



Génie logiciel : les tests

Karën Fort

karen.fort@univ-lorraine.fr / <https://members.loria.fr/KFort>

Quelques sources d'inspiration

- ▶ Plus que de l'inspiration, j'ai repris le cours de Martine GAUTIER (FST, Univ. de Lorraine), avec son accord.
- ▶ Cours de Lydie du Bousquet, avec son accord.
- ▶ Cours de Xavier Blanc (Univ. de Bordeaux) :
<https://www.youtube.com/watch?v=hBCaoN421Qs>

Les tests et vous

Les types de tests

Quand tester ?

Que tester ?

Pour finir



1 Allez sur wooclap.com

2 Entrez le code d'événement dans le bandeau supérieur

Code d'événement
FFHHKL

 Activer les réponses par SMS

Exercice

A quoi servent les tests ?

Exercice

Avez-vous déjà fait des tests ?

Exercice

Citez différents types de tests

Ce qu'est un test

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]

Cours de Martine Gautier, avec son accord.

Ce que n'est pas un test

Attention !

« Le test de programmes peut être une façon très efficace de montrer la présence de bugs mais est **désespérément inadéquat pour prouver leur absence** »

[Edsger Dijkstra, "The Humble Programmer", Communications of the ACM, vol. 15, n10, 1972, p. 859-866. (Wikipédia)]

Inspiré du cours de Martine Gautier, avec son accord.

Plus concrètement

Tester un programme, c'est :

- ▶ l'exécuter
- ▶ en maîtrisant les données en entrée
- ▶ en s'assurant que le comportement est celui attendu

Inspiré du cours de Lydie du Bousquet, avec son accord.

Plus concrètement

Tester un programme, c'est :

- ▶ l'exécuter
- ▶ en maîtrisant les données en entrée
- ▶ en s'assurant que le comportement est celui attendu

→ Maîtrise des entrées :

- ▶ Sélectionner = choisir
- ▶ assez de données, mais pas trop

Inspiré du cours de Lydie du Bousquet, avec son accord.

Plus concrètement

Tester un programme, c'est :

- ▶ l'exécuter
- ▶ en maîtrisant les données en entrée
- ▶ en s'assurant que le comportement est celui attendu

→ Maîtrise des entrées :

- ▶ Sélectionner = choisir
- ▶ assez de données, mais pas trop

→ Vérification du comportement :

- ▶ pouvoir observer le comportement
- ▶ adopter une vision critique

Les tests et vous

Les types de tests

Quand tester ?

Que tester ?

Pour finir

Quels types d'erreurs recherche-t-on ?

On attend du logiciel certaines propriétés :

- ▶ absence d'erreur (correction fonctionnelle),
- ▶ robustesse,
- ▶ performances,
- ▶ utilisabilité, ...

Inspiré du cours de Lydie du Bousquet, avec son accord.

Quels types d'erreurs recherche-t-on ?

On attend du logiciel certaines propriétés :

- ▶ absence d'erreur (correction fonctionnelle),
- ▶ robustesse,
- ▶ performances,
- ▶ utilisabilité, ...

Caractérisation des tests selon l'objectif :

- ▶ Test de conformité
- ▶ Test de robustesse
- ▶ Test de performances, de montée en charge

Inspiré du cours de Lydie du Bousquet, avec son accord.

Tests statiques vs tests dynamiques vs tests formels

Tests statiques

Revue de code, de spécifications, de documents de conception

Tests dynamiques

Exécuter le code pour s'assurer d'un fonctionnement correct (60 % des tests actuellement)

Tests formels

Vérifications formelles : Preuves

Cours de Martine Gautier, avec son accord.

Tests dynamiques : pour quoi faire ?

- ▶ Aujourd'hui : vérifier que ce qui est implanté correspond à ce qui est attendu (respect des spécifications)
- ▶ Demain : s'assurer que les modifications n'ont pas cassé ce qui était correct (non-régression)

Cours de Martine Gautier, avec son accord.

Tests boîte blanche vs tests boîte noire

Tests boîte noire

On exécute le logiciel sur différentes données et on compare les résultats produits avec les résultats attendus.

vs

Tests boîte blanche

On teste chaque classe, chaque fonction l'une après l'autre. Les tests passent dans tous les chemins d'exécution possibles.

Cours de Martine Gautier, avec son accord.

Tests boîte blanche vs tests boîte noire

Tests boîte noire

On exécute le logiciel sur différentes données et on compare les résultats produits avec les résultats attendus.

⇒ Indépendants du code

vs

Tests boîte blanche

On teste chaque classe, chaque fonction l'une après l'autre. Les tests passent dans tous les chemins d'exécution possibles.

