

Rapport du processus de test



"Application mobile visant à faciliter l'exploration et la découverte du Maroc pour les touristes et les visiteurs."

Réalisé par :

OUGNI Imane

Encadré par :

Mme. Asmaa RETBI

Mme. Asmaa EL KASSIRI

M. Zohair ELMOURABIT

Sommaire

1.	Introduction	3
2.	C'est quoi tester un logiciel ?	3
3.	Les types des tests ?	3
4.	Processus de test	3
4.1.	Planification des tests	3
4.1.1.	Plan de test	4
4.2.	Conception des tests	4
4.2.1.	Scénario de test	5
4.2.2.	Cas de test	5
4.2.3.	Différence entre scénario de test et cas de test	5
4.3.	Implémentation des tests	6
4.3.1.	Script	6
4.3.2.	Code	7
4.3.3.	Les logiciels de tests	7
4.4.	Exécution des tests	7
4.4.1.	Rapport de test	7
4.5.	Clôture des tests	8
4.5.1.	Bilan de test	8
5.	Application du processus de test	8
5.1.	Plan de test	8
5.2.	Scénario de test et cas de test	9
6.	Conclusion	9
7.	Références	9

Liste des figures

Figure 1 : Les étapes du processus de test [1].....	3
Figure 2 : Planification des tests [2]	4
Figure 3 : Sommaire du modèle de plan de test.....	4
Figure 4 : Conception des tests [3]	5
Figure 5 : Modèle de scénario de test [4].....	5
Figure 6 : Modèle de cas de test [5].....	5
Figure 7 : explication de la relation entre de scénario de test et le cas de test [6]	6
Figure 8 : Implémentation des tests [7]	6
Figure 9 : Exécution des tests [8].....	7
Figure 10 : Clôture des tests [9].....	8

1. Introduction

Le processus de test est une composante essentielle du développement de logiciels, visant à garantir la qualité et la fiabilité des applications. À travers ce rapport, nous explorerons les différentes étapes du processus de test, depuis la planification jusqu'à la clôture, en mettant en lumière l'importance de chaque phase et en détaillant les activités associées. Nous illustrerons également l'application pratique de ce processus dans le contexte d'une application dédiée à la découverte du Maroc pour les touristes et les visiteurs.

2. C'est quoi tester un logiciel ?

Tester un logiciel est une activité qui fait partie du processus de développement. Il est mené selon les règles de l'assurance de la qualité et débute une fois que l'activité de programmation est terminée. Il s'intéresse aussi bien au code source qu'au comportement du logiciel. Son objectif consiste à minimiser les chances d'apparitions d'une anomalie avec des moyens automatiques ou manuels qui visent à détecter aussi bien les erreurs.

3. Les types des tests ?

- * **Tests d'acceptation** : ils permettent de vérifier si l'ensemble du système fonctionne comme prévu.
- * **Tests d'intégration** : ils permettent de s'assurer que les composants logiciels ou les fonctions fonctionnent ensemble.
- * **Tests unitaires** : ils permettent de valider que chaque unité logicielle fonctionne comme prévu. Une unité est le plus petit composant testable d'une application.
- * **Tests fonctionnels** : ils permettent de vérifier les fonctions en émulant des scénarios métiers, en fonction des besoins fonctionnels. Les tests en boîte noire sont un moyen courant de vérifier les fonctions.
- * **Tests de performance** : ils permettent d'évaluer les performances du logiciel sous différentes charges de travail. Les tests de charge, par exemple, sont utilisés pour évaluer les performances dans des conditions de charge réelles.
- * **Tests de régression** : ils permettent de vérifier si de nouvelles fonctionnalités brisent ou dégradent la fonctionnalité. Les tests d'intégrité peuvent être utilisés pour vérifier les menus, les fonctions et les commandes au niveau de la surface, lorsqu'il n'y a pas suffisamment de temps pour effectuer un test de régression complet.
- * **Tests de charge** : ils permettent de tester la quantité de contraintes que le système peut supporter avant qu'il ne tombe en panne. Ils sont considérés comme un type de test non fonctionnel.
- * **Tests d'utilisation** : ils permettent de valider la capacité d'un client à utiliser un système ou une application Web pour effectuer une tâche.

