

Les tests et les plans de tests

Table des matières

1. Tester vs Déboguer
2. Définition du test informatique
3. ISTQB : International Software Testing Qualifications Board
4. Les 7 principes généraux des tests
5. Quatre niveaux de test
6. Leur positionnement
7. Test amont/test aval
8. Pourquoi si peu de tests sont effectués

Table des matières

- 9. Pourquoi faire des tests
- 10. Classifications des tests
- 11. Les modes d'exécution des tests
- 12. Les modalités de test
- 13. Les méthodes de tests
- 14. Les exigences de qualité (iso 25010)
- 15. Exemples de types de tests
- 16. La gravité des anomalies

Table des matières

- 17. L'analyse du risque
- 18. Comment estimer un projet de tests
- 19. Les activités liées aux tests
- 20. Plan Qualité Projet (PQP)
- 21. Plan de tests

1. Tester vs Déboguer

■ **L'erreur en programmation est normale**

■ **Tester vs Déboguer**

➤ **Tester et déboguer, ce n'est pas la même activité et les responsables sont différents:**

✓ **Les testeurs testent**

✓ **Les développeurs déboguent (Déboguer est une activité de développement permettant d'analyser, trouver et supprimer les causes de la défaillance)**

2. Définition du test informatique

- En informatique, un test désigne une procédure de vérification partielle d'un système.
- Son objectif principal est d'identifier un nombre maximum de comportements problématiques du logiciel.
- Il permet ainsi, dès lors que les problèmes identifiés seront corrigés, d'en augmenter la qualité
- [https://fr.wikipedia.org/wiki/Test \(informatique\)](https://fr.wikipedia.org/wiki/Test_(informatique))

3. ISTQB : International Software Testing Qualifications Board

- L'ISTQB est le Comité international de qualification du test logiciel.
- Cette organisation propose une certification reconnue dans le monde entier.
- Elle a été fondée en novembre 2002 à Édimbourg, comme association à but non lucratif, et est légalement enregistrée en Belgique.
- Sites : <https://fr.wikipedia.org/wiki/ISTQB>
<https://www.istqb.org/>

3. ISTQB

- L'ISTQB propose de valider le titre de testeur certifié ISTQB, une qualification standardisée pour le test logiciel.
- Les qualifications validées sont hiérarchisées et suivent des directives d'accréditation et d'examen.
- A ce jour, plus de 500 000 certificats ISTQB ont été délivrés dans le monde et plus de 10 000 en France (cf. RNCP).
- Le comité de l'ISTQB est composé de membres représentant plus de 71 pays.
- L'ISTQB propose trois niveaux de certification : le niveau fondamental, le niveau avancé, le niveau expert

4. Les 7 principes généraux des tests

- Principe 1 – Les tests montrent la **présence de défauts**. Les tests peuvent prouver la présence de défauts, mais **ne peuvent pas en prouver l'absence**. Les tests réduisent la probabilité que des défauts restent cachés dans le logiciel mais, même si aucun défaut n'est découvert, ce n'est pas une preuve d'exactitude.
- Principe 2 – **Les tests exhaustifs sont impossibles**. Tout tester (toutes les combinaisons d'entrées et de préconditions) n'est pas faisable sauf pour des cas triviaux. Plutôt que des tests exhaustifs, nous utilisons **l'analyse des risques et des priorités** pour focaliser les efforts de tests.

4. Les 7 principes généraux des tests

- Principe 3 – **Tester tôt**. Pour trouver des défauts tôt, les activités de tests devraient commencer aussi tôt que possible dans le cycle de développement du logiciel ou du système, et devraient être focalisées vers des objectifs définis.
- Principe 4 – **Regroupement des défauts**. L'effort de test devrait être fixé proportionnellement à la densité des défauts prévus et constatés dans les différents modules. **Un petit nombre de modules contiennent généralement la majorité des défauts détectés** lors des tests pré- livraison, ou affichent le plus de défaillances en opération

4. Les 7 principes généraux des tests

- Principe 5 – **Paradoxe du pesticide**. Si les mêmes tests sont répétés de nombreuses fois, il arrivera que **le même ensemble de cas de tests ne trouvera plus de nouveaux défauts**. Pour prévenir ce “paradoxe du pesticide”, **les cas de tests doivent être régulièrement revus et révisés**, et de nouveaux tests, différents, doivent être écrits pour couvrir d’autres chemins dans le logiciel ou le système de façon à permettre la découverte de nouveaux défauts