Cours de Martine Gautier, avec son accord.

Tests boîte blanche vs tests boîte noire

Tests boîte noire

On exécute le logiciel sur différentes données et on compare les résultats produits avec les résultats attendus.

⇒ Indépendants du code

vs

Tests boîte blanche

On teste chaque classe, chaque fonction l'une après l'autre. Les tests passent dans tous les chemins d'exécution possibles.

⇒ Liés au code

Tests boîte noire : pour quoi faire ?

- ▶ S'appuyer sur le cahier des charges établi en accord avec le client = contrat qui engage le client et l'équipe de développement
- ▶ Le cahier des charges impose des scénarios d'exécution qu'il convient de reproduire

Cours de Martine Gautier, avec son accord.

Tests boîte blanche : pour quoi faire ?

- ▶ Tester classe par classe, fonction par fonction (théorie)
- ▶ S'appuyer sur :
 - ▶ la documentation de la classe et de la fonction à tester
 - ▶ les instructions qui constituent le corps de la fonction

Adapté du cours de Martine Gautier, avec son accord.

Tests unitaires vs tests d'intégration vs tests de validation

Tests de validation

On teste l'application (point de vue utilisateur)

Tests d'intégration

On teste un composant et son intégration avec les autres

Tests unitaires

On teste une unité de code (niveau fonction)

Exercice

Quels types de tests sont des tests boîtes noires ?

Exercice

Qu'appelle-t-on les "monkey tests" ?

Les tests et vous

Les types de tests

Quand tester ?

Que tester ?

Pour finir

Exercice

Quand faut-il tester ?

Quand tester ?

On peut tester à différents moments :

- ▶ Au cours du développement de chaque fonction
- ▶ Quand on intègre les modules entre eux
- ▶ Avant et pendant la livraison
- ▶ Après les modifications (maintenance, évolution. . .)

Inspiré du cours de Lydie du Bousquet, avec son accord.

Quand tester ?

On peut tester à différents moments :

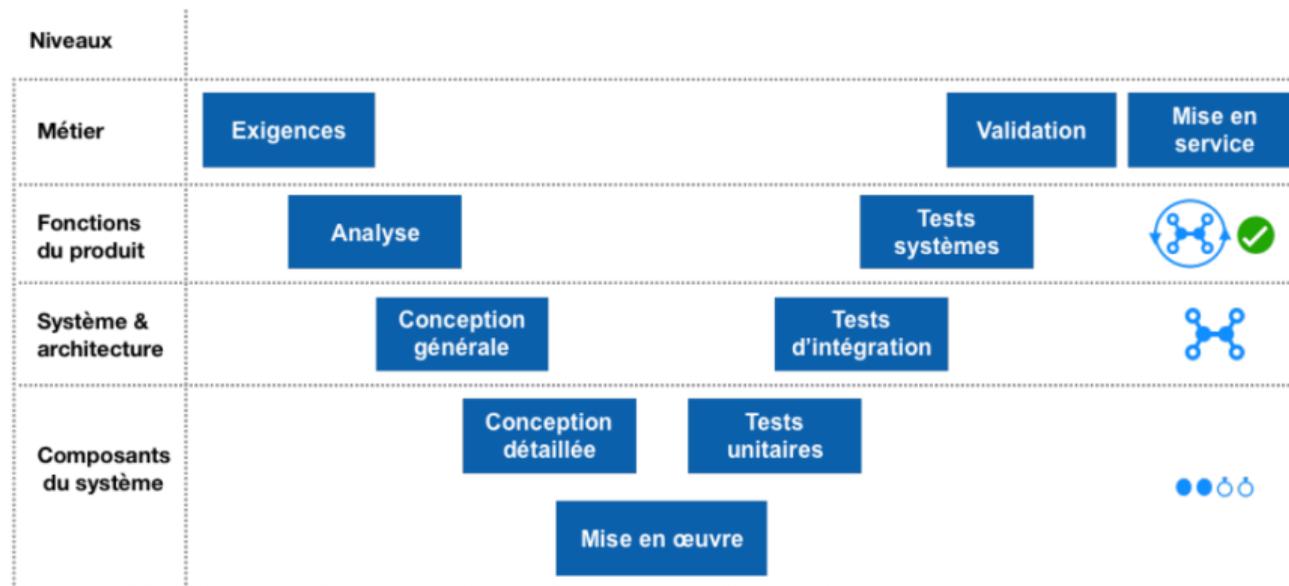
- ▶ Au cours du développement de chaque fonction
- ▶ Quand on intègre les modules entre eux
- ▶ Avant et pendant la livraison
- ▶ Après les modifications (maintenance, évolution...)

