

MÉMOIRE

Présenté à l'école nationale des sciences appliquées pour l'obtention du diplôme :

« Ingénieur d'Etat en Génie informatique »

Titre

Automatisation des tests fonctionnels

Réalisé Par :

MAYZRANE HASSAN

Effectué à :

NEOXIA MAROC - CASABLANCA

Sous la direction de :

Pr. JRAIFI Abdelilah, Encadrant ENSAS : Enseignant chercheur.

Mme. BENNANI Meryem, chef de projet et manager des tests chez Neoxia Maroc.

Soutenu le XX/06/2019 devant le jury :

M. JRAIFI Abdelilah, Encadrant ENSAS : Enseignant chercheur.

M. ATLAS Abdelghafour, Enseignant chercheur à L'ENSA de Safi

M. BOUARIFI Walid, Enseignant chercheur à L'ENSA de Safi

M. OUJAOURA Mustapha, Enseignant chercheur à L'ENSA de Safi

Année universitaire : 2018/2019

DEDICACE

C'est avec profonde gratitude et sincères mots, que je dédie ce travail à mes chers parents, qui ont tout fait pour m'apporter soutien, ont fait de grands sacrifices pour ma réussite et m'ont éclairé la voie avec leurs précieux conseils.

J'espère qu'un jour, je pourrai leurs rendre un peu de ce qu'ils ont fait pour moi.

Je dédie aussi ce travail à mon frère et à tous mes amis sans exception.

REMERCIEMENT

Au terme de ce travail, je veux exprimer mes gratitudes et mes remerciements pour toutes les personnes qui ont contribué à sa réalisation.

Je tiens tout d'abord à remercier **Pr. JRAIFI Abdelilah**, mon encadreur de stage, **Mme. BENNANI Meryem**, pour son aide, ses conseils, son encouragement, sa disponibilité et pour le temps qu'il alloué pour corriger mes erreurs afin d'avoir une meilleure qualité de travail.

Mes profonds remerciements pour les membres du jury qui ont accepté d'évaluer ce travail.

Résumé

Le présent document représente un bilan du travail accompli dans le cadre de stage du 3^{ème} année du cycle d'ingénieur d'état en génie informatique à l'Ecole Nationale des Sciences Appliquées de Safi, au sein de **Neoxia Maroc**.

Dans le cadre de mon stage, je me suis intéressé à réaliser des projets qui servent à automatiser le lancement des tests fonctionnels, parmi les objectifs attendus :

- ✧ Éviter la redondance ;
- ✧ Le pert de temps;
- ✧ Optimiser les coûts associés au test manuel.

Un bref aperçu sur le déroulement du projet de l'automatisation des tests fonctionnels :

Il consiste à concevoir un environnement pour lancer des tests automatisés à une simple clique et avoir comme résultat un rapport sur l'état d'exécution.

Afin de répondre mieux à ces exigences, nous avons choisi de suivre la méthodologie **Scrum** comme procédure de gestion de projet, Après analyse du cahier de charges, nous avons identifié les fonctionnalités requises et commençons à consommer des différents outils de l'automatisation des tests fonctionnels (**Cypress, Cucumber, Behat, Squash-TA et Selenium**).

SOMMAIRE

<i>Introduction générale</i>	10
<i>Chapitre I Contexte général du projet</i>	11
Introduction.....	12
1. Présentation de l'organisme d'accueil	12
1.1 Présentation	12
1.2 Lignes de business de Neoxia	12
1.3 Neoxia en chiffre	13
1.3.1 Effectif.....	13
1.3.2 Chiffre d'affaire	14
1.4 Offre de service	15
2. Description détaillée de la problématique	17
3. La solution envisagée.....	18
<i>Chapitre II</i>	20
<i>Analyse des besoins</i>	20
1. Cahier des Charges	21
1.1 Objectifs.....	21
1.2 Cibles	21
2. Planification	21
2.1 Méthodologie de travail.....	21
Méthodes agiles.....	22
<i>Chapitre III</i>	24
<i>Conception du projet</i>	24
1. Analyse	25
1.1 Problématique	25
1.2 Test fonctionnel	25
2.1.1 Comment préparer efficacement un test fonctionnel?	26
1.3 Mise en œuvre de tests fonctionnels automatisés.....	26
1.3.1 Intérêt des tests automatisés.....	26
1.3.2 Qu'est-ce qu'un test automatisé ?.....	26
1.3.3 Outils de test automatisé	27

2. Elaboration des diagrammes UML.....	27
1. Le langage UML	27
1.1 Identification des acteurs	27
1.2 Diagrammes des cas d'utilisation.....	28
1.3 Description textuelle	30
1.4 Diagramme de séquence	31
1.5 Diagramme de classe.....	32
<i>Chapitre IV.....</i>	<i>33</i>
<i>Réalisation du projet.....</i>	<i>33</i>
<i>Conclusion et Perspectives</i>	<i>34</i>
<i>Références</i>	<i>35</i>
<i>Annexes</i>	<i>36</i>

LISTE DES FIGURES

Figure 1 : Lignes de business de Neoxia	11
Figure 2 : Offre de prestations Neoxia	12
Figure 3 : Répartition de l'effectif par type de profil.	12
Figure 4 : Chiffre d'affaires Neoxia France en million d'Euros.	13
Figure 5 : Chiffre d'affaire de Neoxia Maroc en Million de Dirham.	13
Figure 6 : Chiffre d'affaire par secteur d'activité.	14
Figure 7 : Différentes typologies de mission de Neoxia.	16
Figure 8 : Vue synthétique du processus Scrum.	23
Figure 9 : Les Acteurs	23
Figure 10 : Cas d'utilisation concernant l'ingénieur de test.	24
Figure 11 : diagramme de séquence, la rédaction d'un cas de test.	26
Figure 12 : diagramme de classe.	33

LISTE DES ABREVIATIONS

Squash-TA	S quash T est A utomation
Squash-TM	S quash T eam M anagement
SQL	S tandard Q uering L anguage
XML	E xtensible M arkup L anguage
HTML	H yper T ext M arkup L anguage
JDBC	J ava D ata B ase C onnectivity
SGBD	S ystème de G estion des B ases de D onnées
MVC	M odel V iew C ontroller

Introduction générale

De nos jours, la robotisation est très présente dans l'industrie, ainsi que dans l'informatique. La robotisation des tests en informatique, c'est-à-dire le principe de réaliser les tests par un ordinateur est appelé l'automatisation des tests. La qualité est également très normée, telle que ISO 2000, et il est nécessaire de la contrôler.

Lorsqu'une application est créée ou améliorée dans une entreprise, elle est soumise aux différents types de tests. Ceux-ci permettent de vérifier le bon fonctionnement de ce qui a été développé et contrôler que la solution correspond à ce que le client a demandé.

Je m'attarderai dans un premier temps sur les tests fonctionnels, tout d'abord en expliquant ce que sont les plans et cahier de tests (aussi appelés Recette) utilisés pour les réaliser. Ensuite je vous présenterai les différents outils de recettage qui existent et ceux utilisés dans mon propre domaine professionnel. Enfin, je répondrai à la question concernant la pertinence et la possibilité de l'automatisation totale des tests fonctionnels.

Nous sommes en effet en droit de nous demander si les robots sont en mesure de réaliser le travail effectué par des humains aujourd'hui.