4. Les 7 principes généraux des tests

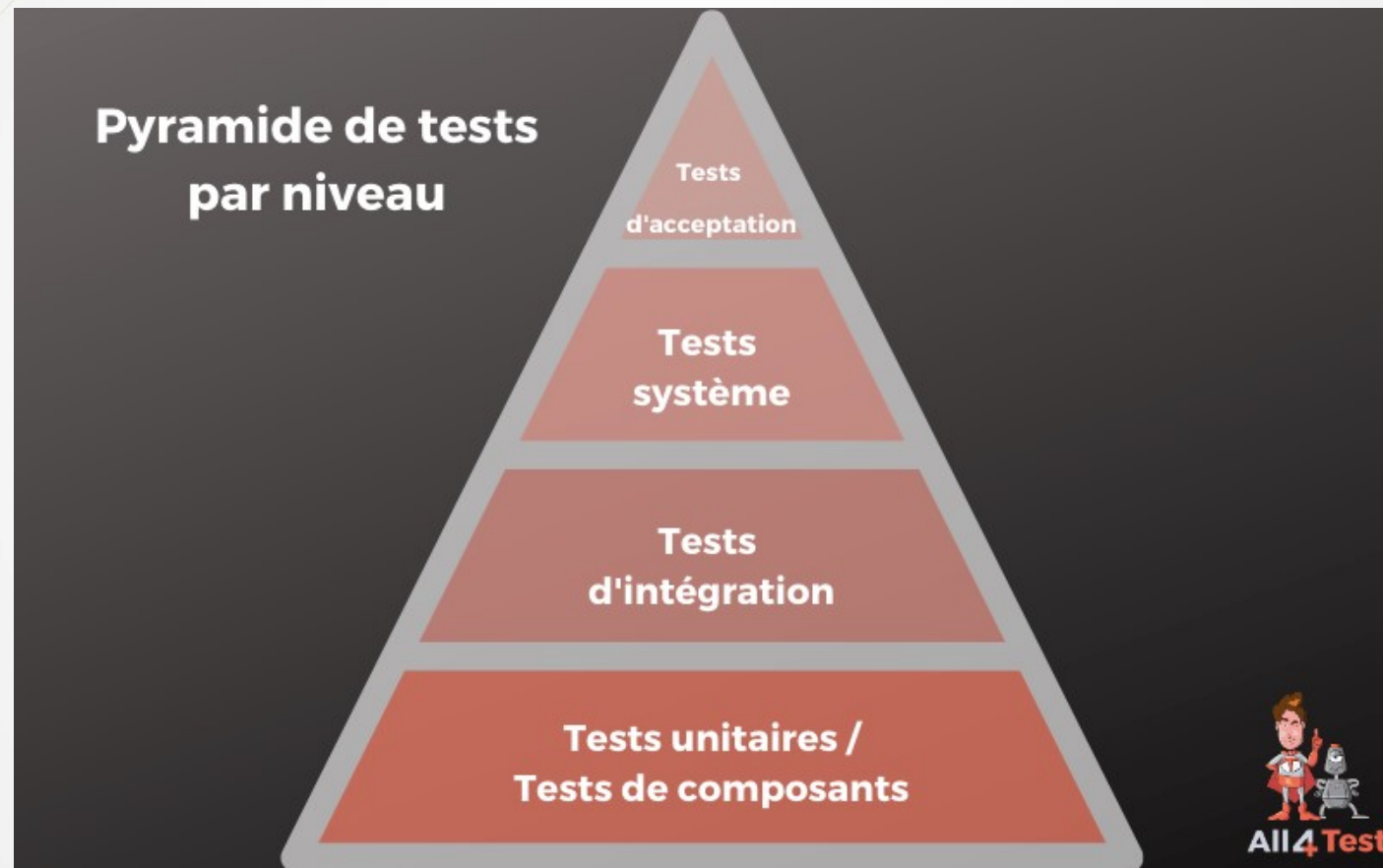
- Principe 6 – **Les tests dépendent du contexte. Les tests sont effectués différemment dans des contextes différents.** Par exemple, les logiciels de sécurité critique seront testés différemment d'un site de commerce électronique.
- Principe 7 – **L'illusion de l'absence d'erreurs.** Trouver et corriger des défauts n'aide pas si le système conçu est inutilisable et ne comble pas les besoins et les attentes des utilisateurs