4. Processus de test

Le processus de test est un ensemble des étapes essentielles qui garantissent la qualité et la fiabilité d'un logiciel. Il se déroule en parallèle du processus de développement, et les deux se croisent lors de l'exécution des tests.

Les étapes du processus de test :

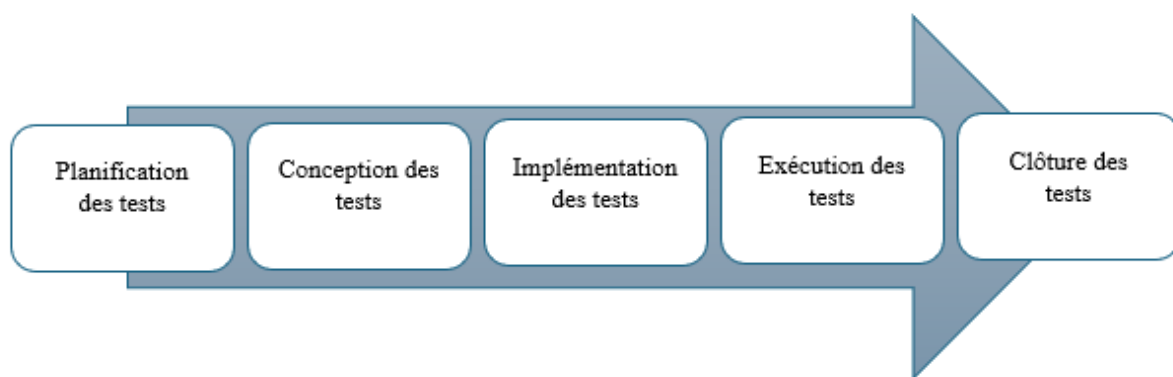


Figure 1 : Les étapes du processus de test [1]

4.1. Planification des tests

La planification des tests consiste à définir les objectifs de test et à spécifier les activités de test à mettre en œuvre pour atteindre les objectifs et la mission. Elle prend en compte les retours des activités de contrôle et de suivi.

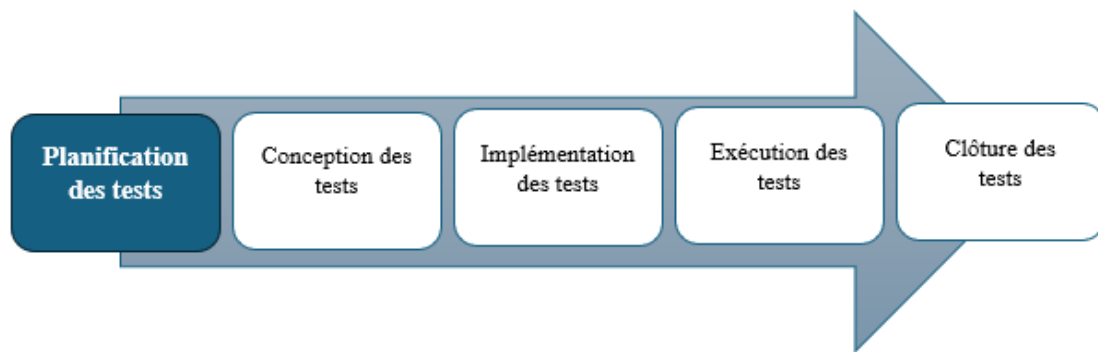


Figure 2 : Planification des tests [2]

4.1.1. Plan de test

Un plan de test est un document détaillé qui décrit la stratégie de test, les objectifs, le calendrier, l'estimation et les livrables, ainsi que les ressources requises pour les tests. Il nous aide à déterminer l'effort nécessaire pour valider la qualité de l'application testée. Il sert à mener les activités de test logiciel en tant que processus défini qui est minutieusement surveillé et contrôlé par le responsable des tests.

1 - Présentation du projet
1.1 - Contexte
1.2 - Description de l'organisation
1.3 - Application à tester
1.4 - Planning du projet
2 - Description de l'application à tester
2.1 - Éléments à tester (fonctionnalités, composants...)
2.2 - Éléments à ne pas tester (fonctionnalités, composants...)
3 - Définition de la stratégie de test
3.1 - Parties prenantes
3.2 - Niveaux de test
3.3 - Priorisation des tests
3.4 - Environnements de test
3.5 - Outils de test
3.6 - Référentiels de test
3.7 - Jeux de données
4 - Définition des responsabilités
4.1 - Matrice RACI

Figure 3 : Sommaire du modèle de plan de test

4.2. Conception des tests

La conception des tests représente les activités ou les objectifs de test généraux sont transformés en des conditions de test et des conceptions de test tangibles.