Chapitre I

Contexte général du projet

Introduction

Ce premier chapitre décrit le contexte général du projet qui sera développé en quatre sections. La première section est dédiée à la présentation de l'entreprise d'accueil. La deuxième sera dédiée à la description détaillée de la problématique du projet ainsi que la solution envisagée dans la troisième section.

1. Présentation de l'organisme d'accueil

1.1 Présentation

Neoxia Maroc est une filiale de Neoxia, qui est un cabinet de conseil en gouvernance et architecture des Systèmes d'Information implantée en France.

Neoxia Maroc a ouvert ses portes en juin 2008 et elle a pour mission de reprendre le modèle éprouvé qui a fait le succès de Neoxia, ajusté au marché marocain.

Depuis sa création en 2000, les consultants de Neoxia participent à la conception, la mise en place, et l'évolution des systèmes d'information de clients grands comptes et grandes administrations.

1.2 Lignes de business de Neoxia

Les activités de Neoxia sont structurées en 4 lignes de business qui sont les suivantes :

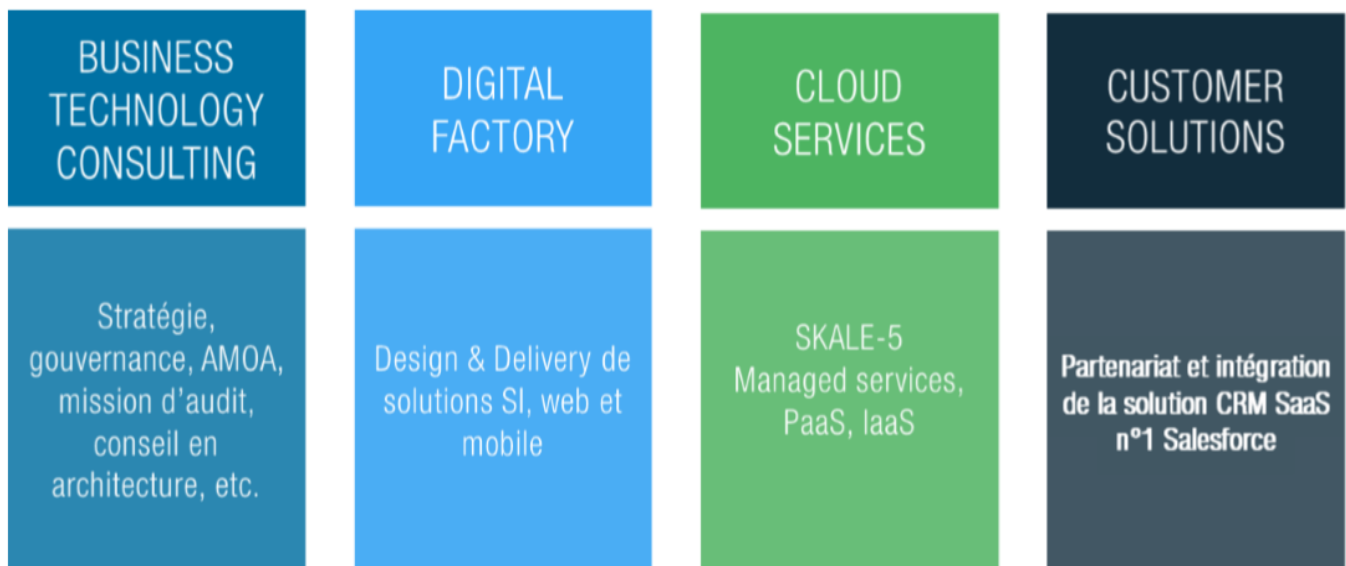


Figure 1 : Lignes de business de Neoxia

Les prestations de Neoxia sont structurées en 3 pôles :

- ✧ Conseil en stratégie et management des SI ;

- ✧ Conseil en architecture des SI ;
- ✧ Technologies et solutions.

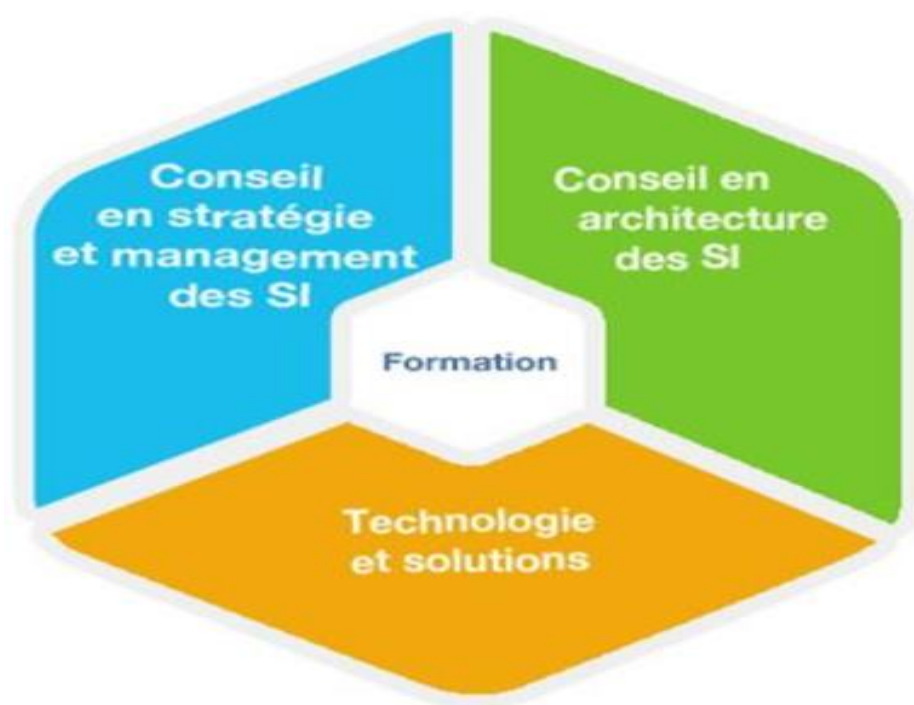


Figure 2 : Offre de prestations Neoxia

1.3 Neoxia en chiffre

1.3.1 Effectif

Neoxia compte plus de 70 collaborateurs en France et au Maroc. La figure ci-dessous montre la répartition de son effectif par type de profil.

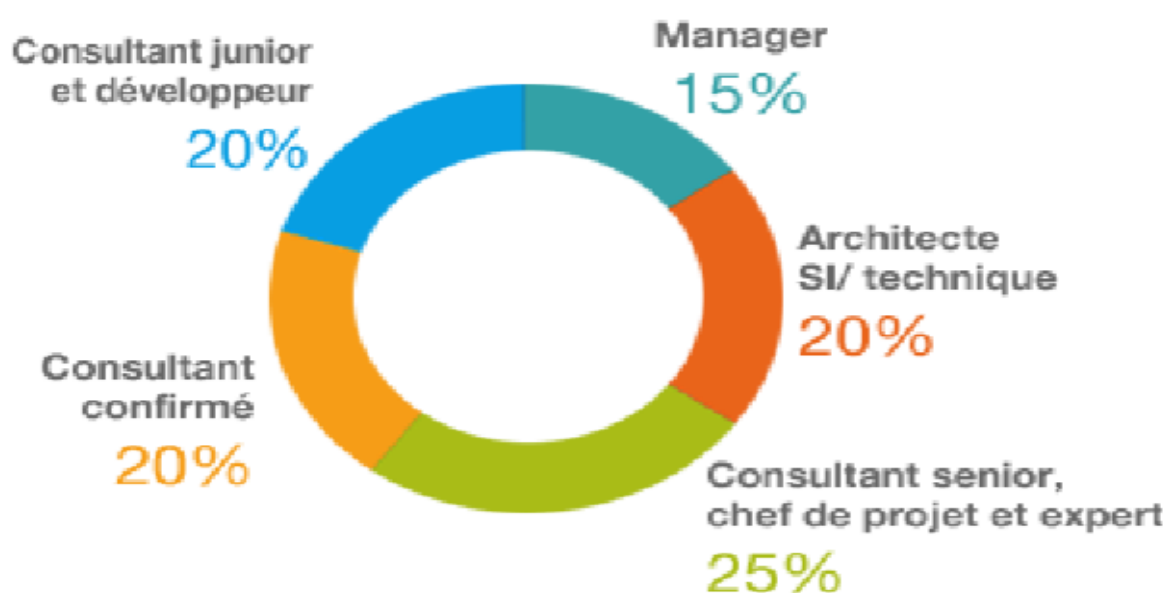


Figure 3 : Répartition de l'effectif par type de profil.

