développées pour une évaluation cyclique de fonctions.  Modifier les fichiers de données pour accroître le nombre de transaction ou les scripts pour accroître le nombre d’itérations des transactions.  Les scripts doivent être exécutés sur une machine, le meilleur des cas pour référencer un seul utilisateur et une seule transaction et être répété avec plusieurs clients, virtuel ou réel selon les considérations particulières ci-dessous. |
| Critère de complétion: | Un seul utilisateur et une seule transaction : Complétion réussie des scripts de test sans problèmes et dans le temps requis prévu par transaction.  Plusieurs transactions et plusieurs utilisateurs: Complétion réussie des scripts de test sans problèmes et dans le temps requis prévu par transaction. |
| Considérations: particulières | Une évaluation de performance valable suppose une charge de travail en arrière pan sur le serveur :  Pour y arriver, différentes méthodes peuvent être utilisées :   * Émuler des transactions directement sur le serveur à partir de requêtes SQL. * Créer une charge utilisateur virtuelle, pour simuler plusieurs centaines de clients. Des outils d’émulation de terminal à distance peuvent être utilisés. La technique peut aussi être utilisés pour tester le trafic réseau. * Utiliser plusieurs clients physiques, par l’exécution de scripts de test pour charger le système.   Les tests de performance doivent être exécutés sur une machine dédiée ou à un moment dédié, afin de s’assurer de mesures contrôlées exactes.  Les bases de données utilisées pour les tests de performance doivent avoir une taille courante ou avoir une taille proportionnellement comparable. |

### Tests de configuration

[Les tests de configuration valident les opérations pour des cibles de test pour différentes configurations logicielles et matérielles. Les spécifications matérielles des postes de travail, des réseaux et des serveurs de base de données, en environnement de production, varient d’un client à l’autre. Les postes de travail peuvent avoir différents logiciels en opération selon différentes combinaisons et utilisant différentes ressources.]

|  |  |
| --- | --- |
| **Stratégie pour les tests de configuration** | |
| Objectif de test: | Vérifier que la cible de test fonctionne correctement avec les configurations logicielle et matérielle définies. |
| Technique: | Utiliser des scripts de tests fonctionnels  Ouvrir et fermer des logiciels qui ne sont pas des logiciels liés à la cible de tests, comme Microsoft Excel ou Microsoft Word, avant de démarrer le test.  Exécuter des transactions afin de simuler des acteurs qui interagissent avec la cible de test et avec un logiciel qui n’est pas associé à la cible de tests.  Répéter le processus en réduisant la mémoire conventionnelle disponible sur le poste de travail. |
| Critère de complétion: | Our chaque combinaison de cible de test et de logiciel non lié à la cible de test, toutes les transactions doivent être complétées sans défaillance. |
| Considérations: particulières | Déterminer quels sont les logiciels non liés à la cible de test disponible et accessible sur le bureau.  Déterminer quelles sont les applications habituellement utilisées.  Déterminer les données traitées par ces applications comme, par exemple, un document Word de 100 pages.  Tout le système, le réseau, les serveurs réseau, les bases de données, etc. devraient être aussi documentées lors de l’exécution de ces tests. |

### Tests d’installation

[Les tests d’installation poursuivent deux objectifs. D’abord, s’assurer que le logiciel peut être installé dans différentes conditions, tant pour les nouvelles installations, les mises à jour que pour compléter une installation personnalisée dans des conditions normales et des conditions anormales. Les conditions anormales comprennent un espace disque insuffisant, des autorisations insuffisantes pour créer des répertoires, etc. L’autre objectif est de vérifier que, une fois installé, le logiciel fonctionne correctement. Cela signifie qu’il faut exécuter un certain nombre de tests fonctionnels.]

|  |  |
| --- | --- |
| **Stratégie pour les tests d’installation** | |
| Objectif de test: | Vérifier que la cible de test est installée correctement pour chaque configuration matérielle requise, avec les conditions suivantes :   * Nouvelle installation, sur une nouvelle machine où l’application n’a jamais été installée auparavant. * Mise à niveau d’une machine avec une application de la même version. * Mise à jour d’une machine avec une application d’une plus vieille version. |
| Technique: | Valider les conditions de la machine cible manuellement ou avec des scripts automatiques : nouvelle installation, réinstallation de la même version ou réinstallation d’une vieille version.  Exécuter une installation.  Utiliser un sous-ensemble de scripts de tests fonctionnels et exécuter les transactions. |
| Critère de complétion: | Les transactions sont exécutées avec succès sans défaillance. |
| Considérations: particulières | Déterminer quelles sont les transactions qui garantiront que l’application sera installée avec succès sans que des composants majeurs ne soient oubliés. |

## Outils

Les outils suivants seront employés pour ce projet:

[NOTE: Supprimer ou ajouter les items selon le cas.]

|  |  |
| --- | --- |
|  | **Outil** |
| Gestion des tests |  |
| Suivi des anomalies |  |
| Tests fonctionnels |  |
| Tests de performance |  |
| Gestion de projet |  |
|  |  |
|  |  |

# Ressources

[Énumérer les ressources nécessaires à l’effort de test, leurs principales responsabilités, leur connaissance et leurs qualifications.]