5. Quatre niveaux de test

- Le CFTL, Comité Français du Test Logiciel (<https://www.cftl.fr>) identifie quatre **niveaux** de test :
 1. Tests unitaires
 2. Tests d'intégration
 3. Tests système
 4. Tests d'acceptation

[Les différents types de test d'un logiciel - Guide complet \(cloudnetcare.fr\)](#)

5. Quatre niveaux de test



5. Quatre niveaux de test

1. **Tests unitaires** (https://fr.wikipedia.org/wiki/Test_unitaire) (ou test de composants).

- Ils doivent pouvoir être refait si les résultats ne sont pas conformes à ceux attendus
- Ils permettent de tester un composant, ou un bout de code, isolé de ses dépendances.
- Ils doivent permettre une couverture de code importante
- Leur jeu de tests : les cas généraux et tous les cas particuliers.
- Il doivent être en échec si un seul cas est non conforme

5. Quatre niveaux de test

2. **Tests d'intégration** (anciennement test technique ou test d'intégration technique)

- Ils permettent de vérifier que des bouts de code isolés fonctionnent bien ensemble. Mais on est encore sur des fonctionnalités, pas l'application complète
- ✓ Exemple : Dans un blog, suite à la saisie d'un formulaire de commentaire, est-ce que le commentaire a bien été créé dans la base de données

5. Quatre niveaux de test

3. **Tests système** (anciennement test fonctionnel ou test d'intégration fonctionnel ou homologation).

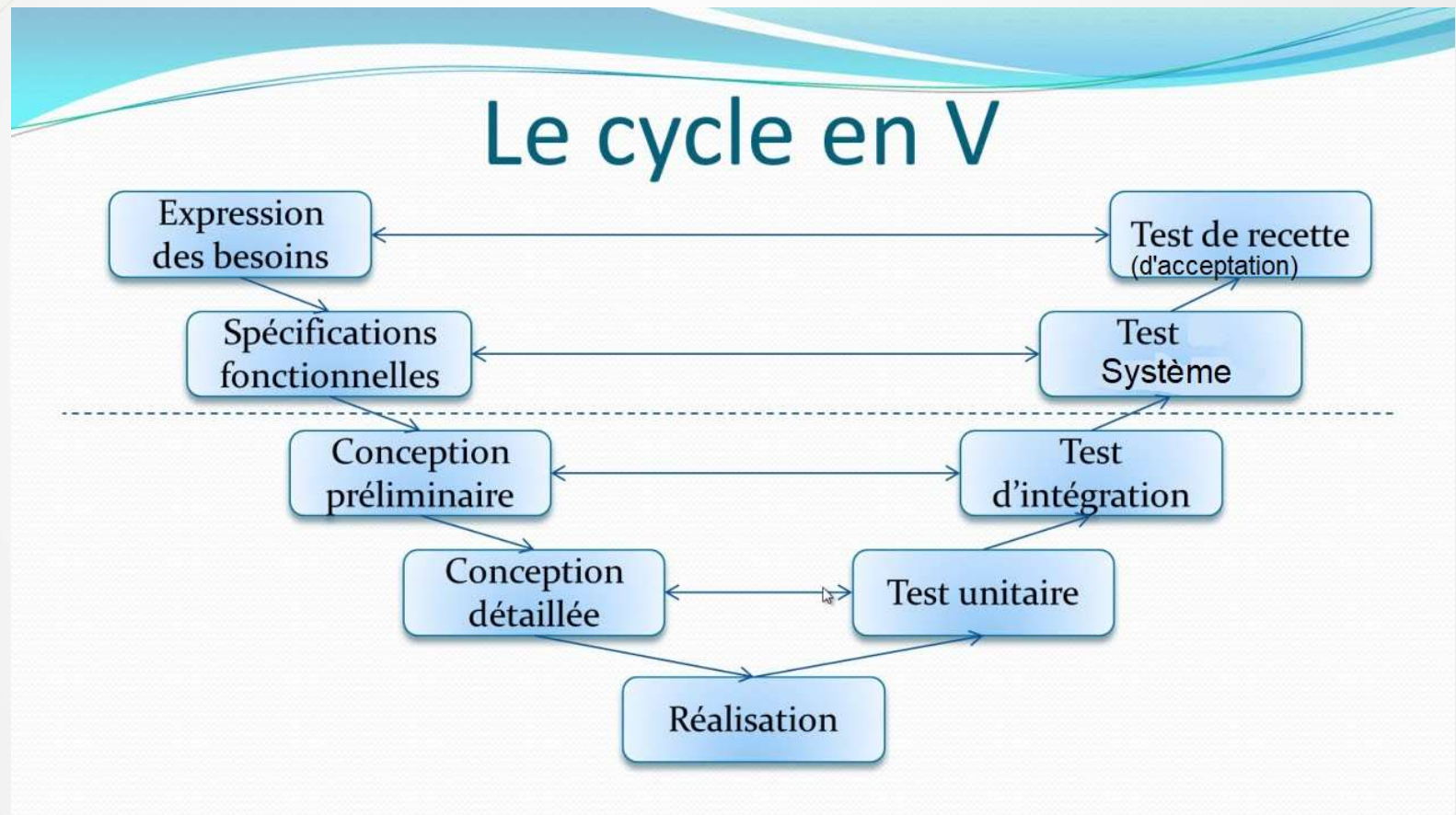
- Ils permettent de tester l'application tout entière, avec tous les serveurs concernés, comme les serveurs de BDD
- Ils peuvent utiliser les tests d'intégration, mais avec des serveurs ayant une configuration proche de celle de l'environnement de production