1.3.2 Chiffre d'affaire

Le graphe ci-dessous montre le chiffre d'affaire de Neoxia en France et son évolution soutenue.

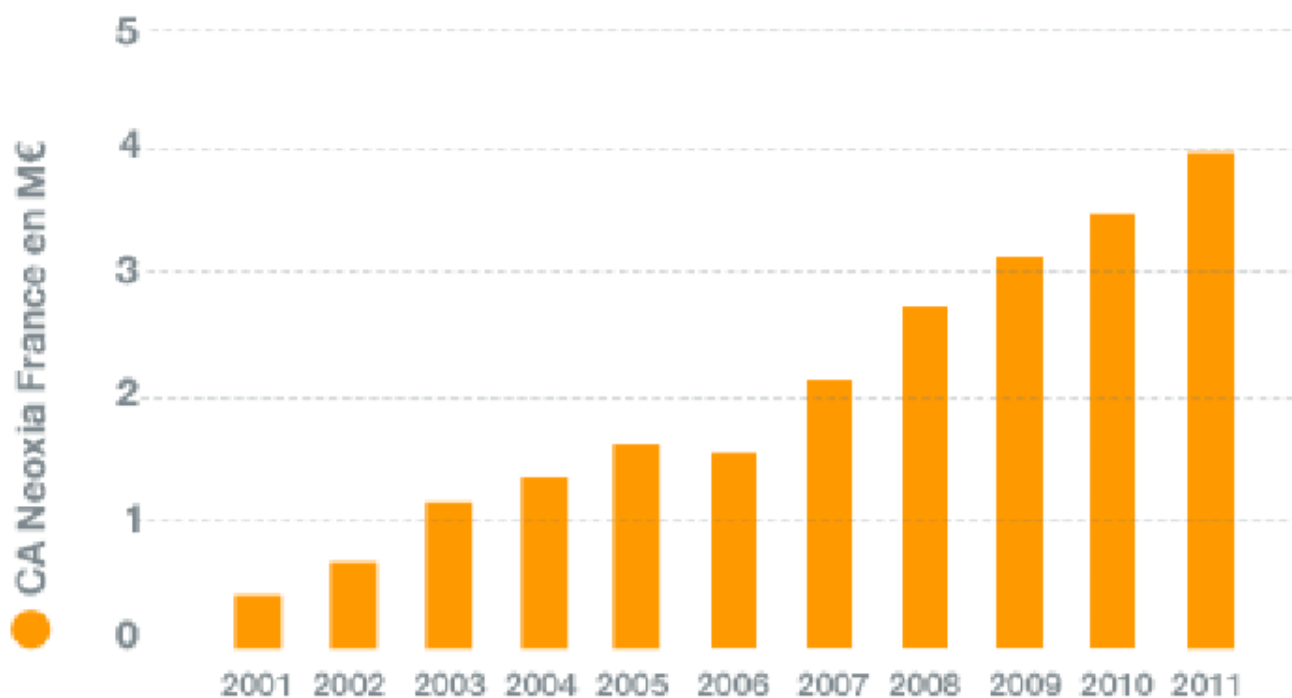


Figure 4 : Chiffre d'affaires Neoxia France en million d'Euros.

Le graphe ci-dessous montre le chiffre d'affaire de Neoxia Maroc et son évolution soutenue.

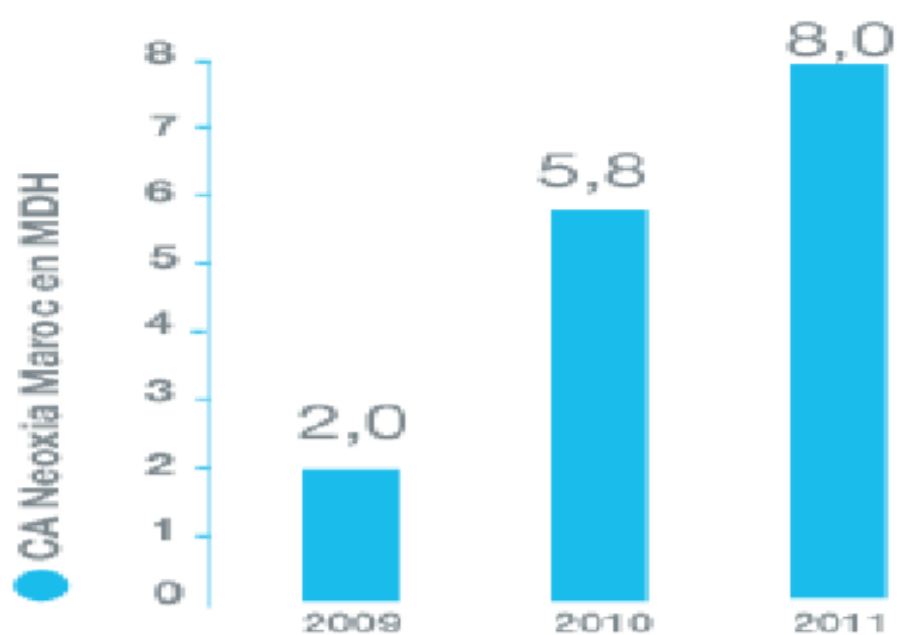


Figure 5 : Chiffre d'affaire de Neoxia Maroc en Million de Dirham.