Caractérisation des tests par rapport aux phases de développement :

- ▶ Test unitaire
- ▶ Test d'intégration
- ▶ Test système
- ▶ Test d'acceptation (ou de recette)
- ▶ Test de non-régression

Rappel : modèle en V

Approche séquentielle (= cascade), mais le test du produit est planifié en parallèle de chaque phase de développement (mais réalisé après) :



Les tests et vous

Les types de tests

Quand tester ?

Que tester ?

Pour finir

Choisir les données des tests

- ▶ La donnée est dans un domaine infini (ex : un point) :

Cours de Martine Gautier, avec son accord.

Choisir les données des tests

- ▶ La donnée est dans un domaine infini (ex : un point) :
 - ▶ Identifier des classes d'équivalence d'objets supposés se comporter **de la même façon**

Cours de Martine Gautier, avec son accord.

Choisir les données des tests

- ▶ La donnée est dans un domaine infini (ex : un point) :
 - ▶ Identifier des classes d'équivalence d'objets supposés se comporter **de la même façon**
 - ▶ Souvent, on identifie un élément particulier, tout seul dans une classe

Cours de Martine Gautier, avec son accord.

Choisir les données des tests

- ▶ La donnée est dans un domaine infini (ex : un point) :
 - ▶ Identifier des classes d'équivalence d'objets supposés se comporter **de la même façon**
 - ▶ Souvent, on identifie un élément particulier, tout seul dans une classe
- ▶ La donnée est dans un domaine borné (ex : un intervalle) :

Cours de Martine Gautier, avec son accord.

Choisir les données des tests

- ▶ La donnée est dans un domaine infini (ex : un point) :
 - ▶ Identifier des classes d'équivalence d'objets supposés se comporter **de la même façon**
 - ▶ Souvent, on identifie un élément particulier, tout seul dans une classe
- ▶ La donnée est dans un domaine borné (ex : un intervalle) :
 - ▶ Tester une valeur quelconque mais aussi les **valeurs limites**

Cours de Martine Gautier, avec son accord.

Choisir les données des tests

- ▶ La donnée est dans un domaine infini (ex : un point) :
 - ▶ Identifier des classes d'équivalence d'objets supposés se comporter **de la même façon**
 - ▶ Souvent, on identifie un élément particulier, tout seul dans une classe
- ▶ La donnée est dans un domaine borné (ex : un intervalle) :
 - ▶ Tester une valeur quelconque mais aussi les **valeurs limites**
- ▶ La donnée est une collection :

Cours de Martine Gautier, avec son accord.

Choisir les données des tests

- ▶ La donnée est dans un domaine infini (ex : un point) :
 - ▶ Identifier des classes d'équivalence d'objets supposés se comporter **de la même façon**
 - ▶ Souvent, on identifie un élément particulier, tout seul dans une classe
- ▶ La donnée est dans un domaine borné (ex : un intervalle) :
 - ▶ Tester une valeur quelconque mais aussi les **valeurs limites**
- ▶ La donnée est une collection :
 - ▶ S'inquiéter du **taux de remplissage** de la collection

Cours de Martine Gautier, avec son accord.

Exemple/exercice

- ▶ Programme qui lit un mot à partir du clavier.
- ▶ Si le mot a un nombre pair de lettres, le programme écrit :
“L contient un nombre pair de lettres”
- ▶ Sinon le programme écrit :
“L contient un nombre impair de lettres”

→ Proposez des tests pour cet exemple

Inspiré du cours de Lydie du Bousquet, avec son accord.

Principes et réalisation

Produire les tests à partir de la spécification

- ▶ « Couverture des fonctionnalités »
- ▶ « Couverture des cas d'utilisation »

Ici, la spécification décrit explicitement 2 cas généraux :

- ▶ Mot avec nombre de lettres pair
- ▶ Mot avec nombre de lettres impair

On peut imaginer des cas limites :

- ▶ Mot vide
- ▶ Mot très long
- ▶ Mot avec un ou deux caractères

Principes et réalisation

Produire les tests à partir de la spécification

- ▶ « Couverture des fonctionnalités »
- ▶ « Couverture des cas d'utilisation »

Ici, la spécification décrit explicitement 2 cas généraux :

- ▶ Mot avec nombre de lettres pair : “ours” (p)
- ▶ Mot avec nombre de lettres impair

On peut imaginer des cas limites :

- ▶ Mot vide
- ▶ Mot très long
- ▶ Mot avec un ou deux caractères

Principes et réalisation

Produire les tests à partir de la spécification

- ▶ « Couverture des fonctionnalités »
- ▶ « Couverture des cas d'utilisation »

Ici, la spécification décrit explicitement 2 **cas généraux** :

