FONDAMENTAUX DES TESTS

Test des logiciels

certification ISTQB

Sarra MEJRI:

- Docteur en informatique
- sarra.mejri.isim@gmail.com

CHAPITRE 1 : FONDAMENTAUX DES TESTS

180 minutes

Objectifs :

- Le candidat apprend les principes de base sur les tests, les raisons pour lesquelles les tests sont requis et quels sont les objectifs des tests.
- Le candidat comprend le processus de test, les principales activités de test et le testware.

o Le candidat comprend les compétences essentielles pour les tests

1.1 Qu'est-ce que le test ?

FL-1.1.1 (K1) Identifier les objectifs habituels du test

FL-1.1.2 (K2) Faire la différence entre tester et déboguer

1.2 Pourquoi est-il nécessaire de tester?

FL-1.2.1 (K2) Donner des exemples montrant la nécessité des tests

FL-1.2.2 (K1) Rappeler la relation entre les tests et assurance qualité

FL-1.2.3 (K2) Faire la distinction entre la cause racine, l'erreur, le défaut et la défaillance.

1.3 Principes du test

FL-1.3.1 (K2) Expliquer les sept principes du test

1.4 Activités de test, testware et rôles dans le test

FL-1.4.1 (K2) Résumer les différentes activités et tâches de test

FL-1.4.2 (K2) Expliquer l'impact du contexte sur le processus de test

FL-1.4.3 (K2) Différencier les composants du testware qui soutiennent les activités de test

FL-1.4.4 (K2) Expliquer la valeur du maintien de la traçabilité

FL-1.4.5 (K2) Comparer les différents rôles dans le test

1.5 Compétences essentielles et bonnes pratiques en matière de test

FL-1.5.1 (K2) Donnez des exemples de compétences génériques requises pour le test

FL-1.5.2 (K1) Rappeler les avantages de l'approche équipe intégrée

FL-1.5.3 (K2) Distinguer les avantages et les inconvénients de l'indépendance du test

PLAN

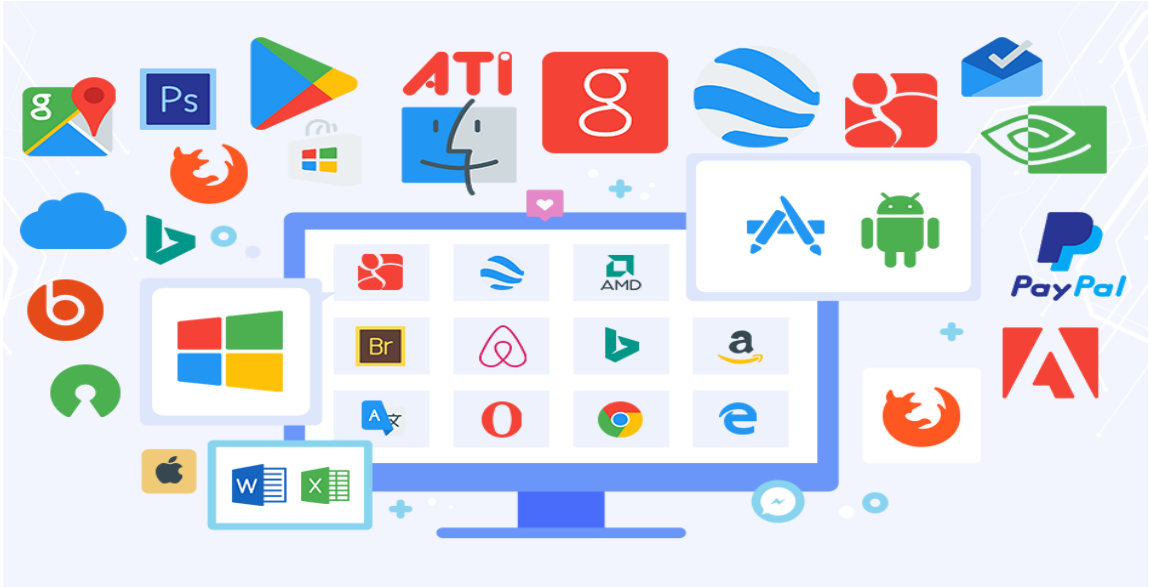
1. Définition du test logiciel
2. Terminologie
3. 7 principes généraux des tests
4. Processus de test
5. Psychologie des tests

DÉFINITION DES TESTS

Que sont les tests ?

QU'EST CE QU'UN LOGICIEL ?

Logiciel : Programmes informatiques, procédures et, éventuellement, documentation et données connexes se rapportant au fonctionnement d'un système informatique.



Les systèmes logiciels font partie intégrante de la vie quotidienne, depuis les applications commerciales (p. ex., les services bancaires) jusqu'aux produits pour les consommateurs (p. ex., voitures).

MON LOGICIEL NE FONCTIONNE PAS CONVENABLEMENT !!

La plupart des gens ont eu une expérience avec un logiciel qui n'a pas fonctionné comme prévu.

- Un logiciel **qui ne fonctionne pas correctement** peut entraîner de nombreux problèmes, y compris :
  - La perte d'argent,
  - La perte de temps
  - entacher la réputation de l'entreprise
  - entraîner des blessures qui peuvent être mortelles.

actualite

Samsung: après les téléphones qui explosent, les lave-linge qui se désintègrent

Samsung avait été contraint de rappeler 2,5 millions de ses Galaxy Note 7, il doit maintenant en faire de même avec 2,8 millions de machines à laver. Certains de leurs éléments se détacheraient lorsqu'elles sont en marche.

Logiciel défaillant : Škoda suspend les livraisons d'Octavia

21/05/2020

SUIVEZ NOUS

En raison d'un problème dans le fonctionnement du logiciel eCall, la marque Škoda Auto a suspendu la livraison de ses modèles Octavia. Le système, qui doit prévenir les secours en cas d'accident, est défaillant.

"Nous analysons tous les détails et travaillons à une solution. Les prochaines étapes seront annoncées dès que possible", a déclaré le porte-parole de Škoda, Pavel Jína.

D'autres modèles de marques appartenant au groupe Volkswagen sont concernés par ce problème de logiciel, dont la Golf 8. Une mise à jour serait en cours de développement.

Auteur: Alexis Rosenzwein

Un homme de 28 ans décède après l'explosion de ses écouteurs

Par **David Manfredini** - 12 août 2021

Il préparait ses examens en écoutant de la musique avec ses écouteurs.

Un homme de 28 ans est décédé au Rajasthan en Inde dans sa maison à la périphérie de la capitale Jaipur après l'explosion de ses écouteurs vendredi dernier. Selon le journal Times of India, Rakesh Kumar Nagar, l'homme en question, avait branché ses écouteurs sur son téléphone portable qui se chargeait.

Le jeune homme se préparait aux tests de recrutement du gouvernement avec des cours en ligne et passait la plupart de son temps à étudier dans une petite pièce située dans la ferme de sa famille. Selon la police, Rakesh s'est marié en février et était l'aîné de la fratrie.

