



# **Mise en place des tests d'acceptation dans un environnement agile**

**SABOUNI Ahmed**

**ING3 Informatique**

**Etude bibliographique**

**École d'Ingénieurs du Littoral-Côte-d'Opale**

**5 février 2023**

# Abstract

Insert your abstract text here. Approximately 250-300 words should do the trick !

# Table des matières

	Page
<b>1 Introduction</b>	<b>3</b>
<b>2 Les tests logiciels et leurs automatisations</b>	<b>4</b>
2.1 L'intérêt des tests logiciels . . . . .	4
2.2 Classification des tests . . . . .	4
2.3 les tests d'acceptation et l'automatisation . . . . .	5
<b>3 Les tests agiles</b>	<b>7</b>
3.1 le testing agile et ses objectifs . . . . .	7
3.2 valeurs fondamentales de l'agilité s'appliquent à l'approche de testing agile. . . . .	7
<b>4 Etat de l'art</b>	<b>8</b>
4.1 Outils et méthodes de tests . . . . .	8
4.2 Outils de gestion de projet . . . . .	8
<b>5 Exemple concret</b>	<b>9</b>
5.1 Liaison d'un projet de test automatisé dans un outil de gestion de projet . . . . .	9
<b>6 Conclusions</b>	<b>10</b>
<b>References</b>	<b>11</b>

# 1 Introduction

Les tests d'acceptation sont un élément clé de la qualité des produits logiciels. Ils permettent de vérifier que le produit répond aux exigences du client et peut être utilisé de manière efficace. Dans un environnement agile, où la livraison rapide et la flexibilité sont des priorités, il est important de mettre en place une approche efficace pour les tests d'acceptation.

L'objectif de cet article bibliographique est d'explorer les différents aspects de la mise en place des tests d'acceptation dans un environnement agile. Nous examinerons les avantages et les défis de l'intégration des tests d'acceptation dans une méthodologie agile, ainsi que les meilleures pratiques pour les mettre en œuvre avec succès. Nous explorerons également les outils et les technologies disponibles pour soutenir les tests d'acceptation agiles.

Cet article bibliographique aidera les professionnels du développement logiciel qu'ils soient programmeur, testeurs ou experts métiers qui cherchent à comprendre les tests et leurs rapport avec les méthodes agiles.

## 2 Les tests logiciels et leurs automatisations

### 2.1 L'intérêt des tests logiciels

Les tests logiciels sont un processus systématique de vérification du logiciel pour s'assurer de son fonctionnement correct et de la conformité aux exigences spécifiées. Les tests logiciels sont importants pour garantir la qualité et la fiabilité du produit logiciel. Ils permettent de détecter les erreurs et les bugs avant la livraison, ce qui peut réduire les coûts liés à la correction des erreurs en production. Les tests logiciels peuvent également renforcer la confiance des clients et des utilisateurs finaux dans la qualité du produit, ce qui peut améliorer la réputation de l'entreprise et augmenter les chances de vente. Bien que les tests logiciels nécessitent un investissement en temps et en ressources, ils peuvent finalement permettre de réaliser des économies durant toute la durée de vie des produits.(UMAR, 2020)

### 2.2 Classification des tests

Les tests logiciels peuvent être classifiés en fonction de différents critères tels que la catégorie, le niveau, la technique et les types. Chacun de ces critères a ses propres avantages et inconvénients.

- Catégories : Les tests peuvent être classés en tests dynamiques et tests statiques. Les tests dynamiques peuvent être automatisés, tandis que les tests statiques sont généralement effectués manuellement. Les tests dynamiques ont l'avantage d'être plus rapides et plus fiables, mais peuvent être coûteux à mettre en place. Les tests statiques, quant à eux, sont moins coûteux, mais nécessitent une intervention humaine pour les effectuer et peuvent être plus longs et moins fiables.
- Niveau : Les tests peuvent également être classés en fonction du niveau dans lequel ils sont effectués. Les tests de niveau unitaire sont effectués sur des composants individuels du logiciel. Les tests d'intégration sont effectués lorsque plusieurs composants du logiciel sont combinés et testés ensemble. Les tests de système sont effectués sur le système dans son intégralité. Enfin, les tests d'acceptation sont effectués pour vérifier si le produit répond aux exigences des utilisateurs. Chacun de ces niveaux de test a des avantages et des inconvénients spécifiques en termes de coût, de rapidité et de fiabilité.(TIAN, 2005)
- Technique : Les tests peuvent également être classés en tests fonctionnels et non-fonctionnels. Les tests fonctionnels vérifient si le logiciel fonctionne correctement en répondant aux exigences fonctionnelles. Les tests non-fonctionnels, quant à eux, vérifient les aspects du logiciel tels que la performance, la sécurité et la qualité de l'expérience utilisateur. Les tests fonctionnels ont l'avantage de pouvoir être automatisés et de couvrir les exigences fonctionnelles, mais peuvent ne pas être suffisants pour couvrir les aspects non-fonctionnels du logiciel. Les tests

non-fonctionnels, quant à eux, peuvent être plus difficiles à automatiser et à mesurer, mais ils fournissent une vue complète de la qualité du logiciel.

- Types : Enfin, les tests peuvent également être classés selon leur type spécifique, tels que les tests de sécurité, de performance, d'utilisabilité, etc. Les tests de sécurité sont effectués pour vérifier la sécurité du produit et éviter les menaces potentielles, tandis que les tests de performance sont effectués pour vérifier la rapidité et l'efficacité du produit. Les tests d'utilisabilité sont axés sur l'expérience utilisateur et la facilité d'utilisation du produit.