Le graphe ci-dessous montre la répartition du chiffre d'affaire de Neoxia par secteur d'activité :

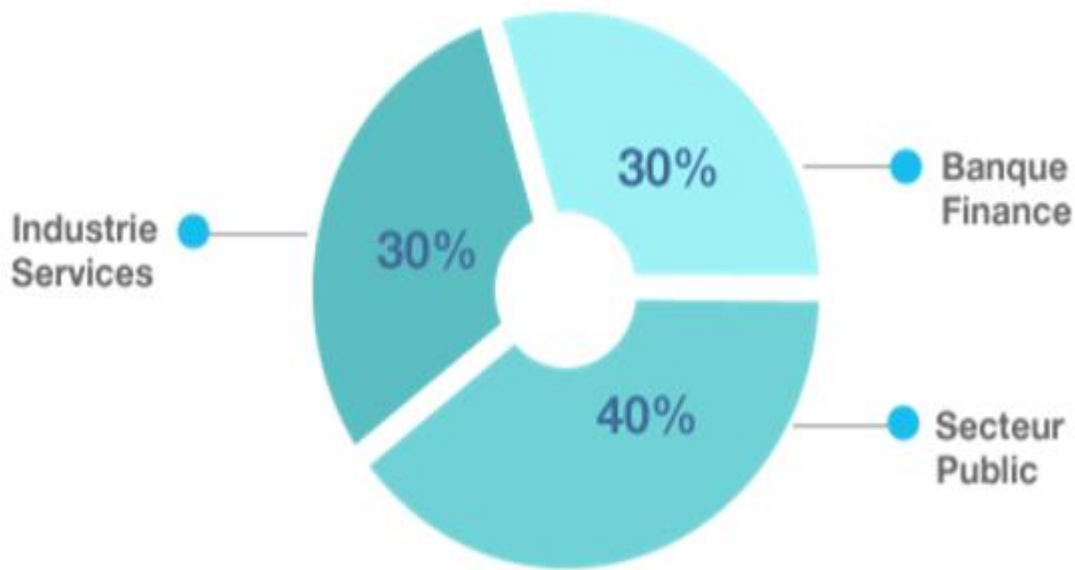


Figure 6 : Chiffre d'affaire par secteur d'activité.

1.4 Offre de service

Neoxia accompagne ses clients pour adresser les enjeux majeurs de leurs SI à différents niveaux :

Conseil en stratégie et management des SI

↳ Stratégie et pilotage

- ✧ Schéma directeur et planification stratégique ;
- ✧ Architecture d'Entreprise ;
- ✧ Cartographie et Urbanisation.

↳ Processus et organisation

- ✧ Management de projets et de portefeuille ;
- ✧ Audit et amélioration des processus ;
- ✧ Conduite de changement ;
- ✧ Management de la sécurité.

↳ Conseil en architecture des SI

✧ Architecture SI

- Architecture orientée services et d'intégration ;
- Management de l'information ;
- Architecture logicielle et socles d'industrialisation ;
- Usine de développement et Socles technique ;
- Ateliers des tests logiciels ;
- Audit et refactoring du code.

✧ Performance et sécurité IT

- Performance applicative et d'infrastructure ;
- Sécurité applicative et d'infrastructure.

✧ Infrastructure

- Architecture et revue d'infrastructure ;

- Cloud computing et Virtualisation ;
 - Industrialisation de la gestion d'infrastructure ;
 - Rationalisation des investissements d'infrastructure.
- ↳ Technologie et solutions
- ✧ Projets SI
 - Assistance à maîtrise d'ouvrage ;
 - Méthodologies agiles ;
 - Coaching technique ;
 - Développement d'applications.
 - ✧ Mobile
 - Développement iPhone, Android, Black Berry ;
 - Solutions mobiles ;
 - Industrialisation du développement mobile ;
 - SI entreprise mobile.
 - ✧ Solutions
 - Outils de management de projets ;
 - Outils pour le management du SI ;
 - Messagerie unifiée et collaboration ;
 - Solutions SI métiers : Agricole, Mutuelle, Etablissements de formation.

Pour assurer cet accompagnement, NEOXIA mène différentes typologies de missions :

- ✧ **Décider** : Planification stratégique, Evaluation de maturité, Organisation, Audit, Études, Urbanisation et Prototypage.
- ✧ **Agir** : Expertise, AMOE, Développement et Intégration, Support, Coaching et Formation, Accompagnement, Co-pilotage, AMOA, Maîtrise des risques, Contrôle qualité.
- ✧ **Capitaliser** : Amélioration, Industrialisation, Réutilisation, Méthodes et bonnes pratiques.



Figure 7 : Différentes typologies de mission de Neoxia.

2. Description détaillée de la problématique

L'identification d'un nombre maximum de comportements problématiques d'un logiciel afin d'en augmenter la qualité. Le test permet de vérifier que le système réagit de la façon prévue par les développeurs tout en étant conforme aux besoins du client.

Les phases de test dans le cycle de développement d'un produit logiciel permettent d'assurer un niveau de qualité défini en accord avec le client. Une procédure de test peut donc être plus ou moins fine, et par conséquent l'effort de test plus ou moins important et coûteux selon le niveau de qualité requis. Aujourd'hui, les métiers dédiés au monde du test commencent à apparaître. C'est en grande partie grâce à une prise de conscience de la complexité et de la criticité des produits. Il est alors important que ces différentes phases soient bien intégrées dans le cycle de développement sur la base de bonnes pratiques et de la rationalisation du processus.

Dans les applications non critiques, l'écriture des tests a longtemps été considérée comme une tâche secondaire. En effet, l'absence d'une forme structurée de test peut engendrer les problèmes suivants :

- ✚ Le code source n'est pas testé en profondeur, ce qui peut donner lieu à des instabilités dans l'application ou à des problèmes de sécurité classique. Les conséquences peuvent être plus ou moins critiques ;
- ✚ Le code source n'est pas suffisamment robuste. Toute modification du code source est susceptible d'engendrer des régressions,
- ✚ Le code source n'est pas réutilisable ni transmissible. Si un autre développeur doit reprendre le travail, il aura d'autant plus de mal à le faire en l'absence d'un protocole de test uniformisé ;
- ✚ Le code source n'est pas évolutif. Plus l'application sera complexe et plus il sera difficile de déceler des erreurs et problèmes de conception de façon empirique. La nécessité peut se présenter à moyen ou long terme de reprogrammer entièrement l'application afin de repartir sur une base saine.

Le test automatisé a pour objectif de simplifier autant que possible les efforts de test grâce aux scripts. Le test est alors exécuté selon celui-ci, les résultats sont signalés et comparés aux résultats des essais antérieurs.

Son principal intérêt réside dans le fait qu'il permet de gagner du temps et de l'argent. En termes de budget, il rend possible des économies sur les charges car l'humain est moins sollicité, si ce n'est pour effectuer la maintenance du test.

En ce qui concerne le temps, le caractère répétitif du test automatisé permet de tester les applications en continu mais aussi de tester plus et mieux. C'est notamment le cas pour les tests de non-régression. Cela favorise une mise en production plus rapide ainsi qu'une réduction des délais de livraison. Aussi, le test automatisé permet une flexibilité au niveau du temps : les tests peuvent être exécutés en dehors des horaires de travail.

Enfin, les campagnes de test peuvent être tracées grâce à l'automatisation des tests car les automates préservent l'historique de l'exécution des tests. Les chefs d'équipe peuvent ainsi assurer un suivi fiable de la qualité d'exécution des tests.

Pour résumer, l'absence d'un protocole de test peut avoir un impact sur la fiabilité d'une application développée (et donc de la qualité de l'application livrée au client) et sur le temps alloué pour le développement et la maintenance du projet (en considérant le temps passé important à rechercher et corriger les anomalies).

3. La solution envisagée

Malgré le fait que les tests automatisés présentent de nombreux avantages pour les entreprises, leur mise en place doit être préparée et mûrement réfléchie afin d'éviter un éventuel échec. Les gains de temps et d'argent qu'ils promettent ne peuvent se faire que sur la durée. En effet, la mise en place des tests automatisés nécessite un investissement conséquent comparé aux tests manuels, notamment au niveau du temps (l'exécution, l'enregistrement et la variabilisation).

De plus, la mise en place de cette méthode de test nécessite non seulement de la recherche, mais aussi du matériel, des logiciels et des personnes ayant des notions en développement.