Android : plusieurs applications Google victimes d'un important bug

Par **Jennifer Mertens** - 23 mars 2021

Un correctif est désormais disponible depuis le Google Play Store.

Durant plusieurs heures, les applications mobiles de Google ont été victimes d'un important dysfonctionnement. De nombreux utilisateurs Android ont en effet rapporté avoir rencontré des problèmes avec Chrome et Gmail, notamment. Mais le souci semble également concerner plusieurs autres applications mobiles.

Les utilisateurs affectés par le problème rapportent avoir reçu une notification leur indiquant que leurs applications ouvertes avaient planté. Malheureusement, après ce message d'erreur, il leur était impossible de relancer les applications concernées. Les

Windows 10 et KB4601319, des problèmes apparaissent

Auteur : Jérôme Giardi Dans Windows 10 09/03/2021 13 commentaires

L'une des dernières mises à jour pour Windows 10 est à l'origine de plusieurs problèmes. L'historique des fichiers rencontre des soucis et la reconnaissance des Webcams ne fonctionne pas correctement. Dans d'autres cas il s'agit d'applications bloquées ou d'installation impossible.

L'historique des fichiers est une solution de sauvegarde native de Windows 10. Elle permet de sauvegarder des fichiers sur un autre lecteur. Il est alors possible de les restaurer si les originaux sont supprimés ou endommagés. Le lecteur peut être une clé USB, un SSD externe ou encore une seconde unité de stockage [illegible]PC ou un emplacement réseau.


SOLUTION


SOFTWARE
TESTING

LE TEST LOGICIEL

✗	✓
Le test se concentre entièrement sur la vérification de l'objet de test, c'est-à-dire le contrôle de la conformité du système aux exigences spécifiées (spécification).	Si le test implique la vérification, il implique également la validation, c'est-à-dire le contrôle de la conformité du système aux besoins des utilisateurs et des autres parties prenantes dans son environnement opérationnel.
Le test consiste uniquement à exécuter des tests (faire fonctionner le logiciel et à vérifier les résultats des tests)	le test de logiciels comprend également d'autres activités et doit être aligné sur le cycle de vie du développement logiciel
Le test est seulement une activité technique.	Le test doit également être correctement planifié, géré, estimé, piloté et contrôlé

TEST LOGICIEL

dynamique VS statique

Le test peut être dynamique ou statique.

- Le test dynamique :
  - implique l'exécution du logiciel
  - utilise différents types de techniques de test et d'approches de test pour dériver des cas de test
- Le test statique consiste à effectuer des tests sans exécuter l'objet de test. Il comprend les revues et l'analyse statique

LE TEST LOGICIEL

Le test logiciel est une méthodologie visant à évaluer **la qualité d'un logiciel** ainsi que les propriétés des artefacts qui le composent. Chaque artefact testé est désigné sous le terme d'**objet de test**. Cet ensemble d'activités vise à faciliter la détection des **défauts**, contribuant ainsi à réduire le risque de **défaillance** du logiciel lors de son fonctionnement.

Le test logiciel est essentiel pour le succès du projet.

OBJECTIFS DU TEST (1)

- Évaluer les produits d’activités tels que les exigences, les User Stories, la conception et le code
- Provoquer des défaillances et trouver des défauts.
- Assurer la couverture requise d'un objet de test.
- Réduire le niveau de risque d'une qualité logicielle insuffisante.
- Vérifier si les exigences spécifiées ont été satisfaites.
- Vérifier qu'un objet de test est conforme aux exigences contractuelles, légales et réglementaires

OBJECTIFS DU TEST (2)

- Fournir des informations aux parties prenantes pour leur permettre de prendre des décisions éclairées.
- Construire la confiance dans la qualité de l'objet de test.
- Valider si l'objet de test est complet et fonctionne comme attendu par les parties prenantes

OBJECTIFS DU TEST (3)

Selon le contexte

Les objectifs du test peuvent varier en fonction du contexte :

- le produit d'activités testé,
- le niveau de test, les risques
- le cycle de vie du développement logiciel (SDLC)
- les facteurs liés au contexte métier :
  - la structure de l'entreprise,
  - les considérations concurrentielles ou
  - le délai de mise sur le marché.

TERMINOLOGIE

Test et qualité

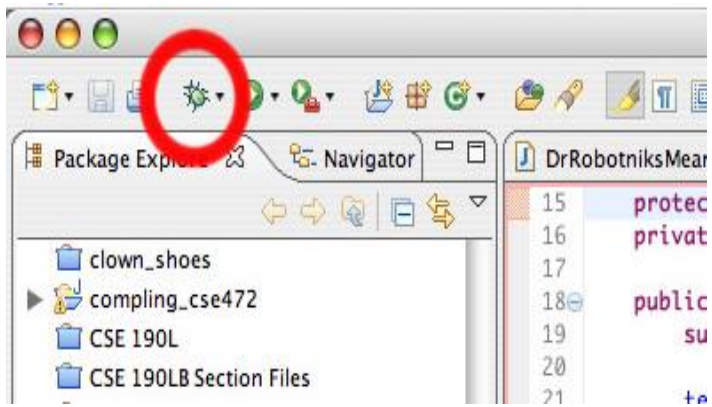
Erreur, défaut, défaillance

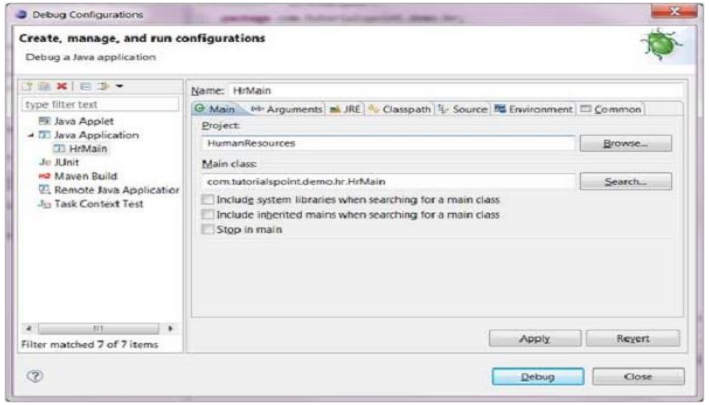
Cause racine, effet

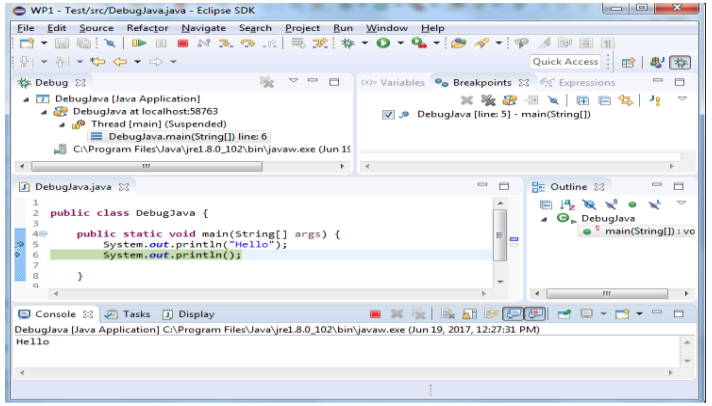
TEST ET DÉBOGAGE

Les activités de tests et de débogage sont différentes.