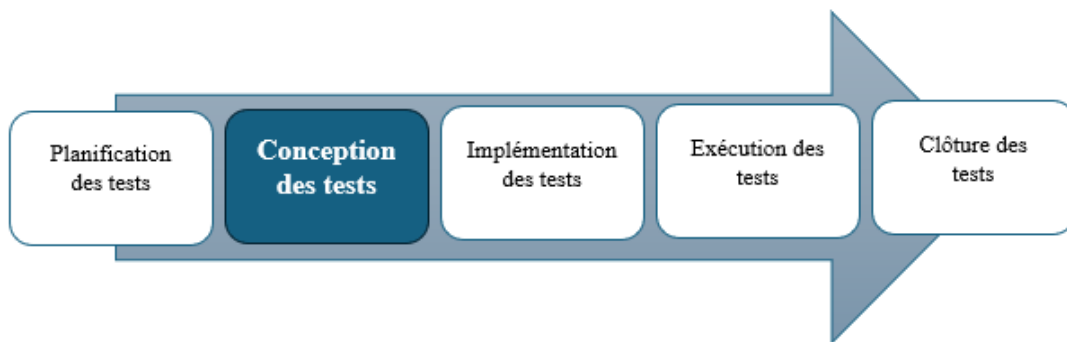


Figure 4 : Conception des tests [3]

4.2.1. Scénario de test

Un scénario de test représente une fonctionnalité ou un aspect du logiciel à tester. Il regroupe plusieurs cas de test qui permettent d'évaluer les aspects positifs et négatifs du projet. Il offre une vision d'ensemble des éléments à tester, aidant ainsi l'équipe à planifier et à exécuter les tests de manière efficace.

	A	B	C
1			
2			Test Scenario Template
3			
4	* Test Cases field is Optional		
5			
6	Test Scenario #	Requirement	Test Scenario Description
7	1	S1.1	Check the Login Functionality
8	2	S1.2	Check the Search Functionality
9	3	S1.3	Check the Product Description Page
10	4	S1.4	Check the Payments Functionality
11	5	S1.5	Check the Order History

Figure 5 : Modèle de scénario de test [4]

4.2.2. Cas de test

Un cas de test est une série d'actions planifiées pour vérifier une fonctionnalité spécifique d'une application. Il comprend des étapes de test, des données d'entrée, des préconditions et des postconditions, et permet de comparer les résultats attendus avec les résultats réels pour garantir que le logiciel répond aux exigences du client.

	A	B	C	D	E	F	G	H	I	J	K
1	Test Case ID	BU_001	Test Case Description	Test the Login Functionality in Banking							
2	Created By	Mark	Reviewed By	Bill	Version	2.1					
3											
4	QA Tester's Log	Review comments from Bill Incorporate in version 2.1									
5											
6	Tester's Name	Mark	Date Tested	1-Jan-2017	Test Case (Pass/Fail/Not	Pass					
7											
8	S #	Prerequisites:			S #	Test Data					
9	1	Access to Chrome Browser			1	UserId = mg12345					
10	2				2	Pass = df12@434c					
11	3				3						
12	4				4						
13											
14	Test Scenario	Verify on entering valid userid and password, the customer can login									
15											
16	Step #	Step Details		Expected Results		Actual Results		Pass / Fail / Not executed / Suspended			
17	1	Navigate to http://demo.guru99.com		Site should open		As Expected		Pass			
18	2	Enter UserId & Password		Credential can be entered		As Expected		Pass			
19	3	Click Submit		Customer is logged in		As Expected		Pass			
20	4										
21											
22											
23											

Figure 6 : Modèle de cas de test [5]

4.2.3. Différence entre scénario de test et cas de test

Voici les différences significatives entre un scénario de test et un scénario de test