Tous les cas de tests ne peuvent pas être automatisés, ou sont difficilement automatisables. Il est important de réfléchir en amont à ce qui va et pourra être automatisé, et travailler main dans la main avec les équipes de développement pour faciliter l'intégration des tests automatisés en voyant s'il est possible de les automatiser. Il y a moins d'intérêt à tester un cas de test qui a peu de probabilité d'arriver. Afin de décider si un cas de test peut être automatisé ou non, plusieurs questions sont à se poser :

- ↳ **L'interface graphique de la fonctionnalité est-elle stable ?**
- ↳ **L'application doit-elle être stable afin de supporter le test automatisé.**
- ↳ **Le scénario est-il lourd à reproduire ?**
- ↳ **Le test manuel est-il performant ?**
- ↳ **Le scénario des campagnes de test est-il répétitif ?**

Le test automatisé nécessite une mise à niveau des compétences des testeurs. Ceux-ci devront être formés afin que leurs compétences soient adaptées. Ils devront au préalable disposer de connaissances en programmation afin qu'ils puissent utiliser l'outil d'automation de manière optimale et ainsi mieux résoudre les problèmes liés aux tests de l'application.

Autre point à ne pas négliger : l'entreprise elle-même. En effet, des changements en termes d'organisation et de culture d'entreprise sont à prévoir. Un tel changement n'étant pas anodin, il pourra susciter une certaine résistance de la part des employés.

Pour conclure, l'intelligence artificielle ne remplacera jamais l'intelligence humaine. Les tests automatisés doivent impérativement aller de pair avec les tests manuels, qui devront être effectués à certains niveaux (tests de convivialités) afin de confronter le produit à une utilisation complexe en conditions réelles. Par conséquent, cela permettra de réduire les bugs de manière significative et ainsi livrer une application de meilleure qualité et d'avoir une satisfaction client plus importante.

Chapitre II

Analyse des besoins

1. Cahier des Charges

Parce qu'il est de plus en plus important d'assurer la qualité d'une application web, la mise en place de tests fonctionnels automatisés devient primordiale. Nous tenterons d'établir une introduction à ces pratiques en montrant quelques exemples.

1.1 Objectifs

Il s'agit de l'automatisation des tests fonctionnels, qui permettra aux ingénieurs de test de lancer les tests automatisés à un simple clic et consulter les rapports sur l'état d'exécutions.

Ce projet vise à:

- Exploiter le maximum des outils de test fonctionnel ;
- Préparer un environnement pour lancer et exécuter les tests fonctionnels automatisés ;
- Adapter une rédaction claire pour les scripts de tests ;
- Paramétrer les scripts de test ;
- Récupérer automatiquement les jeux de données pour lancer les tests automatisés.

1.2 Cibles

- Les ingénieurs de test ;
- Les développeurs.

2. Planification

La planification est une tâche très importante pour suivre les étapes de la réalisation d'un projet. C'est l'activité qui consiste à déterminer et à ordonnancer les tâches du projet, à estimer leurs charges et à déterminer les profils nécessaires à leur réalisation.

2.1 Méthodologie de travail

Pour bien gérer un projet, une méthodologie de travail est jugée primordiale afin d'aboutir à un projet de qualité. C'est pourquoi, nous avons choisi **Scrum** comme procédure de gestion de projet.

Méthodes agiles

2.1.1 MÉTHODES AGILE

Les méthodes agiles sont des groupes de pratiques pouvant s'appliquer à divers types de projets, mais se limitant plutôt actuellement aux projets de développement en informatique.

Les méthodes agiles reposent sur une structure (cycle de développement) commune (Itérative, incrémentale et adaptative) et quatre valeurs à savoir :

- ✧ L'équipe : « Les individus et leurs interactions, plus que les processus et les outils »).
- ✧ L'application : « Des logiciels opérationnels, plus qu'une documentation exhaustive »).
- ✧ La collaboration : « La collaboration avec les clients, plus que la négociation contractuelle ».
- ✧ L'acceptation du changement : « L'adaptation au changement, plus que le suivi d'un plan ».

2.1.2 Scrum

QU'EST CE QU'UN PROCESSUS DE DÉVELOPPEMENT ?

Un processus de développement est une séquence d'étapes ordonnées, qui contribuent à l'obtention d'un système logiciel ou à l'évolution d'un système existant. La raison d'être d'un processus de développement est de produire un logiciel de qualité qui répond aux besoins de ses utilisateurs dans un temps et avec un coût prévisible.

Scrum définition ?

Scrum est un processus léger permettant de gérer et de contrôler le développement de produits logiciels. Mettant en œuvre des pratiques itératives et incrémentales, Scrum est, en soi, une méthode efficace. C'est l'une des méthodes agiles les plus employées, car elle permet d'améliorer notablement la productivité tout en accélérant le retour sur investissement.

Scrum repose sur une théorie moderne du contrôle des processus et permet d'élaborer de façon systématique les meilleurs logiciels possibles à partir des ressources disponibles, à un niveau de qualité acceptable et dans le respect des délais de livraison. Le cycle de vie Scrum s'articule en brèves itérations de développement appelées «

Sprints ». Scrum synchronise étroitement les exigences logicielles avec toute une série de prototypes itératifs.

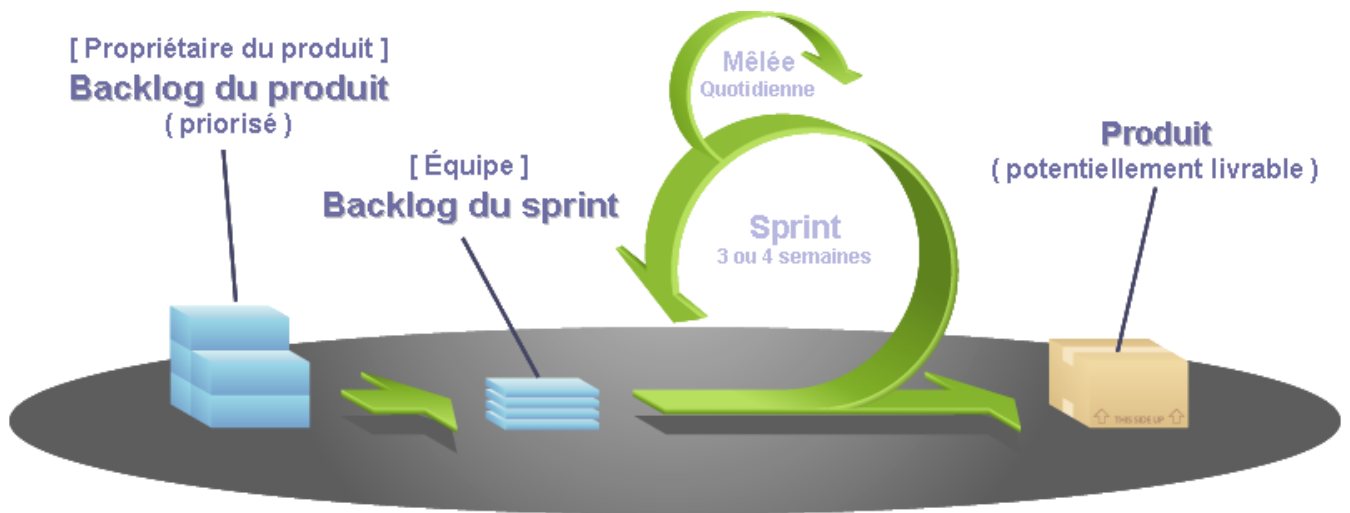


Figure 8 : Vue synthétique du processus Scrum.

