



Cas de test en génie logiciel

Charles Guin-Vallerand
Colombiano Kedowide

Octobre 2013

Introduction

Le test est l'exécution ou l'évaluation d'un système ou d'un composant par des moyens automatiques ou manuels, pour vérifier qu'il répond à ses spécifications ou identifier les différences entre les résultats attendus et les résultats obtenus.

IEEE (Standard Glossary of Software Engineering Terminology)

Tester, c'est exécuter le programme dans l'intention d'y trouver des anomalies ou des défauts.

G. Myers (The Art of Software testing)

Il existe deux grandes familles de tests : les tests fonctionnels (ou tests boîte noire) et les tests structureaux (ou tests boîte de verre).

Les tests fonctionnels permettent de valider une solution proposée à la partie prenante lors du développement d'un produit logiciel. Ce qui permet ainsi de vérifier si toutes les spécifications logicielles fournies par la partie prenante sont respectées. Les tests effectués vérifient les exigences de la partie prenante du point de vue des cas d'utilisation. Le but essentiel de ces tests est de vérifier la validité des données, de leur traitement et de leur récupération ainsi que des règles d'affaire. Chaque grand cas d'utilisation est validé isolément puis par la suite tous les cas d'utilisation sont validés ensemble. L'intérêt est de valider la stabilité d'une solution par rapport aux différents modules qui la composent, en soumettant cette solution à un ensemble d'actions représentatives de ce qui sera fait une fois que le produit logiciel sera en production. La technique de la boîte noire permet de réaliser ces tests. Elle consiste dans un premier temps à une vérification du produit logiciel et de ses processus internes et par la suite à une analyse des résultats.

Pourquoi les tests fonctionnels sont-ils basés sur des cas d'utilisation? Après tout, les cas d'utilisation sont des besoins fonctionnels et/ou comportementaux. Ces besoins indiquent donc ce que le système devra faire. Dans le processus unifié, ce sont des cas d'utilisation qui sont recommandés pour détecter et définir les besoins. Ils sont donc plus aptes pour vérifier si ces besoins sont bien respectés. Il faut donc créer au moins un **cas de test** pour le scénario principal et également au moins un cas de test pour chaque scénario alternatif.

Pour effectuer ces tests, il faut d'abord définir une stratégie de tests. Cette stratégie consiste à identifier les techniques et les critères de complétion de tests. Il est recommandé d'exécuter ces tests dans un environnement sécurisé avec des données connues et contrôlées. Pour les tests fonctionnels, vous devez donc définir les objectifs du test, déterminer les techniques utilisées, définir les critères de complétion et déterminer et décrire les questions internes ou externes qui ont un impact sur l'implémentation et l'exécution d'un test fonctionnel.

Identification des scénarios de test

Les scénarios définis dans les jeux de tests sont généralement fondés sur des spécifications valides. Pour ce faire, il faut alors se servir du déroulement normal d'un cas d'utilisation. Pour pouvoir obtenir un jeu de tests complet, il faut également identifier les cas d'utilisation qui génèrent des erreurs, des utilisations inhabituelles donc des cas d'utilisation exceptionnels. Chacun des cas d'utilisation constituera un scénario de test. Un scénario de test permet de contrôler l'enchaînement des étapes qui sont prévues de façon formelle ou implicite dans les spécifications fonctionnelles.

En somme, pour chaque scénario de tests, il faut que toutes les étapes qui se déroulent sans ambiguïté soient identifiées à partir des spécifications et sans redondances, par la suite, il faut aussi que les préconditions et les post-conditions soient identifiées et il faut finalement choisir les données de tests. L'objectif des données de tests est de s'assurer de la cohérence des données saisies (format des données, des contraintes sur les données etc.) et de la vérification de la cohérence des actions du système sur les données. Le tableau ci-dessous montre un exemple de stratégie pour les cas de tests. Cette stratégie est fondée sur la définition de l'objectif de test, les techniques utilisées, les critères de complétion et d'autres considérations particulières.