Le débogage est le processus **qui trouve, analyse et corrige de tels défauts**.







TEST ET DÉBOGAGE

Test dynamique

- L'exécution de tests peut mettre en évidence des **défaillances** causées par des **défauts** dans le logiciel
- Le débogage consiste à trouver les causes de cette défaillance (défauts), à analyser ces causes et à les éliminer

Processus du débogage

Test statique

- Lorsque le test statique identifie un défaut, le débogage consiste à l'éliminer.
- Il n'est pas nécessaire de le reproduire ou de le diagnostiquer, puisque le test statique constate directement les défauts et ne peut pas provoquer de défaillances. .

TEST ET ASSURANCE QUALITE

Assurance qualité, contrôle qualité et gestion de la qualité

- **L'assurance qualité** est la définition des processus et des normes qui devraient conduire à des produits de haute qualité et la mise en place de processus de qualité dans le processus de fabrication.
- Le **contrôle de qualité** est l'application de ces processus de qualité pour éliminer les produits qui ne sont pas du niveau de qualité requis.
- La gestion de la qualité :
  - est un concept plus large.
  - comprend toutes les activités qui dirigent et contrôlent une organisation en matière de qualité.
  - comprend à la fois l'assurance de la qualité et le contrôle de la qualité.

TEST ET ASSURANCE QUALITE

- .La **qualité du logiciel** n'est pas seulement de savoir si les **fonctionnalités** du logiciel ont été correctement mises en œuvre, mais elle dépend aussi des attributs **non fonctionnels** du système
- Le contrôle de la qualité implique les séries d'inspections, de révisions et de tests utilisés tout au long du processus de développement du logiciel afin de s'assurer que chaque composante rencontre les spécifications définies dans l'analyse.

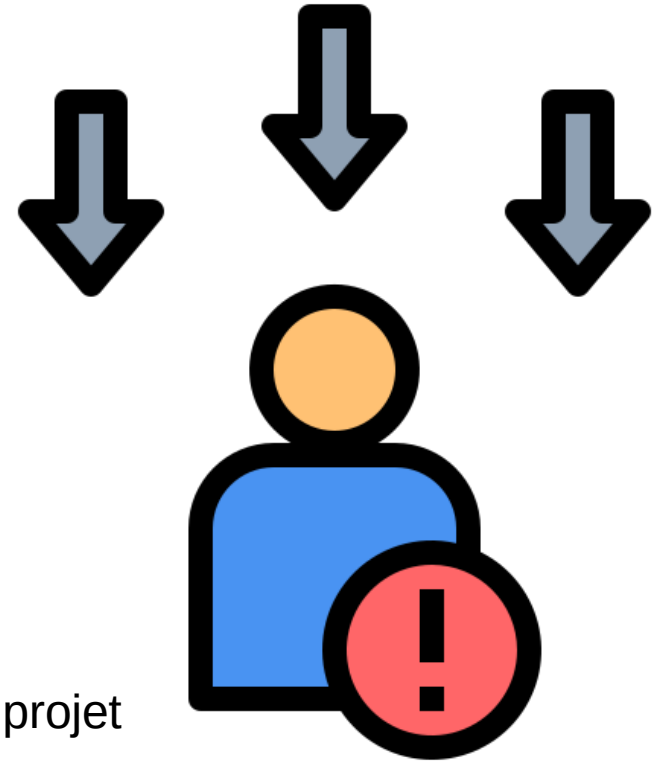
L'assurance qualité et le test ne sont pas les mêmes, mais ils sont liés
Le test est une forme de contrôle de la qualité.

ERREUR, DÉFAUT ET DÉFAILLANCE

- Une personne peut faire **une erreur**, ce qui peut conduire à **l'introduction d'un défaut (faute ou bogue)** dans le code du logiciel ou dans un autre produit d’activités connexe.
    - Par exemple, une erreur d'élucidation d'exigences peut conduire à un défaut au niveau des exigences, qui se traduit ensuite par une erreur de programmation qui conduit à un défaut dans le code.
- Si un défaut dans le code est exécuté, cela peut causer une défaillance, mais pas nécessairement dans toutes les circonstances.
    - Par exemple, certains défauts nécessitent des valeurs de données d’entrée ou des conditions préalables très spécifiques pour déclencher une défaillance, ce qui peut se produire rarement ou jamais.

ERREUR, DÉFAUT ET DÉFAILLANCE

Erreur : origine et raisons



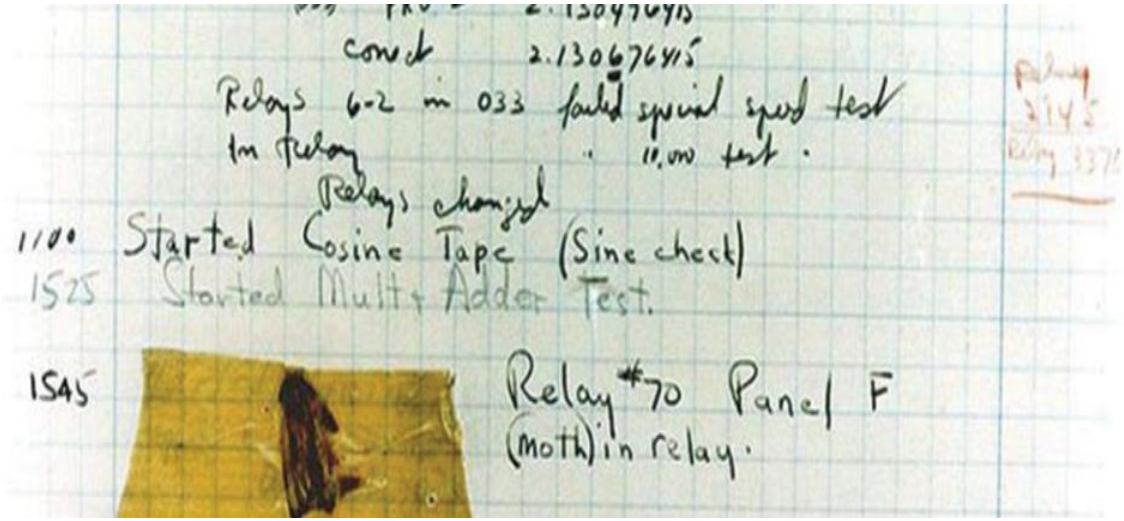
Une erreur fait toujours référence à une erreur humaine.