Chapitre III

Conception du projet

1. Analyse

1.1 Problématique

Lors de la création d'une nouvelle application ou d'un nouveau logiciel dans une compagnie, celle-ci ne peut se permettre de livrer un logiciel à un client sans vérifier le fonctionnement de celui-ci.

On peut donc affirmer que « **toute modification d'un programme engendrant une nouvelle livraison nécessite une phase de tests** ». Bien qu'idéal, cette affirmation n'est pas toujours respectée. Certaines entreprises ayant des délais très courts et peu de ressources se passent de cette phase ou la réduit au maximum.

Cependant, les grandes entreprises respectent bien souvent cette phase et effectuent différents types de tests. Il en va de même pour les évolutions sur une application déjà existante : chaque livraison doit être précédée de vérifications par l'entreprise.

Le développement d'une application est toujours régi par des règles techniques et métier. Celles-ci doivent être respectées afin de satisfaire la demande du client. Il ne suffit donc pas de développer l'application en question sans se soucier du bon fonctionnement. Livrer un produit à destination d'un client final, avec de multiples erreurs, nuit à l'image de l'entreprise.

1.2 Test fonctionnel

Le test fonctionnel est un service d'assurance qualité qui consiste à s'assurer qu'un système (*ou un de ses composants*) fonctionne adéquatement. Les tests fonctionnels peuvent être effectués durant la phase de développement, au cas par cas pour des sections spécifiques développées par votre équipe, ou au terme du développement de votre projet, une fois qu'il est complet.

L'objectif principal de ce type de test est d'analyser votre produit fini et de déterminer s'il répond bien à vos attentes sur le plan technique. Prenons un exemple. Si vous souhaitez vérifier que le formulaire de contact de votre site web est fonctionnel, il convient de vous poser les questions suivantes :

- ↳ *Avez-vous bien reçu les informations entrées dans le formulaire de contact par un de vos utilisateurs ?*
- ↳ *Votre utilisateur reçoit-il un mail de confirmation après l'envoi de son formulaire ?*
- ↳ *Votre formulaire met-il bien en évidence les erreurs de saisie ?*

↳ *Votre utilisateur est-il bien informé de l'obligation de remplir certains champs vides ?*

Bien entendu, la difficulté et la fréquence des tests dépendent de la complexité du produit que vous souhaitez faire tester. Dans un grand nombre de cas, il convient de fournir un plan de test pour vous assurer que le test est effectué selon vos attentes.

2.1.1 Comment préparer efficacement un test fonctionnel?

Pour effectuer un test fonctionnel de manière efficace, il est important de détailler de manière claire et précise le produit ou les sections de votre produit que vous souhaitez faire tester.

Il est recommandé de transmettre aux testeurs une liste d'objectifs fonctionnels pour chaque section à tester. Ces objectifs doivent indiquer les résultats attendus pour chaque section. Si nous reprenons l'exemple du formulaire de contact, les objectifs pourraient être les suivants :

- ↳ Les visiteurs doivent pouvoir envoyer un message à l'équipe commerciale à travers le formulaire de contact ;
- ↳ Les visiteurs doivent donner leur nom, numéro de téléphone, adresse mail lorsqu'ils contactent l'équipe commerciale.

1.3 Mise en œuvre de tests fonctionnels automatisés

1.3.1 Intérêt des tests automatisés

Les tests fonctionnels manuels sont nécessaires, et seront toujours nécessaires, en informatique comme dans les autres domaines. Imaginez une voiture qui serait vendue aux clients sans aucun essai sur route avec un conducteur humain...

Néanmoins, beaucoup de tests relativement basiques doivent être réalisés très régulièrement durant le cycle de vie d'un logiciel, ce qui rend leur exécution manuelle fastidieuse pour un ROI faible. D'autre part, dans certains contextes (projets web notamment), les tests doivent être réalisés sur différentes plateformes, différents navigateurs de différentes versions, etc.

⇒ Pour toutes ces raisons, il est souvent intéressant d'automatiser certains tests fonctionnels.

1.3.2 Qu'est-ce qu'un test automatisé ?

Un test automatisé est un test dont l'exécution ne nécessite pas l'intervention d'un humain.

L'exécution de tests automatisés requiert donc l'utilisation de solutions informatiques dont le but est d'exécuter des actions, soit spécifiquement dans un navigateur web, soit plus généralement au niveau du système d'exploitation.

1.3.3 Outils de test automatisé

Il existe de nombreux outils de test automatisé. Parmi eux, citons notamment :

- ✧ Selenium;
- ✧ Quality Center;
- ✧ Silk Test;
- ✧ iMacro.

2. Elaboration des diagrammes UML

1. Le langage UML

UML est un standard de modélisation qui permet de représenter l'architecture des logiciels comportant des éléments différents (acteurs, activités ...).

Son utilisation nous a permis de spécifier, visualiser, modifier et construire les documents nécessaires au bon développement de notre application. Notre choix s'est basé surtout sur le fait que c'est :

- Un langage sans ambiguïtés
- Un langage universel pouvant servir de support pour tout langage orienté objet
- Un moyen de définir la structure d'un programme
- Une représentation visuelle permettant la communication entre les acteurs d'un même projet
- Une notation graphique simple, compréhensible même par des non informaticiens

1.1 Identification des acteurs

Un acteur, au sens UML, représente le rôle d'une entité externe (utilisateur humain ou non) interagissant avec le système. Il est représenté par un bonhomme en fil de fer (en anglais *stick man*).

Les acteurs participants dans notre système d'informations sont :