Stratégie pour les cas de tests	
Objectif de test :	Permet de vérifier si le cas d'utilisation choisi rencontre la bonne cible de test, y compris la navigation, la saisie de données, le traitement et la récupération.
Technique :	Il faut exécuter chaque chemin du cas d'utilisation ou fonction en utilisant des données valides et non valides préalablement choisies tout en vérifiant: <ul style="list-style-type: none"> • Les résultats attendus avec des données valides (attendus). • Les messages d'erreur et d'avertissement lorsque des données non valides sont utilisées.
Critère de complétion :	Tous les tests prévus ont été exécutés. Toutes les anomalies identifiées ont été enregistrées.
Considérations particulières :	Quels sont les impacts sur l'implémentation et l'exécution d'un test fonctionnel

Pour définir une bonne stratégie pour les cas de tests, il faut préalablement un cas d'utilisation détaillé comme le propose **Alistair Cockburn** dans son livre *Writing Effective Use Cases*. L'idée est de fournir un format de présentation textuelle à la fois souple et riche. En nous inspirant du livre d'Alistair Cockburn sur la question (Cockburn, 2003), nous retiendrons le format suivant :

Cas d'utilisation	le nom du cas d'utilisation
Acteur	Le nom du ou des acteurs principaux, déclencheurs du cas
Événement déclencheur	l'événement déclencheur du cas
Parties prenantes et intérêts	listes des parties prenantes et leurs intérêts
Niveau	une valeur parmi : stratégique, objectif utilisateur, sous-fonctionnalité
Portée	la portée du cas (entreprise, sous-système ou système)
Pré-conditions	les pré-conditions éventuelles
Post-conditions	les post-conditions éventuelles
Scénario nominal	
	1. description de l'action
	2. description de l'action
	3. ...
	4. description de la dernière action avant la fin du cas
Extensions	
	• numéro de l'étape : condition : action ou sous-cas d'utilisation
	• numéro de l'étape : condition : action ou sous-cas d'utilisation
Contraintes	
...	
Questions-ouvertes	
...	
Annexes	
...	

A partir de ce format de cas d'utilisation, nous proposons le format suivant pour les cas de test :

FORMAT DE CAS DE TEST

Cas de test	Numéro du cas
Nom du cas de test	Un nom significatif
Auteur du cas de test	Nom de personne ou d'entreprise
Système ou sous-système	Le ou le sous-système sur lequel on fait les cas
Une courte description du cas de test	
Date d'exécution	

Les pré-conditions

Étapes	Actions	Réponses attendues	Succès/Échec	Commentaire

Les post-conditions

Exemple

Avant de faire le cas de test, nous allons définir le cas d'utilisation et les stratégies de test. Soit un cas d'utilisation qui permet de changer le numéro d'identification personnel (NIP). Ci-dessous le cas d'utilisation :

Cas d'utilisation	Changer NIP
Acteur	Client
Événement déclencheur	NIP expiré
Parties prenantes et intérêts	Client de la banque
Niveau	Objectif sécurité client
Portée	Auto-gestion sécurité compte utilisateur
Pré-conditions	Carte de banque valide, accès à un guichet automatique, connaître son NIP courant, le sous-système de NIP doit afficher le menu principale
Post-conditions	Confirmation que le NIP a été bien changé
Scénario nominal	
	1. Entrer la carte de banque dans le guichet automatique
	2. Entrer le NIP courant
	3. Entrer le nouveau NIP
	4. Entre de nouveau NIP
	5. Confirmer le changement NIP en cliquant sur le bouton « oui »
	4. description de la dernière action avant la fin du cas
Extensions	
	<ul style="list-style-type: none"> le guichet automatique ne fonctionne pas : les cas s'arrêtent
	<ul style="list-style-type: none"> le sous-système informe l'utilisateur que le NIP à l'étape 3 est différent de celui à l'étape 4. Il faut retourner à l'étape 2.
	<ul style="list-style-type: none"> l'utilisateur a oublié de confirmer le changement de NIP. Il faut retourner à l'étape 2.
Contraintes	
...	
Questions-ouvertes	
...	
Annexes	
...	

A partir de ce cas d'utilisation, nous allons dans un premier temps élaborer les stratégies de tests et par la suite un cas de test proprement dit.