- ▶ Mot avec nombre de lettres pair : “ours” (p)
- ▶ Mot avec nombre de lettres impair : “poire” (i)

On peut imaginer des **cas limites** :

- ▶ Mot vide
- ▶ Mot très long
- ▶ Mot avec un ou deux caractères

Principes et réalisation

Produire les tests à partir de la spécification

- ▶ « Couverture des fonctionnalités »
- ▶ « Couverture des cas d'utilisation »

Ici, la spécification décrit explicitement 2 **cas généraux** :

- ▶ Mot avec nombre de lettres pair : “ours” (p)
- ▶ Mot avec nombre de lettres impair : “poire” (i)

On peut imaginer des **cas limites** :

- ▶ Mot vide : “” (p)
- ▶ Mot très long
- ▶ Mot avec un ou deux caractères

Principes et réalisation

Produire les tests à partir de la spécification

- ▶ « Couverture des fonctionnalités »
- ▶ « Couverture des cas d'utilisation »

Ici, la spécification décrit explicitement 2 cas généraux :

- ▶ Mot avec nombre de lettres pair : “ours” (p)
- ▶ Mot avec nombre de lettres impair : “poire” (i)

On peut imaginer des cas limites :

- ▶ Mot vide : “” (p)
- ▶ Mot très long : “anticonstitutionnellement” (i)
- ▶ Mot avec un ou deux caractères

Principes et réalisation

Produire les tests à partir de la spécification

- ▶ « Couverture des fonctionnalités »
- ▶ « Couverture des cas d'utilisation »

Ici, la spécification décrit explicitement 2 **cas généraux** :

- ▶ Mot avec nombre de lettres pair : “ours” (p)
- ▶ Mot avec nombre de lettres impair : “poire” (i)

On peut imaginer des **cas limites** :

- ▶ Mot vide : “” (p)
- ▶ Mot très long : “anticonstitutionnellement” (i)
- ▶ Mot avec un ou deux caractères : “kw” (p)

Mais encore (robustesse) ?

- ▶ Mot avec des majuscules :
 - « Essai » « essAI » « ESSAI » (i)
 - « AmsTramGramm » (p)

Inspiré du cours de Lydie du Bousquet, avec son accord.

Mais encore (robustesse) ?

- ▶ Mot avec des majuscules :
« Essai » « essAI » « ESSAI » (i)
« AmsTramGramm » (p)
- ▶ Mot avec des espaces :
« tout à fait » (i)

Inspiré du cours de Lydie du Bousquet, avec son accord.

Mais encore (robustesse) ?

- ▶ Mot avec des majuscules :
 - « Essai » « essAI » « ESSAI » (i)
 - « AmsTramGramm » (p)
- ▶ Mot avec des espaces :
 - « tout à fait » (i)
- ▶ Mot avec des caractères spéciaux
 - « test_d'intégration » (p)
 - « β -tests » (i)

Inspiré du cours de Lydie du Bousquet, avec son accord.

Mais encore (robustesse) ?

- ▶ Mot avec des majuscules :
« Essai » « essAI » « ESSAI » (i)
« AmsTramGramm » (p)
- ▶ Mot avec des espaces :
« tout à fait » (i)
- ▶ Mot avec des caractères spéciaux
« test_d'intégration » (p)
« β -tests » (i)
- ▶ Mot avec des chiffres
« test44 » (p)

Inspiré du cours de Lydie du Bousquet, avec son accord.

Autre possibilité

- ▶ Produire les tests à partir du code
- ▶ Graphe de flot de contrôle
 - ▶ Représentation intermédiaire du code
 - ▶ Nœuds et transitions représentent les exécutions possibles
- ▶ On s'assure que l'on est passé « partout » selon certains critères
 - ▶ Toutes les instructions exécutées
 - ▶ Toutes les branches
 - ▶ Toutes les conditions
 - ▶ Tous les chemins

Inspiré du cours de Lydie du Bousquet, avec son accord.

À propos de la couverture des tests

- ▶ Rappel : on teste pour découvrir des erreurs
- ▶ Tous les moyens sont bons pour générer les données
 - ▶ Spécification
 - ▶ Code
- ▶ Aucune méthode ne peut garantir l'absence d'erreur
- ▶ Stratégies complémentaires
- ▶ A propos de la couverture
 - ▶ Indicateur : le test n'est pas fini !
 - ▶ Ne garantit pas l'absence d'erreur

Inspiré du cours de Lydie du Bousquet, avec son accord.

Les tests et vous

Les types de tests

Quand tester ?

Que tester ?

Pour finir

CQFR : Ce Qu'il Faut Retenir



- ▶ les différents types de tests
- ▶ ce qu'il faut tester