1. Les testeurs ;
2. Les développeurs ou programmeurs ;

3. Les managers de test ;

4. Les chefs de projet.

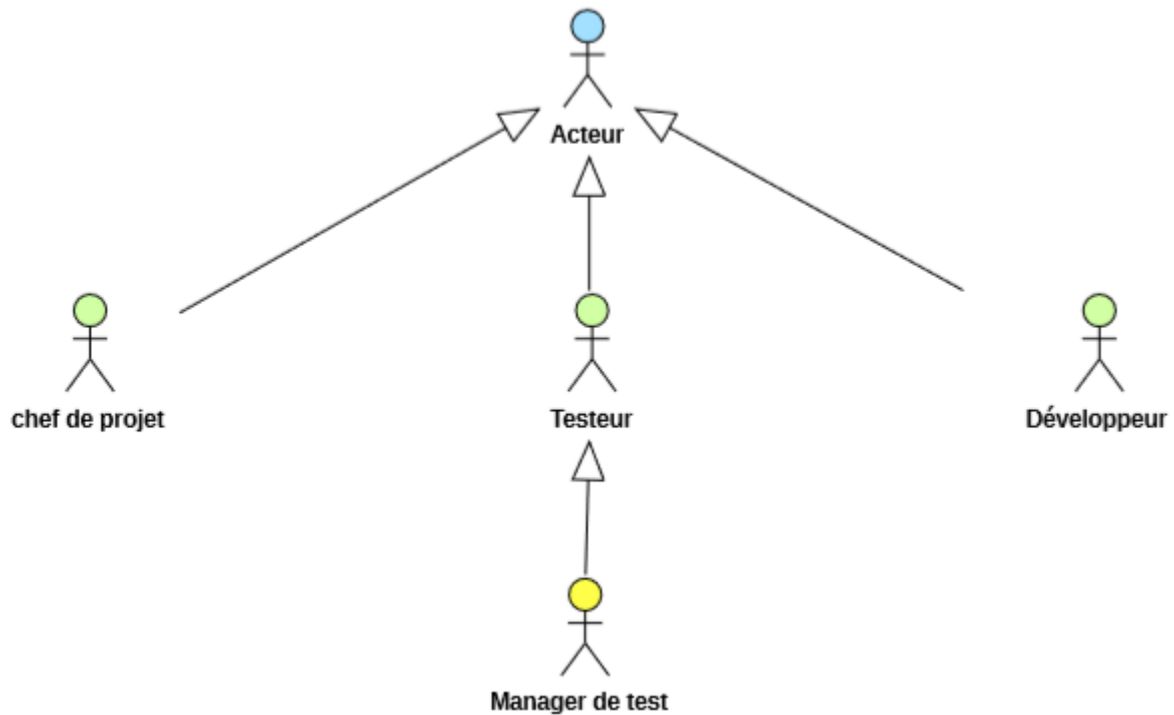


Figure 9 : Les Acteurs

1.2 Diagrammes des cas d'utilisation

Le diagramme de cas d'utilisation permet de représenter visuellement une séquence d'actions réalisées par un système, représenté par une boîte rectangulaire, produisant un résultat sur un acteur, appelé acteur principal, et ceci indépendamment de son fonctionnement interne.

L'objectif de « cas d'utilisation » est de décrire, dans des documents lisibles par tous, la finalité des interactions du système et de ses acteurs.

On commence avec le diagramme de cas d'utilisation qu'a une relation avec les tâches associées aux ingénieurs de tests. On décrit tous les actions possibles pour ces derniers.

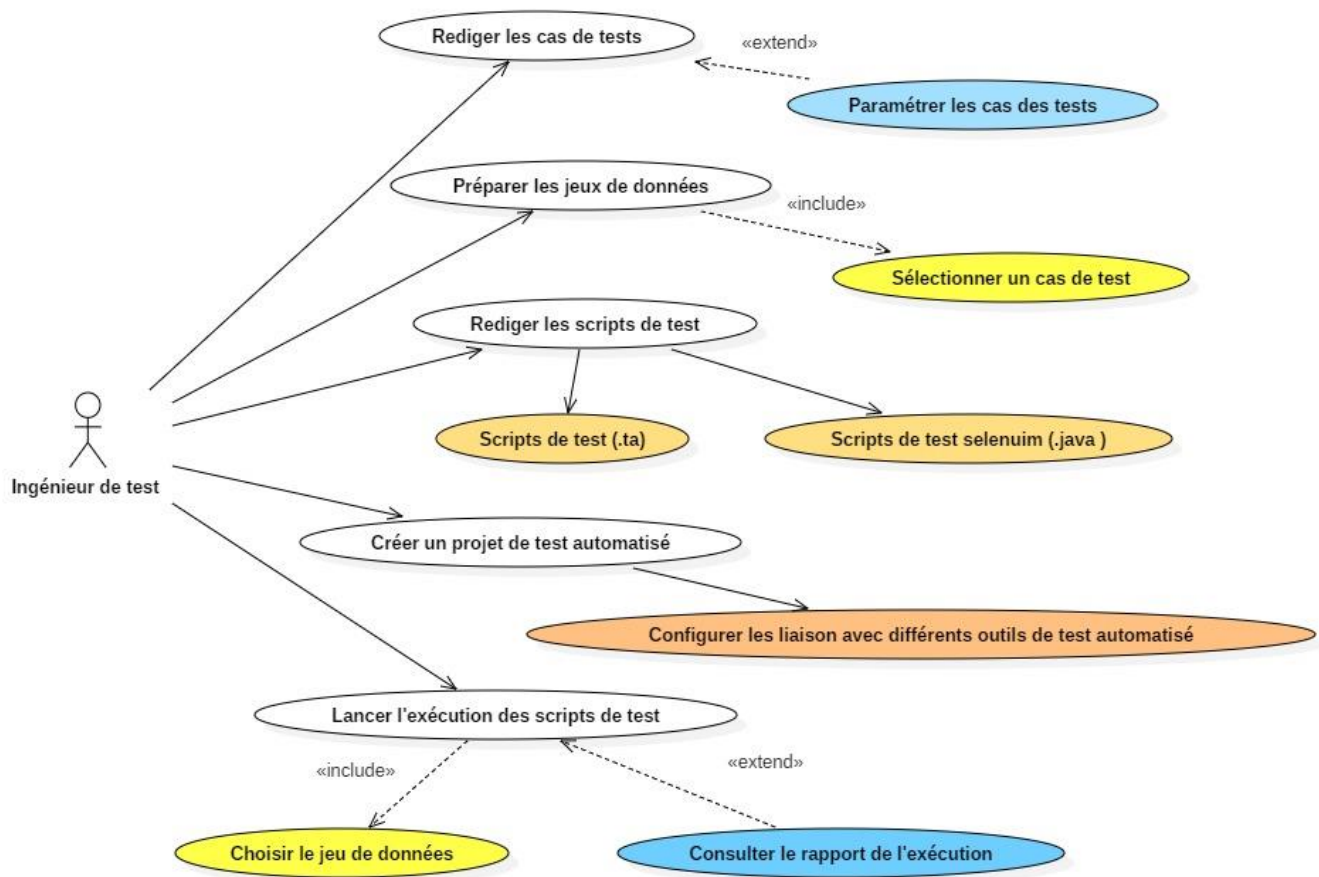


Figure 10 : Cas d'utilisation concernant l'ingénieur de test.

1.3 Description textuelle

Identification:

Titre : Lancer l'exécution des scripts de test.

Objectif : permet aux testeurs de lancer les scripts de test automatisés sous l'outil Squash-TM ainsi de visualiser les résultats et les états des exécutions.

Responsable : le manager de test.

Acteurs : l'ingénieur de test.

Scénario :

Scénario nominal :

1. Le testeur se rendre à la page d'accueil d'outil squash-TM ;
2. S'authentifier ;
3. Consulter l'espace compagnes ;
4. Consulter un projet dans la liste des projets ;
5. Sélectionner l'itération à lancer ;
6. Consulter le rapport du résultat.

Préconditions :

- Avoir choisi des jeux de données pour lancer les cas de tests appels ;
- Avoir accès au projet.

Les exceptions :

- Si le projet d'automatisation n'est pas disponible sous Squash TA ; le test va échouer ;
- Si le testeur n'a pas spécifié un jeu de données pour chaque cas de test ; le test va échouer ;

1.4 Diagramme de séquence

Les testeurs dans notre système d'information ont la responsabilité d'ajouter des cas de test sous l'outil Squash TM.