Stratégie pour les cas de tests : « changer NIP »	
Objectif de test :	Permet de vérifier si le cas d'utilisation « changer NIP » rencontre la bonne cible de test, y compris la navigation, la saisie de données, le traitement et la récupération.
Technique :	<p>Il faut exécuter chaque chemin du cas d'utilisation « changer NIP » en utilisant des données valides et non valides préalablement choisies (des entiers naturels de quatre chiffres) tout en vérifiant:</p> <ul style="list-style-type: none"> • Les résultats attendus avec les entiers naturels valides (attendus). • Les messages d'erreur et d'avertissement lorsque des données ne correspondant pas à des entiers naturels sont utilisées.
Critère de complétion :	<p>Tous les tests prévus ont été exécutés.</p> <p>Toutes les anomalies identifiées ont été enregistrées.</p> <p>Vérification des post-conditions</p>
Considérations particulières :	Protéger les clients contre la fraude financière

Nous allons présenter ici deux types de cas de test : un cas de tests simple et un autre un peu plus complexe.

cas de test simple « changer NIP »

Cas de test	Cas -1
Nom du cas de test	Test de «Changer NIP»
Auteur du cas de test	La banque ABC
Système ou sous-système	Système de NIP
Une courte description du cas de test	Permet de vérifier le bon fonctionnement du système de NIP
Date d'exécution	11 mars 2013

Les pré-conditions

Carte de banque valide, accès à un guichet automatique, connaître son NIP actuel = 2451, le sous –système de NIP doit afficher le menu principale.

Étapes	Actions	Réponses attendues	Succès/ Échec	Commentaire
1	Cliquer sur le bouton «Changer NIP»	Le système affiche un message demandant à l'utilisateur d'entrer son nouveau PIN		
2	Entrer le NIP	Le système affiche un message demandant à l'utilisateur de confirmer son NIP en le re-entrant de nouveau		
3	Entrer de nouveau le NIP	Le système affiche un message de succès de l'opération et demande à l'utilisateur s'il veut faire		

		d'autres opérations		
4	Cliquer sur le bouton OUI	Le système affiche le menu principal		
5	Vérifier la postcondition			

Les post-conditions

Confirmation que le NIP a été bien changé.

cas de test complexe « changer NIP »	
Cas de test	Cas -1-bis
Nom du cas de test	Test de «Changer NIP»
Auteur du cas de test	La banque ABC
Système ou sous-système	Système de NIP
Une courte description du cas de test	Permet de vérifier le bon fonctionnement du système de NIP
Date d'exécution	11 mars 2013

Les pré-conditions

Carte de banque valide, accès à un guichet automatique, connaître son NIP actuel = 2451, le sous –système de NIP doit afficher le menu principale.

Étapes	Actions	Réponses attendues	Succès/ Échec	Commentaire
1	Cliquer sur le bouton «Changer NIP»	Le système affiche un message demandant à l'utilisateur d'entrer son nouveau PIN		
2	Entrer le NIP actuel (=7894)	Le système affiche un message demandant à l'utilisateur de confirmer son NIP en le re-entrant de nouveau		
3	Entrer de nouveau le NIP actuel (=7894)	Le système affiche un message de succès de l'opération et demande à l'utilisateur s'il veut faire d'autres opérations		
4	Cliquer sur le bouton «OUI»	Le système affiche le menu principal		
5	Vérifier la post-condition 1			
6	Répéter les étapes 1, 2 et 3 en utilisant un autre NIP (1234) puis cliquer sur le bouton « NON»	Le système affiche un message en demandant au client d'insérer sa carte le guichet automatique.		
7	Vérifier la post-condition 2			
8	Répéter les étapes 1, 2 en utilisant le NIP (2587)	Le système affiche demandant au client de confirmer son NIP en le rentrant à nouveau.		

9	Entrer un mauvais NIP (4512)	Le système un message d'erreur et demande au client de rentrer un NIP correcte.		
10	Entrer le NIP (2587)	Le système que le NIP a été changé avec succès et demande au client s'il veut faire d'autres opération.		
11	Cliquer sur le bouton « NON»	Le système affiche un message attestant que le client ne voulait pas faire d'autres opérations		
12	Vérifier la post-condition 3			

Les post-conditions

1. Confirmation que le NIP (7894) a été bien changé.
2. Confirmation que le NIP (1234) a été bien changé.
3. Confirmation que le NIP (2587) a été bien changé.