Scénario de test	Cas de test
Un scénario de test contient une documentation de haut niveau qui décrit une fonctionnalité de bout en bout à tester.	Les cas de test contiennent des étapes de test définies, des données et des résultats attendus pour tester toutes les fonctionnalités d'une application.
Il se concentre davantage sur « ce qu'il faut tester » que "comment tester".	Un accent complet sur « ce qu'il faut tester » et "comment tester.".
Les scénarios de test sont une seule ligne. Il y a donc toujours une possibilité d'ambiguïté lors des tests.	Les cas de tests ont défini une étape, des prérequis, un résultat attendu, etc. Il n'y a donc aucune ambiguïté dans ce processus.
Les scénarios de test sont dérivés d'artefacts de test tels que BRS, SRS, etc.	Le cas de test est principalement dérivé de scénarios de test. Plusieurs cas de test peuvent être dérivés d'un seul scénario de test
Cela aide de manière agile à tester la fonctionnalité de bout en bout	Il aide à tester de manière exhaustive une application
Les scénarios de test sont des actions de haut niveau.	Les cas de test sont des actions de bas niveau.
Relativement moins de temps et de ressources sont nécessaires pour créer et tester des scénarios.	Plus de ressources sont nécessaires pour la documentation et l'exécution des cas de test.

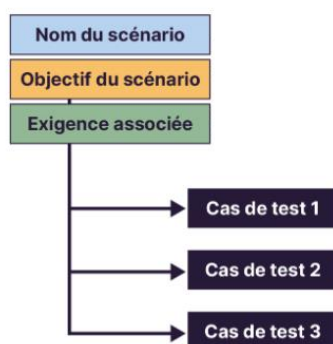


Figure 7 : explication de la relation entre de scénario de test et le cas de test [6]

4.3. Implémentation des tests

L'implémentation des tests consiste à mettre en place et à exécuter des cas de test sur un logiciel pour vérifier ses fonctionnalités et détecter les éventuels défauts. Cela inclut la création de scripts ou de procédures détaillées pour chaque test, ainsi que leur exécution sur le produit logiciel afin de garantir sa qualité et sa fiabilité.

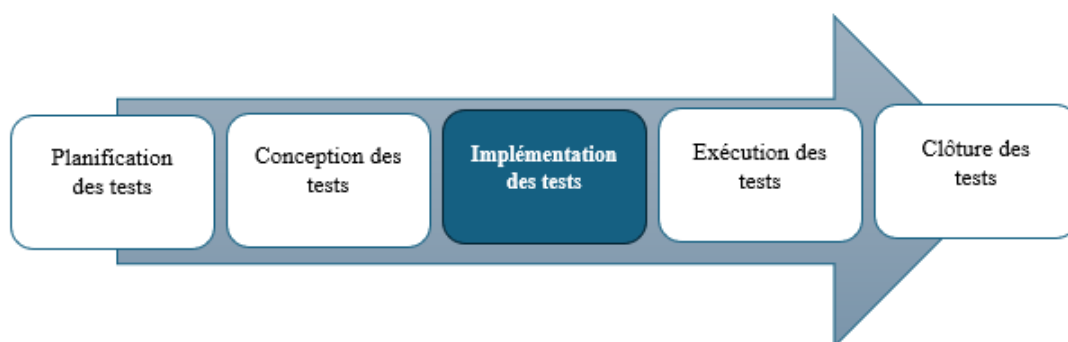


Figure 8 : Implémentation des tests [7]

4.3.1. Script

Un script de test est un ensemble d'instructions détaillées rédigées dans un langage de programmation spécifique pour automatiser l'exécution des tests, visant à valider une fonctionnalité particulière du logiciel. Chaque ligne de ce script décrit les transactions système nécessaires pour tester l'application ou le système, et doit être clairement répertoriée avec les résultats attendus. Ces scripts garantissent une exécution cohérente et reproductible des tests, assurant ainsi la fiabilité et la qualité du logiciel testé.

4.3.2. Code

Le code de test fait référence aux portions de code informatique écrites pour tester spécifiquement une fonctionnalité ou un composant d'une application. Il peut s'agir de petits programmes ou de scripts intégrés directement dans le code source de l'application pour vérifier son fonctionnement. Le code de test est souvent utilisé dans les tests unitaires pour valider le comportement des unités individuelles de code.