Les erreurs peuvent survenir pour de nombreuses raisons :

- Pression temporelle
- Faillibilité humaine
- Manque d'expérience ou compétences insuffisantes des membres de l'équipe projet
- Problèmes d'échange d'informations entre les parties prenantes
- Ambiguïtés concernant la compréhension des exigences et de la documentation du projet
- Complexité du code, de la conception, de l'architecture, du problème à résoudre et/ou de la technologie utilisée
- Malentendus concernant les interfaces au sein et entre les systèmes, surtout lorsqu'il y en a un grand nombre
- Utilisation de nouvelles technologies peu familières

ERREUR, DÉFAUT ET DÉFAILLANCE

Défaut



Le premier bug de l'histoire a été découvert en 1947. À l'Université Harvard à Cambridge, Massachusetts, des chercheurs ont constaté que leur ordinateur, le Mark II, rencontrait constamment des dysfonctionnements. En examinant le matériel informatique, ils ont fait une découverte inattendue : un papillon de nuit s'était coincé à l'intérieur. L'insecte avait perturbé l'électronique de l'ordinateur.

Défaillance

Les défaillances peuvent être causées non seulement par des erreurs humaines mais aussi par des facteurs environnementaux, tels que : Radiation, Champ électromagnétique, contamination physique (poussière, humidité), contamination des données (les attaques de logiciels malveillants)….

ERREUR, DÉFAUT ET DÉFAILLANCE

Exemples

Il peut y avoir un défaut dans le programme. L'exécution d'un morceau de code où il y a un défaut peut ou non entraîner une défaillance.

```
AverageValue := SumOfValues / NumberOfMeasurements
```

Le programme comporte un défaut car il ne vérifie pas si le dénominateur de la fraction en cours de calcul est égal à zéro

Si NumberOfMeasurements est toujours différent de 0, l'exécution de l'instruction ne posera pas un problème -→ pas de défaillance

Si NumberOfMeasurements est égal à 0, l'exécution de l'instruction posera un problème → défaillance

ERREUR, DÉFAUT ET DÉFAILLANCE

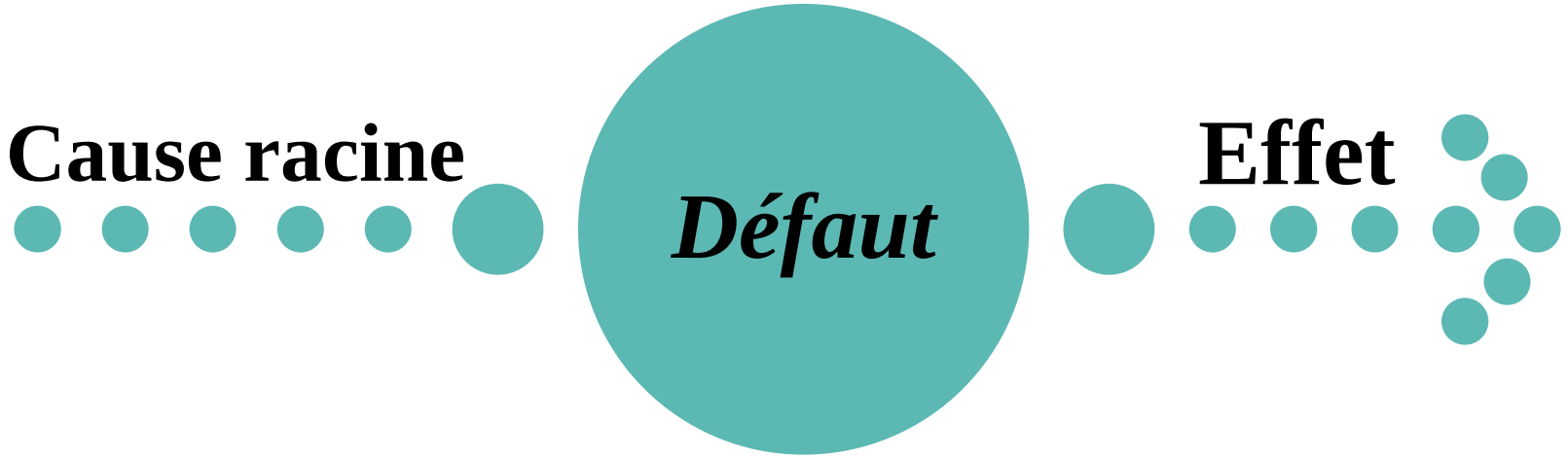
Est commise par une ressource participant à la création de logiciel

Est le résultat d'une erreur

Est un comportement différent du comportement attendu

DÉFAUT, CAUSE RACINE ET EFFET

- Les causes racines des défauts sont les premières actions ou conditions qui ont contribué à la création des défauts.
- Les défauts peuvent être analysés pour identifier leurs causes racines, afin de réduire l'apparition de défauts similaires à l'avenir.



DÉFAUT, CAUSE RACINE ET EFFET

Exemple :

Supposons que des paiements d'intérêts incorrects, dus à une seule ligne de code incorrecte, se traduisent par les plaintes des clients. Le code défectueux a été écrit pour une User Story qui était ambiguë, en raison de la mauvaise compréhension par le Product Owner de la façon de calculer les intérêts. S'il existe un pourcentage élevé de défauts dans les calculs d'intérêt, et que ces défauts ont leur cause racine dans des incompréhensions similaires, le Product Owner pourrait se former au calcul des intérêts afin de réduire le nombre de tels défauts à l'avenir.

Question : Déterminer l'erreur, le défaut, la défaillance, la cause racine, l'effet

- **La cause racine** du défaut initial était un manque de connaissance de la part du Product Owner, ce qui l'a conduit à faire **une erreur** lors de la rédaction de la User Story.
- Les plaintes des clients sont **des effets**.
- Les paiements d'intérêts incorrects sont des **défaillances**.
- Le calcul incorrect dans le code est **un défaut**, et il résulte du défaut d'origine, l'ambiguïté dans la User Story.

LES 7 PRINCIPES DES TESTS

7 PRINCIPES DE TEST

Principe 1 : Le test montre la présence, et non l'absence, de défauts.

Principe 2 : Le test exhaustif est impossible.

Principe 3 : Tester tôt économise du temps et de l’argent

Principe 4 : Regroupement des défauts

Principe 5 : Usure des tests

Principe 6 : Le test dépend du contexte

Principe 7 : L'illusion de l'absence de défaut.

Principe 1: Le test montre la présence, et non l'absence, de défauts.

- Le test peut montrer que des défauts sont présents dans l'objet de test, mais ne peut pas prouver qu'il n'y a pas de défauts
- Le test réduit la probabilité que des défauts restent cachés dans le logiciel mais, même si aucun défaut n’est découvert, ce n’est pas une preuve que l’objet de test est correct.

