Fiche méthodologique de projet

T.3 Mettre en oeuvre des méthodes de test

Objectif

Appliquer des méthodes de test.

Les tests statiques

Les méthodes de tests statiques consistent en l'analyse textuelle du code du logiciel afin d'y détecter des erreurs, sans exécution du programme.

Les méthodes utilisées sont : revue de code, analyse des types, analyse du domaine des variables, ...

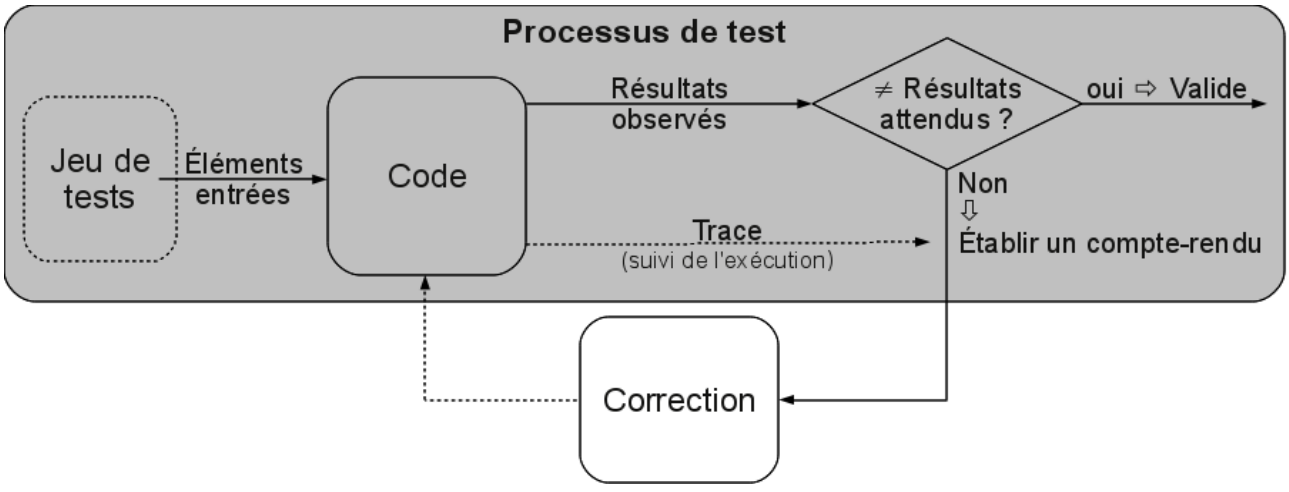
Les revues de code permettent l'examen détaillé d'une spécification, d'une conception ou d'une implémentation par une personne ou un groupe de personnes (lecture croisée), afin de déceler des fautes, des violations de normes de développement ou d'autres problèmes.

Avantages :
Méthodes efficaces et peu coûteuses (60 à 95% des erreurs sont détectées lors de contrôles statiques). Donc, les méthodes de tests statiques sont nécessaires.

Inconvénients :
Méthodes ne permettant pas de valider le comportement d'un programme au cours de son exécution. Donc, les méthodes de tests statiques ne sont pas suffisantes.

Les tests dynamiques

Les méthodes de tests dynamiques consistent en l'exécution du programme à valider à l'aide d'un jeu de tests. Elles visent à détecter des erreurs en confrontant les résultats obtenus à ceux attendus par la spécification.



Méthode aléatoire

Ces méthodes ne garantissent pas une bonne couverture de l'ensemble des entrées du programme. En particulier, elles peuvent ne pas prendre en compte certains cas limites ou exceptionnels. Ces méthodes ont donc une efficacité très variable.

Méthode structurelle (Boîte blanche)

Le jeu de tests repose sur le code du programme. Le jeu de tests est choisi de manière à remplir certaines exigences :
- couverture de toutes les instructions.
- couverture de tous les chemins exécutables.
- couverture de toutes les conditions.

Méthode fonctionnelle (Boîte noire)

Le jeu de tests est dérivé de la spécification du programme. Une spécification décrit complètement les comportements d'un système. La méthode de tests fonctionnelle vise à valider les fonctionnalités d'un programme.

Méthode expérimentale

Le jeu de tests est sélectionné sur la base de l'expérience.
Une base de données contenant toutes les erreurs découvertes dans un logiciel A peut servir de guide lors de la sélection du jeu de tests d'un logiciel B.

Les classes d'équivalence

On peut réduire le nombre de ces tests en utilisant la technique des classes d'équivalence. Cette technique consiste à identifier des classes d'équivalence dans le domaine des données d'entrées vis à vis d'une propriété d'une donnée de sortie. Tout test effectué avec une entrée quelconque appartenant à une classe d'équivalence déterminé entraîne un résultat soit correct (classe valide), soit incorrect (classe invalide).

Une fois les classes déterminées, il suffit de prendre au moins un représentant pour chacune de ces classes (jeu de test).

Plan de test

Un plan de test se représente par un tableau contenant : une description du test, les valeurs en entrées du module et le résultat attendu. On numérotera chaque test (phase).

Exemple pour un test unitaire :

Testeur :		Date :	
Module testé :			Version : 1.0
Classe	Description	Valeurs en entrée	Résultats attendus
valide n°1	3 valeurs positives égales	(2, 2, 2)	3
valide n°2	3 valeurs positives dont deux égales telle que la somme des deux valeurs égales soit supérieure à la troisième	(3, 5, 5)	2
valide n°3	3 valeurs positives différentes telle que la somme de deux d'entre elles soit supérieure à la troisième	(3, 7, 9)	1
invalide n°4	Au moins une valeur négative	(-1, 6, 9)	-1
invalide n°5	3 valeurs positives telle que la somme de deux d'entre elles soit inférieure à la troisième	(3, 8, 4)	-2