4.3.3. Les logiciels de tests

Un logiciel de test est une application ou une suite d'outils conçus pour faciliter la planification, la conception, l'exécution et l'analyse des tests logiciels. Ces outils peuvent inclure des environnements de développement intégrés (IDE) pour l'écriture de tests, des frameworks de test pour l'automatisation, des outils de gestion des cas de test et des rapports, entre autres. Les logiciels de test offrent des fonctionnalités variées pour répondre aux besoins spécifiques des équipes de test dans le processus de développement logiciel.

4.4. Exécution des tests

L'exécution des tests consiste à lancer les cas de test préalablement définis sur le logiciel pour évaluer son comportement. Cela implique de suivre les instructions des scripts ou des procédures de test, d'observer les résultats obtenus et de comparer ces résultats aux attentes définies. L'objectif principal est de détecter les éventuels problèmes ou défauts dans le logiciel afin de les corriger et d'assurer sa qualité.

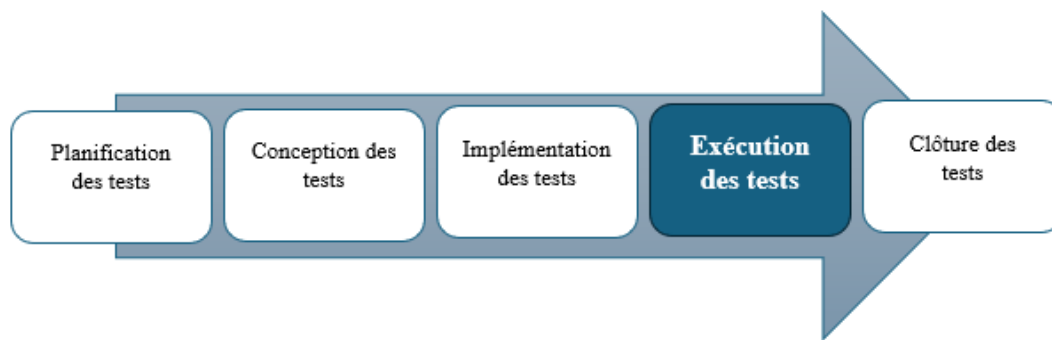


Figure 9 : Exécution des tests [8]

4.4.1. Rapport de test

Un rapport de test est un document qui résume les résultats des tests effectués sur une application ou un système logiciel. Voici ce qu'il pourrait contenir :

Introduction : Une brève description de l'objectif du rapport et du contexte des tests.

Informations sur le logiciel testé : Une présentation succincte du logiciel testé, y compris sa version, sa description et ses fonctionnalités principales.

Méthodologie de test : Une explication des méthodes utilisées pour effectuer les tests, y compris les types de tests (unitaires, fonctionnels, de performance, etc.), les outils utilisés et les critères de succès.

Résultats des tests : Une description détaillée des résultats des tests, y compris les défauts trouvés, les fonctionnalités testées avec succès et celles qui ont échoué, ainsi que toute autre observation pertinente.

Analyse des résultats : Une analyse des résultats, mettant en évidence les tendances, les problèmes récurrents, les zones de risque et les améliorations potentielles.

Conclusion : Une conclusion qui résume les principaux résultats des tests, évalue la qualité globale du logiciel testé et propose des recommandations pour l'amélioration éventuelle du produit.

Annexes : Toute documentation supplémentaire, tels que des captures d'écran, des traces de débogage, des données de performance, etc.

4.5. Clôture des tests

La clôture des tests marque la fin du processus de test. Cela implique la documentation des résultats, l'évaluation de la couverture des tests, la rédaction de rapports et la communication des conclusions. C'est essentiellement la phase de conclusion et de clôture du travail de test.