Même si le test ne trouve aucun défaut, ce n’est pas une preuve que le logiciel est sans défaut.

Principe 1 : Le test montre la présence, et non l'absence, de défauts.

Ce principe a des implications sérieuses :

- les tests sont de nature négative, c'est-à-dire qu'ils montrent que quelque chose ne fonctionne pas, et non pas que tout va bien.
- Cela entraîne des implications psychologiques importantes.

PRINCIPE 2 : LE TEST EXHAUSTIF EST IMPOSSIBLE

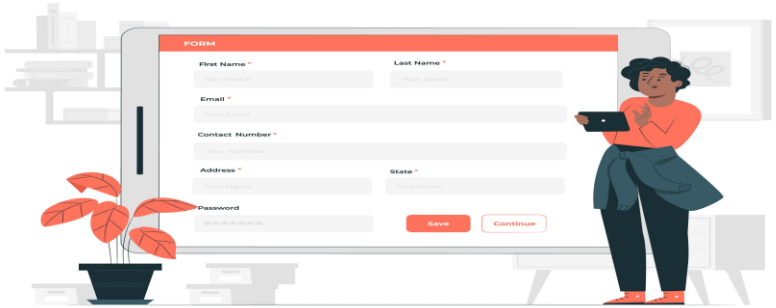
- Test exhaustif - une approche de test dans laquelle toutes les combinaisons de données possibles sont utilisées. Cela inclut les combinaisons de données implicites présentes dans l'état du logiciel/des données au début des tests.
- Tout tester (toutes les combinaisons d’entrées et de préconditions) n’est pas faisable sauf pour des cas triviaux.
- Plutôt que de chercher à faire tests exhaustifs, l’analyse des risques, des techniques de test et des priorités devraient être utilisés pour cibler les efforts de test.

PRINCIPE 2 : LE TEST EXHAUSTIF EST IMPOSSIBLE

Exemple 1

En considérant un petit logiciel où l'on peut entrer un mot de passe (spécifié pour contenir jusqu'à trois caractères)

Quel est le nombre de combinaisons possibles ?



- ✓ En utilisant uniquement des lettres majuscules alphabétiques et en complétant les trois caractères, il existe 26 × 26 × 26 permutations d'entrée.
- ✓ Avec un clavier standard, il n'y a pas 26 × 26 × 26 permutations, mais un nombre beaucoup plus élevé allant jusqu’à 256 × 256 × 256

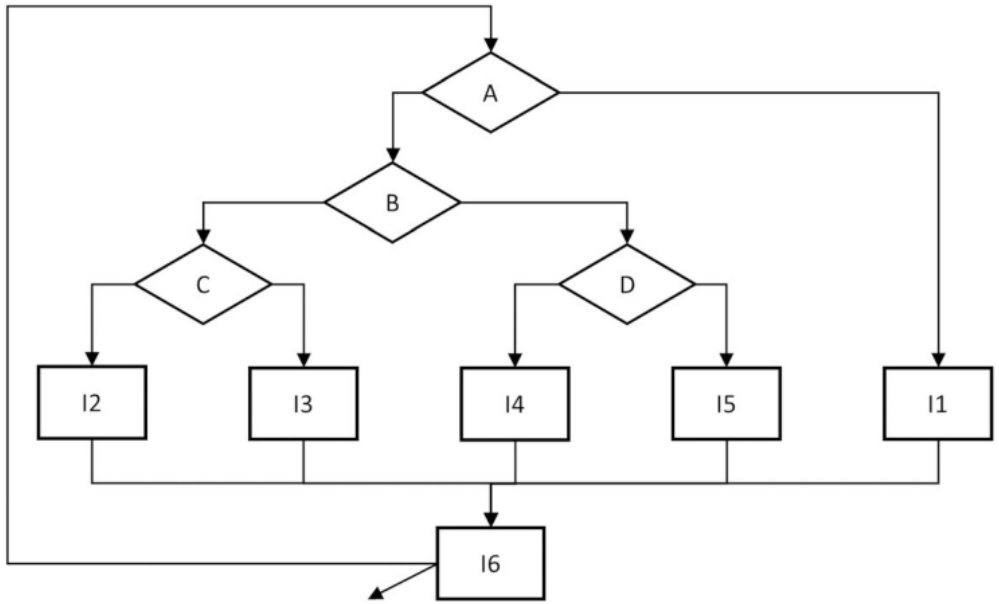
PRINCIPE 2 : LE TEST EXHAUSTIF EST IMPOSSIBLE

Exemple 2

Considérez un morceau de code à tester avec quatre décisions (A, B, C, D) et six instructions (I1, I2, I3, I4, I5, I6).

P1: A I1 I6
P2: !A B D I5 I6
P3: !A B !D I4 I6
P4: !A !B C I3 I6
P5: !A !B !C I2 I6

flux de contrôle du fragment de code à tester



Deux itérations de boucle peuvent réaliser $5^{2=}25$ chemins possibles

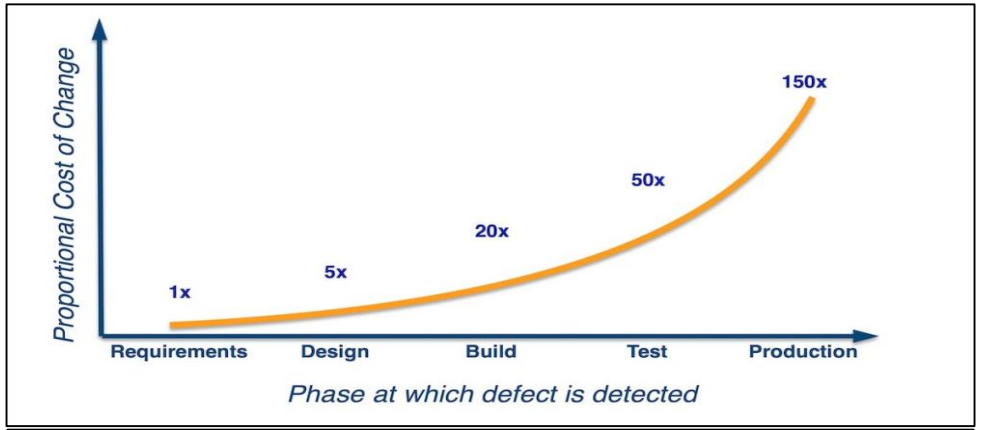
P1 P1; P1 P2; P1 P3; P1 P4; P1 P5;
P2 P1; P2 P2; P2 P3; P2 P4; P2 P5;
P3 P1; P3 P2; P3 P3; P3 P4; P3 P5;
P4 P1; P4 P2; P4 P3; P4 P4; P4 P5;
P5 P1; P5 P2; P5 P3; P5 P4; P5 P5.