Après avoir réussi l'authentification le testeur peut consulter l'un des projets déjà créer et ajouter des cas de test, l'ajout des paramètres et la définition de jeu de données ;

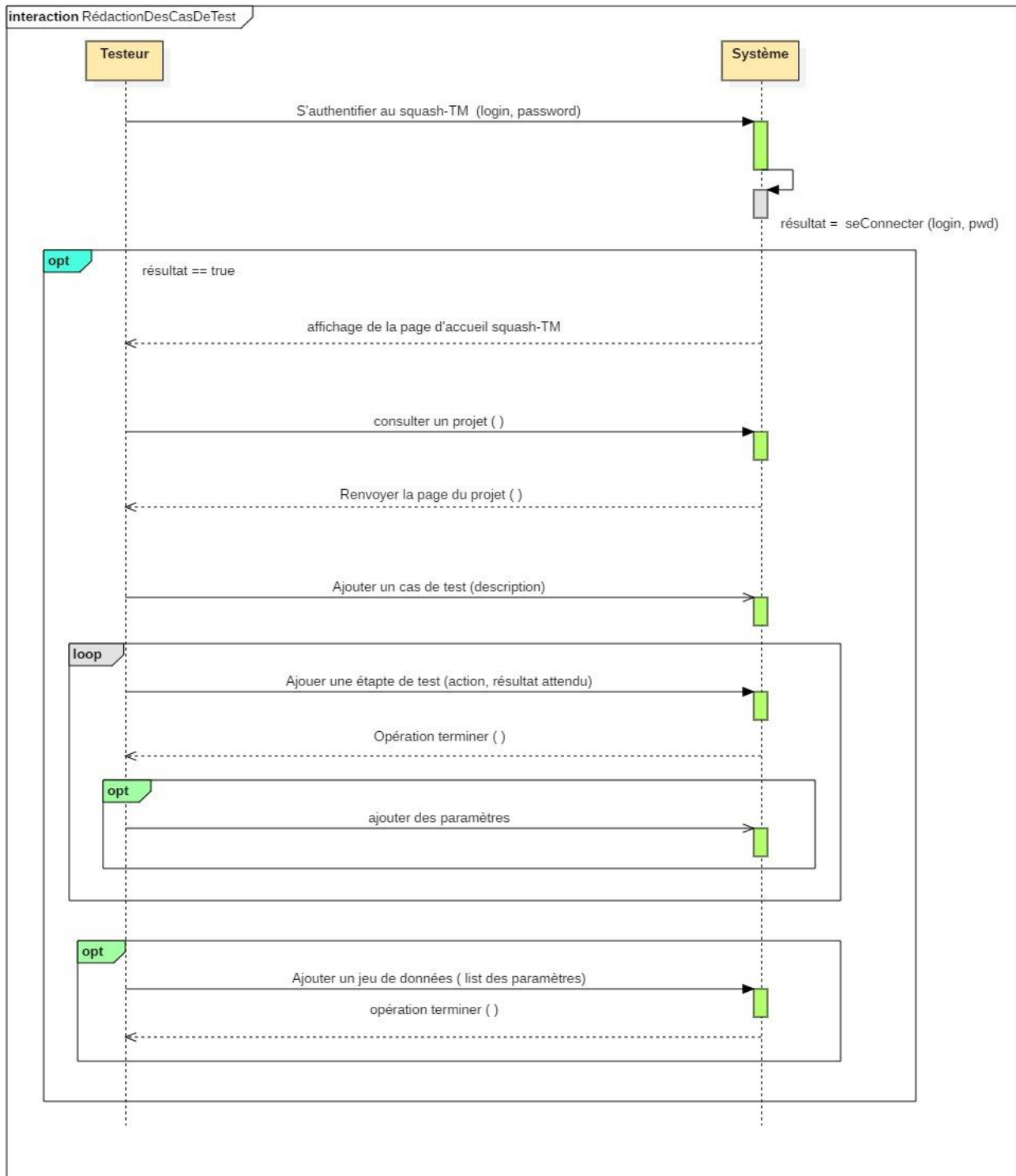


Figure 11 : diagramme de séquence, la rédaction d'un cas de test.

1.5 Diagramme de classe

Ce diagramme représente les cadres de responsabilités entre tous les utilisateurs de l'application web.

- Les utilisateurs sont classés en deux catégories (Demandeur, Administrateur) ;
- Un demandeur ne peut être qu'un doctorant ou un enseignant chercheur ;
- En considère deux types des responsables (type établissement et type présidence).

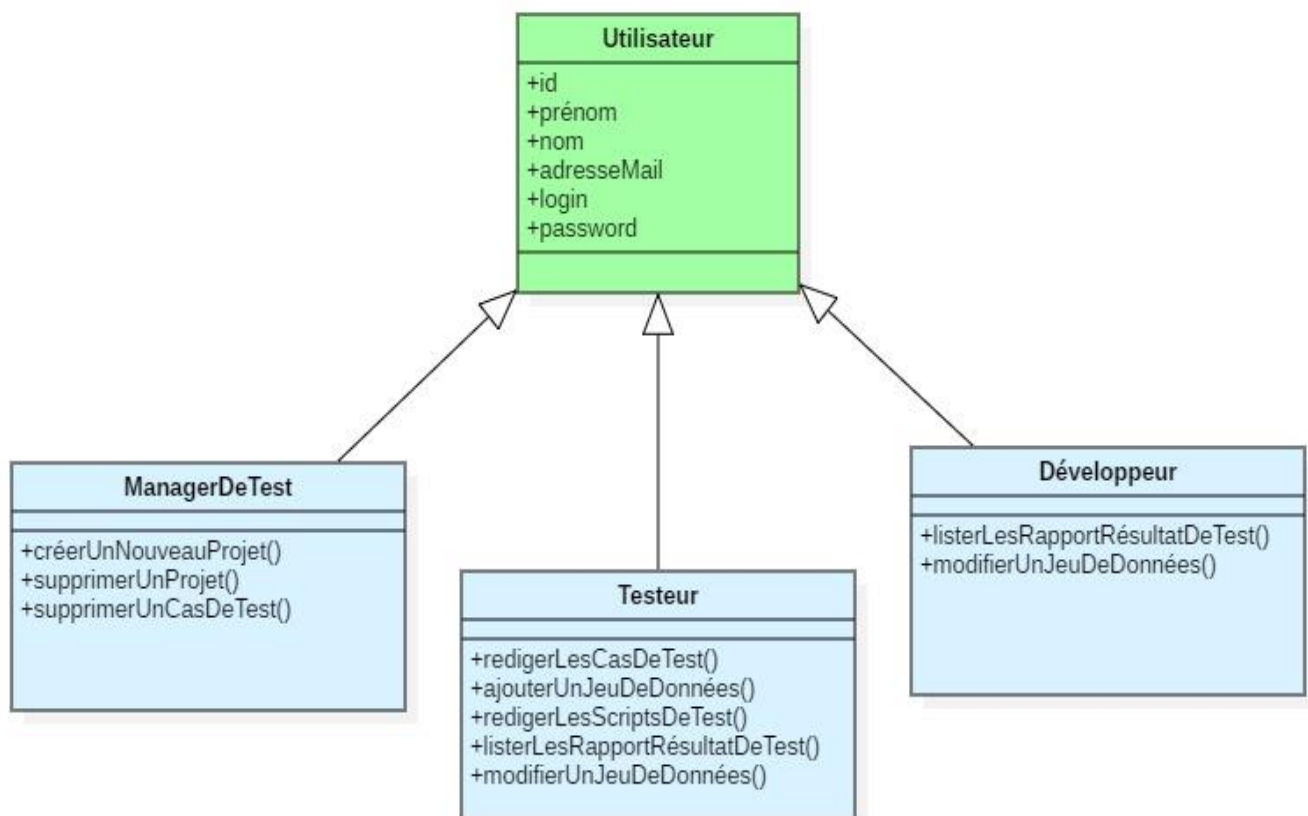


Figure 12 : diagramme de classe.

Chapitre IV

Réalisation du projet

Conclusion et Perspectives

Références

Annexes