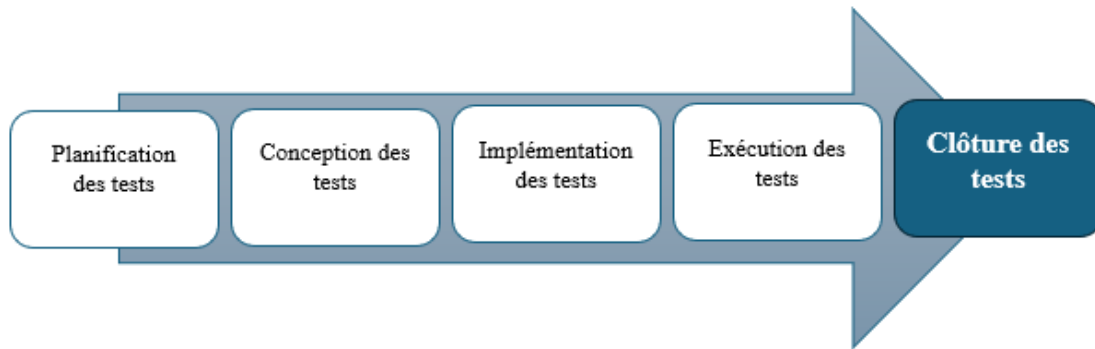


Figure 10 : Clôture des tests [9]

4.5.1. Bilan de test

Un bilan de test est une évaluation globale des résultats et de l'efficacité des tests réalisés sur un logiciel ou un système. Voici ce qu'il pourrait inclure :

Objectifs atteints : Une liste des objectifs de test définis au départ et une évaluation de leur réalisation. Cela implique de vérifier si les tests ont couvert toutes les fonctionnalités et les exigences spécifiées.

Couverture des tests : Une analyse de la couverture des tests, en évaluant si tous les scénarios et cas de test prévus ont été exécutés et si des zones de risque ont été correctement testées.

Défauts identifiés : Un résumé des défauts et des problèmes trouvés pendant les tests, y compris leur gravité, leur fréquence et leur impact potentiel sur le logiciel.

Efficacité des tests : Une évaluation de l'efficacité des tests, en examinant si les méthodes et les outils de test ont permis de détecter les défauts de manière appropriée et en temps opportun.

Qualité du logiciel : Une évaluation générale de la qualité du logiciel testé, en tenant compte des défauts identifiés, de leur gravité et de leur impact sur l'expérience utilisateur.

Recommandations : Des suggestions pour améliorer les processus de test, les méthodologies, les outils ou le logiciel lui-même, afin d'optimiser la qualité et la fiabilité du produit final.

Perspectives : Des recommandations pour les futurs cycles de test, y compris des domaines spécifiques à améliorer, des nouvelles fonctionnalités à tester et des ajustements potentiels aux méthodes de test.

5. Application du processus de test

En me basant sur les rapports précédents et les concepts évoqués au début de ce rapport, j'ai appliqué le processus de test à mon application dédiée à faciliter l'exploration et la découverte du Maroc pour les touristes et les visiteurs. Cela implique la création d'un plan de test détaillé, ainsi que le développement de scénarios de test et de cas de test spécifiques pour évaluer les différentes fonctionnalités de l'application.

5.1. Plan de test

Cette section détaille le plan de test élaboré pour guider et organiser les activités de test tout au long du processus.

Le fichier suivant inclut ces éléments : [Plan de test.pdf](#)

5.2. Scénario de test et cas de test

Cette partie couvre les scénarios de test et les cas de test spécifiques développés pour évaluer les différentes fonctionnalités du logiciel.

Le fichier suivant inclut ces éléments : [les scenarios et cas de test.pdf](#)

6. Conclusion

Le processus de test est bien plus qu'une simple vérification de fonctionnalités ; il s'agit d'une démarche rigoureuse visant à assurer la qualité et la fiabilité des logiciels. De la planification minutieuse à la conception des tests, en passant par l'implémentation et l'exécution, chaque étape joue un rôle crucial dans la détection et la correction des défauts. En appliquant ce processus à notre application de découverte du Maroc, nous avons pu évaluer de manière approfondie ses fonctionnalités et garantir une expérience utilisateur optimale. Ainsi, le processus de test demeure un pilier fondamental du développement logiciel, garantissant des produits de qualité et répondant aux attentes des utilisateurs finaux.

7. Références

[1] [2] [3] [7] [8] [9] : [1.4 Processus de test fondamental – Test Academy](#)

[4] [5] : [Cas de test vs scénario de test – Différence entre eux \(guru99.com\)](#)

[6] : [Assemblez vos cas de test en scénario - Concevez une stratégie de test - OpenClassrooms](#)