Trois itérations de boucle donnent $5^3 = 125$ chemins possibles
n itérations de boucle donnent 5^n chemins possibles

Si nous supposons que nous avons besoin seulement de 0,001 s pour tester un chemin, il nous faudrait environ 3 ans pour tester tous les chemins pour le cas de 20 itérations . Malheureusement, nous n'avons pas autant de temps !

PRINCIPE 3: TESTER TÔT ÉCONOMISE DU TEMPS ET DE L'ARGENT

- Pour détecter tôt les défauts, Pour trouver les défauts le plus tôt possible, les tests statiques et les tests dynamiques doivent être lancés le plus tôt possible.
- Tester tôt dans le cycle de vie du développement logiciel permet de réduire ou d'éliminer des changements coûteux.



Courbe de Boehm, 1984

La courbe est exponentielle, ce qui signifie que plus tard un défaut est découvert, plus le coût de sa réparation augmente

PRINCIPE 4: REGROUPEMENT DES DÉFAUTS

- Un petit nombre de modules contient généralement la plupart des défauts découverts lors des tests avant livraison, ou est responsable de la plupart des défaillances en exploitation.
- Des regroupements prévisibles de défauts, ou des regroupements réellement observés en test ou en exploitation, constituent un élément important de l'analyse des risques utilisée pour cibler l'effort de test.

PRINCIPE 4: REGROUPEMENT DES DÉFAUTS

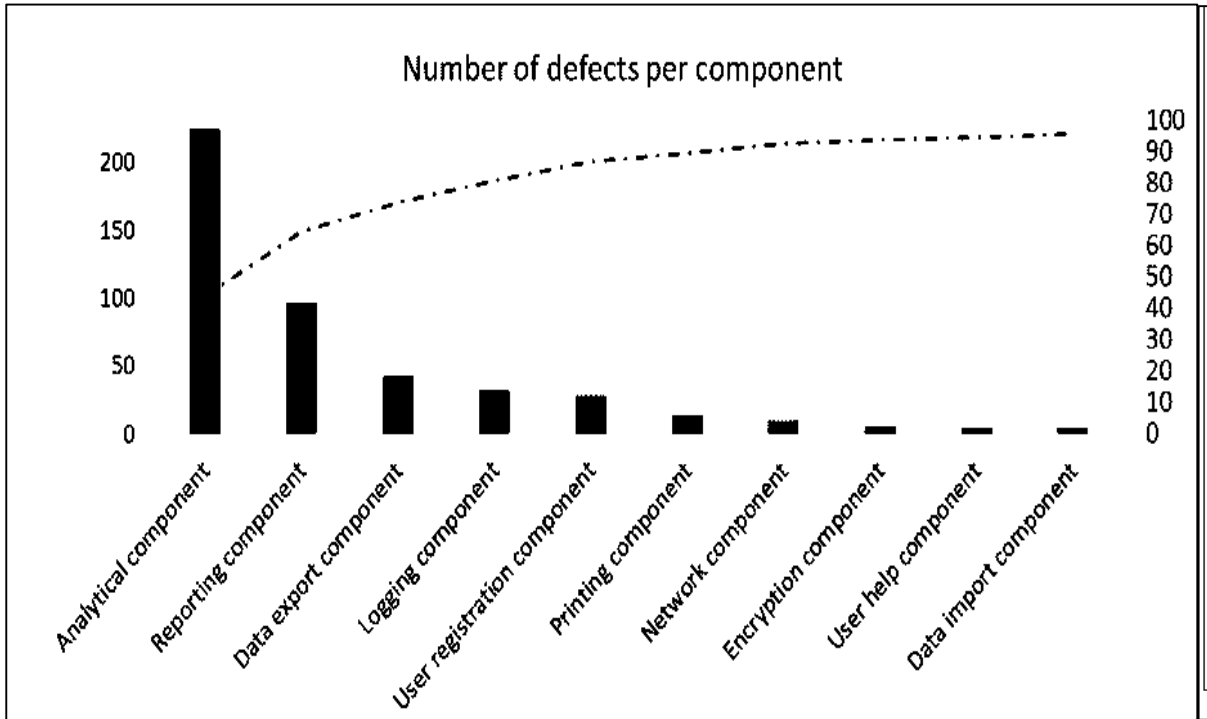
Loi de Pareto

- Ce phénomène est une illustration du principe de Pareto
- C'est l'application du principe de Pareto aux tests de logiciels: **environ 80 % des problèmes se retrouvent dans environ 20 % des modules.**
- Il est utile que l'activité de test cible ces modules de l'application où une forte proportion de défauts peut être trouvée.
- Cependant, il faut se rappeler que **les tests ne se concentrent pas exclusivement sur ces modules**. Il peut y avoir moins de défauts dans le code restant, mais les testeurs doivent toujours les rechercher.

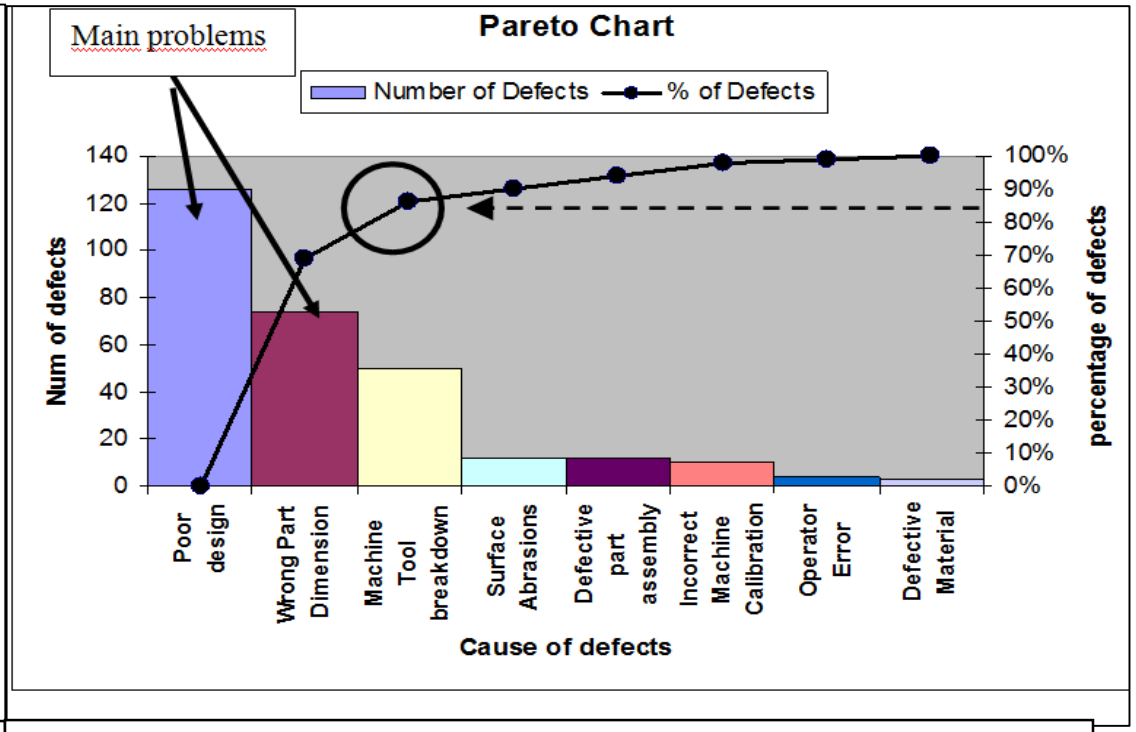
PRINCIPE 4 : REGROUPEMENT DES DÉFAUTS

Loi de Pareto

Le principe 4 nécessiterait de se concentrer encore davantage sur le composant A que sur le composant B dans cette situation.