## Travailleurs

Ce tableau énumère les ressources humaines nécessaires à la réalisation de ce plan.

[NOTE: Supprimer ou ajouter les items selon le cas.]

|  |  |  |
| --- | --- | --- |
| **Ressources humaines** | | |
| **Travailleur** | **Nombre minimum (plein temps)** | **Responsabilités - commentaires** |
| Responsable des tests |  | Fournir un encadrement de gestion  Responsabilités:   * Fournir une direction technique * Recruter les ressources appropriées * Gérer les rapports |
| Concepteur de test |  | Identifier, prioriser et implémenter les cas de test  Responsabilités:   * Générer le plan de tests * Générer le modèle de tests * Évaluer l’efficacité de l’effort de test |
| Testeur |  | Exécuter les tests  Responsabilités:   * Exécuter les tests * Journaliser les résultats * Reprendre sur les erreurs * Documenter les anomalies et les demandes de changement. |
| Administrateur de tests système |  | Mettre en place et maintenir un environnement de tests  Responsabilités:   * Administrer le système de gestion de tests * Installer et gérer l’accès des travailleurs au système de tests. |
| Administrateur de tests de base de données |  | Mettre en place et maintenir les données de tests.  Responsabilités:   * Administrer les bases de données de tests |
| Concepteur |  | Identifier et définir les opérations, attributs et associations des classes test.  Responsabilités:   * Identifier et définir les classes de test. * Identifier et définir les paquetages de test |
| Implémenteur |  | Implémenter et exécuter les tests unitaires de classes de test et de paquetages de test.  Responsabilités:   * Créer des classes de test et les paquetages de test du modèle de test. |

## Système

Le tableau suivant identifie les ressources du système nécessaire à l’exécution du plan de tests.

Les éléments du système de tests ne sont pas tous connus à ce jour. Il est recommandé que le système simule un environnement de production en réduisant les accès et la taille des bases de données lorsque nécessaire.

[NOTE: Supprimer ou ajouter les items selon le cas.]

|  |  |
| --- | --- |
| Ressources du système | |
| Ressource | Nom – Type |
| Serveur de base de données |  |
| * Réseau – sous réseau | TBD |
| * Nom du serveur | TBD |
| * Nom de la base de données | TBD |
| Poste de travail |  |
| * Exigences de configuration spéciales | TBD |
| Référentiel de tests |  |
| * Réseau – sous réseau | TBD |
| * Nom du serveur | TBD |
| Poste de développement de tests | TBD |

# Jalons du projet

[L’exécution des tests doit inclure les activités de test pour chaque effort de test identifié dans les sections précédentes. Des jalons distincts devraient être identifiés pour transmettre le statut du projet de tests et ses réalisations.]

|  |  |  |  |
| --- | --- | --- | --- |
| **Activité** | **Effort** | **Début** | **Fin** |
| Planification des tests |  | <aaaa-mm-jj> | <aaaa-mm-jj> |
| Conception des tests |  | <aaaa-mm-jj> | <aaaa-mm-jj> |
| Implémentation des tests |  | <aaaa-mm-jj> | <aaaa-mm-jj> |
| Exécution des tests |  | <aaaa-mm-jj> | <aaaa-mm-jj> |
| Évaluation des tests |  | <aaaa-mm-jj> | <aaaa-mm-jj> |

# Biens livrables

[Énumérer les artéfacts, documents, outils et rapports qui seront créés, leur auteur, leur destinataire et la date de leur livraison]

## Modèle de test

[Identifier les rapports qui seront créés et extraits du modèle de tests.Faire un document listant toutes les fiches de tests avec la synthèse de chaque résultat – rapport de tests.doc]

## Journaux de test

[Décrire la méthode et les outils utilisés pour enregistrer et faire rapport des résultats de test et du statut des tests cf documents fiches de tests.]

## Rapports d’anomalies

[Identifier la méthode et les outils utilisés pour enregistrer, tracer et faire rapport des incidents de test et de leur statut…..des logiciels existents pour le suivi : TestLink, TestRail, TestPas, TestTrackTCM, HP QUality Center...]

# Annexe A: Tâches du projet

Voici la liste des tâches liées à l’exécution des tests :

* Planifier les tests
  + Identifier les exigences de test
  + Évaluer les risques
  + Identifier les ressources nécessaires aux tests
  + Développer une stratégie de test
  + Créer le calendrier de tests
  + Générer le plan de test
* Concevoir les tests
  + Analyser la charge de travail
  + Identifier et décrire les cas de test
  + Identifier et structurer les procédures de test
  + Réviser la couverture des tests
* Implémenter les tests
  + Enregistrer ou programmer les scripts de test
  + Identifier les fonctionnalités propres au test dans les modèles de conception et d’implémentation.
  + Établir des ensembles de données externes
* Exécuter les tests
  + Exécuter les procédures de test
  + Évaluer l’exécution des tests
  + Récupérer les tests suspendus
  + Vérifier les résultats
  + Examiner les résultats non prévus
  + Enregistrer les anomalies
* Évaluer les tests
  + Évaluer la couverture des cas de tests
  + Évaluer la couverture du code
  + Analyser les anomalies
  + Déterminer si les critères de complétion et de succès des tests ont été atteints.