## 2.3 L'intérêt des tests d'acceptation et l'importance de leur automatisation

Les tests d'acceptation visent à évaluer le logiciel du point de vue de l'utilisateur et se concentrent sur les scénarios d'utilisation, les séquences, les modèles et les fréquences associées. Ils sont généralement plus adaptés aux logiciels lourds et leurs composants en tant qu'ensemble. Ils ont une relation étroite avec les clients et les utilisateurs et sont effectués dans les phases finales de test. L'environnement de test est similaire à celui utilisé par l'utilisateur final. Ces tests doivent être décidé.

	tests d'acceptation
Technique	boite noir
réalisé par	testeurs et developpeurs
Quand ?	à la fin du projet
Comment ?	manuellement ou automatiquement (statique ou dynamique)
Pourquoi ?	pour vérifier si le produit répond aux exigences des utilisateurs

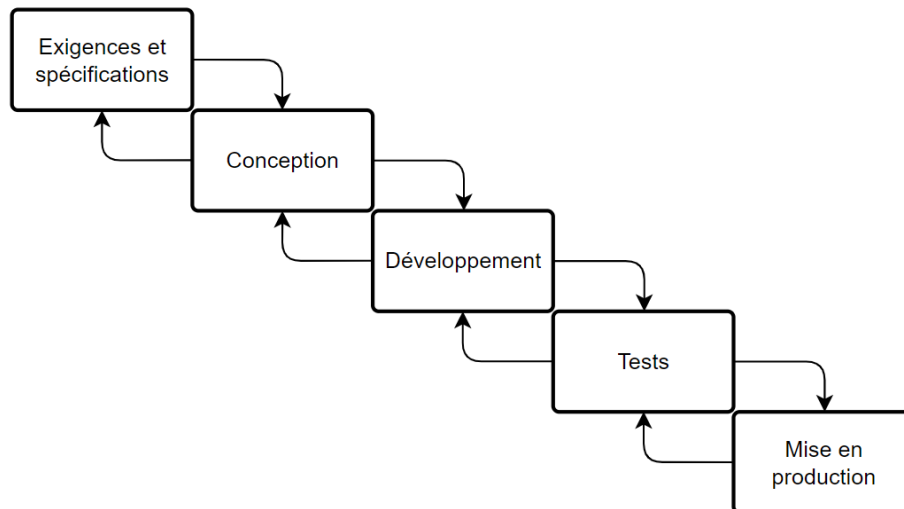
Le test automatisé implique le développement de scripts de test à l'aide de langages de script tels que Python, JavaScript ou Tcl afin que les cas de test puissent être exécutés par des ordinateurs avec un minimum d'intervention humaine. La conception et le développement de tests peuvent être automatisés pour réduire l'effort humain et économiser de l'argent. Le logiciel d'automatisation peut également saisir des données de test dans le système en test, comparer les résultats attendus et réels et générer des rapports de test détaillés. Le test automatisé nécessite des investissements considérables en argent et en ressources. Les cycles de développement successifs nécessiteront l'exécution de la même suite de tests à plusieurs reprises. En utilisant un outil d'automatisation de tests, il est possible d'enregistrer cette suite de tests et de la rejouer au besoin. Une fois la suite de tests automatisée, aucune intervention humaine n'est nécessaire. Le but de l'automatisation est de réduire le nombre de cas de test à exécuter manuellement et non pas d'éliminer complètement les tests manuels. Les avantages du test automatisé sont la rapidité, l'efficacité coût, la répétabilité, la réutilisabilité, la programmabilité, la couverture complète et la fiabilité. Le test automatisé n'est pas toujours nécessaire, approprié ou rentable, mais

une analyse peut aider à déterminer les bénéfices d'un test automatisé par rapport à un test manuel. La bonne gestion peut être obtenue en identifiant et en estimant les coûts et les avantages du test automatisé.

## 3 Les tests agiles

### 3.1 le testing agile et ses objectifs

En général, les équipes de test cherchent à découvrir autant de bogues que possible dans le code développé, car les développeurs n'accordent que peu d'attention à la qualité. Les conséquences sont plus difficiles et coûteuses à corriger. Les équipes de test utilisent des outils pour documenter les cas de test et les bogues et peuvent externaliser la main-d'œuvre, mais cela ne résout pas les problèmes systémiques. Les activités de test intensives ralentissent la livraison et compromettent la qualité, souvent en raison de la pression de livraison tardive des projets et du manque de temps de test. Les défauts découverts à la dernière minute entraînent des retouches coûteuses et des problèmes plus graves si c'est un problème d'architecture ou de conception. Les équipes accumulent également une dette technique qui peut causer des retards futurs. La solution serait de déplacer les tests à des phases plus précoces du cycle de développement.



### 3.2 valeurs fondamentales de l'agilité s'appliquent à l'approche de testing agile.



## **4 Etat de l'art**

### **4.1 Outils et méthodes de tests**

eyfghsdgbksdgkg fkjgh fjkdgh sdfjkghj dfhg sdfjh dfhsg jdfsh gjsdfh gj shdf

### **4.2 Outils de gestion de projet**

dfhgjkhdfjkg kfjghjkdfhgsjkd fkd fjhgsdfjkhgjkdfh gkdfkgsdfkjghjkdfg k

## **5 Exemple concret de mise en place des tests de régressions dans un environnement agile**

### **5.1 Liaison d'un projet de test automatisé dans un outil de gestion de projet**

## 6 Conclusions

Ici la conclusion de mon rapport

# References

- TIAN, J. (2005). *Software Quality Engineering, Tasting Quality Assurance and quantifiable improvement*. Wilson, pp. 74-83. ISBN : 9780471722335, 0471722332.
- UMAR, M. A. (juin 2020). Comprehensive Study of Software Testing: Categories, Levels, Techniques, and Types. English, pp. 709-730.