Exemple de distribution de type Pareto selon les composants

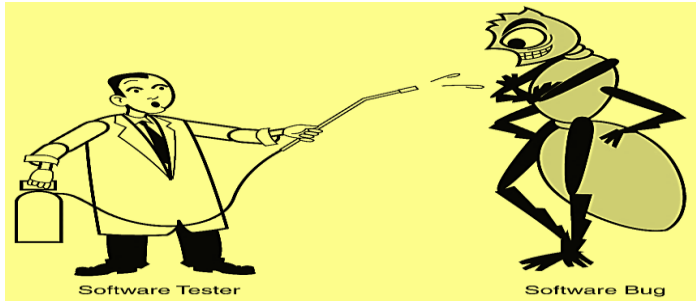


Exemple de distribution de type Pareto selon les cause des défauts

PRINCIPE 5 : USURE DES TESTS

- Si les mêmes tests sont répétés de nombreuses fois, le même ensemble de cas de tests finira par ne plus détecter de nouveaux défauts.
- Pour détecter de nouveaux défauts, il peut être nécessaire de modifier les tests existants et les données de test existantes, ainsi que de rédiger de nouveaux tests.
- Dans certains cas, la répétition des mêmes tests peut avoir un effet bénéfique, par exemple dans les tests de régression automatisés

Paradoxe des pesticides : c'est un paradoxe qui nous vient du monde agricole. Plus on utilise de pesticides, plus on doit en utiliser car les « mauvaises herbes » deviennent de plus en plus résistante.



PRINCIPE 6: LES TESTS DÉPENDENT DU CONTEXTE

- Les tests sont effectués différemment **dans des contextes différents**.

- Exemples :
  - le test dans un projet Agile est effectué différemment du test dans un projet à cycle de vie séquentiel
  - Un site Web où les informations peuvent simplement être consultées sera testé d'une manière différente d'un site de commerce électronique, où les marchandises peuvent être achetées à l'aide de cartes de crédit/débit.
    - Pour un site e-commerce, il faut se concentrer sur les aspects de sécurité. Est-il possible de contourner l'utilisation de mots de passe ? Le « paiement » peut-il être effectué avec une carte de crédit invalide, en saisissant des données excessives dans le numéro de carte ?
  - Les tests de sécurité sont un exemple de domaine spécialisé, qui ne convient pas à toutes les applications. Ces types de tests peuvent nécessiter du personnel spécialisé et des outils logiciels.

PRINCIPE 7 : L'ILLUSION DE L'ABSENCE DE DÉFAUT.

- Certaines organisations s'attendent à ce que les testeurs puissent effectuer tous les tests possibles et trouver tous les défauts possibles, mais les principes 2 et 1, respectivement, nous disent que c'est impossible.
- De plus, il est illusoire (c.-à-d. une croyance erronée) de s'attendre à ce que le simple fait de trouver et de corriger un grand nombre de défauts garantisse la réussite d’un système.

Rôles dans le test

Test manager

Testeur

RÔLES DANS LE TEST

Rôle: test manager

- Le rôle de test manager implique une responsabilité globale pour le processus de test, l'équipe de test et la direction des activités de test.
- Le rôle de test manager est principalement axé sur les activités de planification des tests, de pilotage et de contrôle des tests et de clôture des tests. La manière dont le rôle de test manager est exercé varie en fonction du contexte

Rôle: testeur

- Le rôle de testeur implique une responsabilité globale pour l'aspect technique des tests.
- Le rôle de testeur est principalement axé sur les activités d'analyse de test, de conception des tests, d'implémentation des tests et d'exécution des tests.

Différentes personnes peuvent assumer ces rôles à différents moments. Par exemple, le rôle de test manager peut être assumé par un responsable de test, par un chef de projet de test, par un responsable des développements, etc. Il est également possible qu'une personne assume à la fois les rôles de testeur et de test manager.

QU’EST CE QU’UN TESTEUR LOGICIEL ?

Testeur logiciel : Un professionnel qualifié qui participe aux tests.

Les testeurs utilisent des outils , mais il est important de rappeler que le test est en grande partie une activité intellectuelle, exigeant des testeurs qu'ils aient des connaissances spécialisées, qu'ils utilisent des compétences analytiques et qu'ils appliquent la pensée critique et la pensée systémique.

COMPÉTENCES ESSENTIELLES ET BONNES PRATIQUES EN MATIÈRE DE TEST

Compétences

Bonnes pratiques

COMPÉTENCES GÉNÉRIQUES POUR LE TEST

- Connaissance en matière de test (pour accroître l'efficacité des tests, par exemple en utilisant des techniques de test).
- Rigueur, attention, curiosité, souci du détail, méthode (pour identifier les défauts, en particulier ceux qui sont difficiles à trouver).
- Bonne communication, écoute active, esprit d'équipe (pour interagir efficacement avec toutes les parties prenantes, pour transmettre des informations aux autres, pour se faire comprendre, et pour signaler et discuter des défauts).
- Réflexion analytique, esprit critique, créativité (pour accroître l'efficacité des tests).
- Connaissances techniques (pour accroître l'efficience des tests, par exemple en utilisant les outils de test appropriés).
- Connaissance du domaine (pour être en mesure de comprendre les utilisateurs finaux/représentants métier et de communiquer avec eux).

PSYCHOLOGIE HUMAINE ET TEST (1)

Biais de confirmation

- Identifier les défauts peut être perçu comme une critique du produit et de ses auteurs.
- Un élément de la psychologie humaine appelé biais de confirmation peut rendre difficile à accepter des informations en désaccord avec les croyances actuelles.