5. Quatre niveaux de test

4. **Tests d'acceptation** (anciennement test usine ou recette).

- Ils permettent de s'assurer que l'application répond bien au besoin fonctionnel. Ils sont **manuels** et ils sont effectués par la MOA sur l'environnement de préproduction (UAT : User Acceptance Test)

6. Leur positionnement



7. Test amont/test aval

- Par ailleurs, on parle aussi de niveau de *test amont* pour désigner les niveaux unitaire et intégration, et de niveau de *test aval* pour désigner les niveaux système et acceptation.
- De même, en France, le terme « phase de test » est parfois utilisé à la place de « niveau de test », mais il ne respecte pas le vocabulaire du CFTL.

8. Pourquoi si peu de tests sont effectués

- Pas motivant
- Prend du temps
- Impression de perdre son temps

9. Pourquoi faut t'il faire des tests

- Robustesse de l'application
- Intégrité des données
- Image de marque de l'entreprise
- Eviter les reprises et leurs coût et stress (en cas de problème en production)

10. Classifications des tests

- Leur mode d'exécution
- Leurs modalités
- Leurs méthodes
- Les exigences de qualité (iso 25019)
- Leur gravité

11. Les modes d'exécution des tests

- Manuels

- ✓ Exemple : test d'une UI

- Automatisés

- ✓ Exemple : tests unitaires avec un logiciel d'automatisation de tests unitaires

12. Les modalités de test

- Tests statiques : avant l'exécution
 - Revue de code (simple et croisée sur tout le code ou inspection d'une partie du code précise)
- Tests dynamiques : nécessite l'exécution du code binaire

13. Les méthodes de tests

► Structurel (boîte blanche)

une boîte blanche (de l'anglais white box), ou boîte transparente, est un module d'un système dont on peut prévoir le fonctionnement interne car **on connaît les caractéristiques de fonctionnement de l'ensemble des éléments qui le composent.**

13. Les méthodes de tests

► Fonctionnels (boîte noire)

Le test de la boîte noire, ou test de la boîte opaque, est utilisé en programmation informatique et en génie logiciel pour **tester un programme en vérifiant que les sorties obtenues sont bien celles prévues pour des entrées données.**

L'expression « boîte noire », ou « boîte opaque », vient du fait que les composants et **les processus du dispositif de traitement ne sont pas visibles** ou transparents et que le programme testé n'est pas étudié.