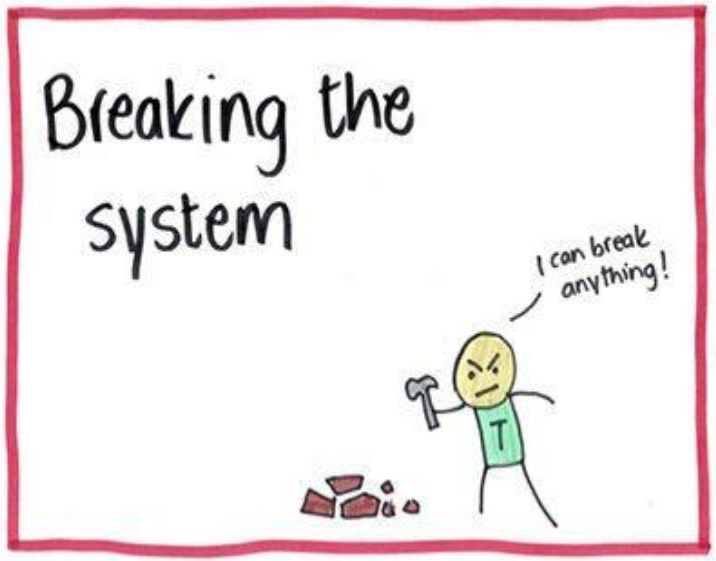
→ Puisque les développeurs attendent de leur code qu'il soit correct, ils ont un biais de confirmation qui fait qu'il est difficile d'accepter que le code est incorrect

Le biais de confirmation, **c'est la tendance instinctive de l'esprit humain à rechercher en priorité les informations qui confirment sa manière de penser, et à négliger tout ce qui pourrait la remettre en cause**.

PSYCHOLOGIE HUMAINE ET TEST (1)

Test est une activité destructive !

- C'est un trait humain commun de blâmer le porteur de la mauvaise nouvelle, et l'information produite par le test contient souvent de mauvaises nouvelles.
- Certaines personnes peuvent percevoir le test comme une activité destructrice, même s'il contribue grandement à l'avancement du projet et à la qualité du produit.



PSYCHOLOGIE HUMAINE ET TEST (3)

Compétences interpersonnelles

- Pour tenter de réduire ces perceptions, l'information sur les défauts et les défaillances **devrait être communiquée de façon constructive**.
- Les testeurs et les Test Managers doivent avoir de bonnes compétences interpersonnelles
  - pour être capable de communiquer efficacement sur les défauts, les défaillances, les résultats des tests, l’avancement des tests et les risques,
  - pour établir des relations positives avec leurs collègues.

PSYCHOLOGIE HUMAINE ET TEST (3)

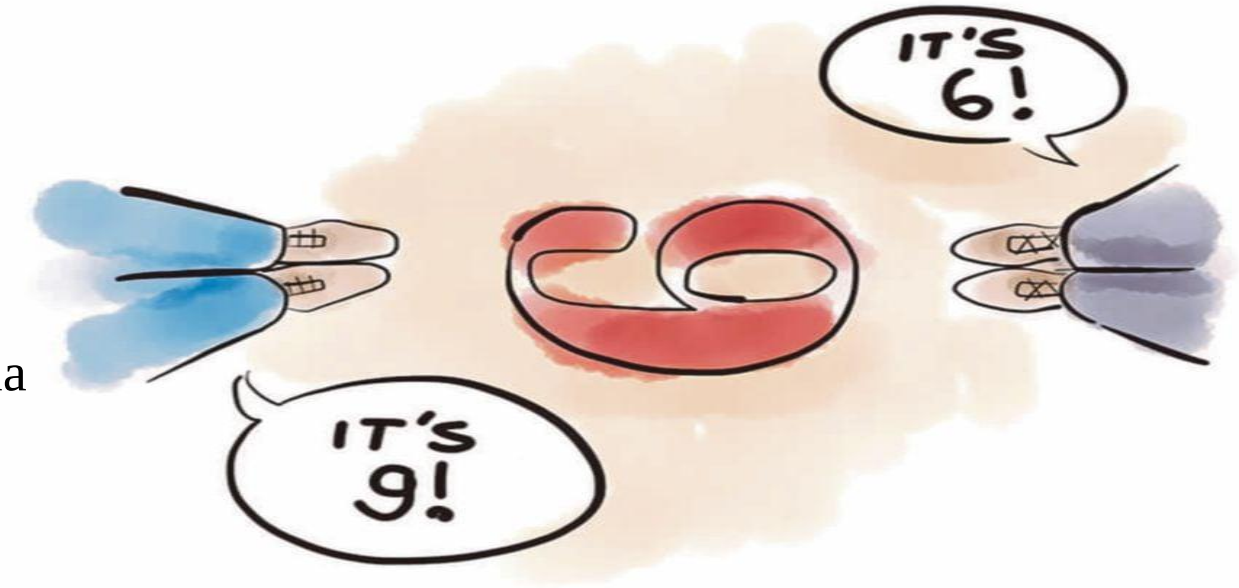
Bien communiquer

- Commencer par la collaboration plutôt que par la confrontation. Rappeler à tous l'objectif commun d'une meilleure qualité des systèmes.
- Mettre l'accent sur les bénéfices du test.
- Communiquer les résultats des tests et d'autres constats d'une manière neutre et axée sur les faits, sans critiquer la personne qui a créé l’item défectueux. Rédiger des rapports objectifs et factuels sur les défauts et les constats des revues.
- Essayer de comprendre ce que ressent l'autre personne et les raisons pour lesquelles elle peut réagir négativement à l'information.
- confirmer que l'autre personne a compris ce qui a été dit et vice versa

ETAT D'ESPRIT DES TESTEURS ET DES DÉVELOPPEURS

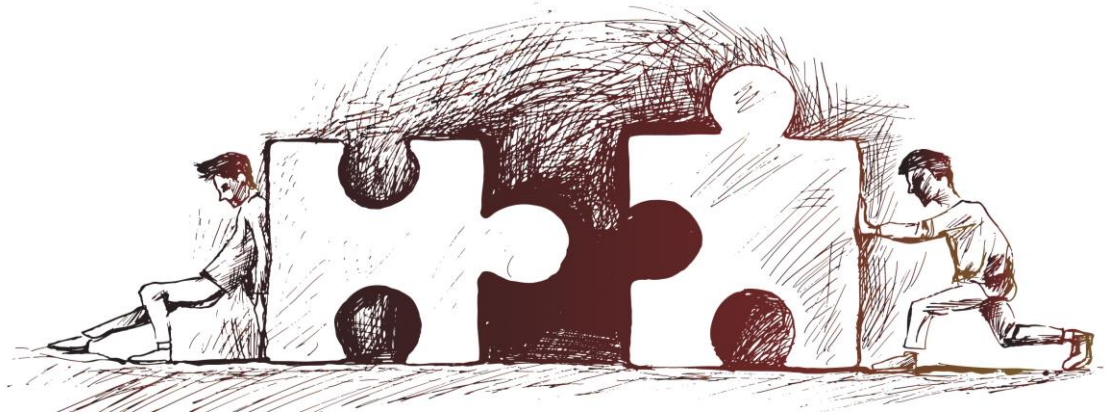
développeur

Testeur



Les développeurs et les testeurs pensent souvent différemment.

- L'objectif premier du développement est de concevoir et de construire un produit.
- Les objectifs du test comprennent la vérification et la validation du produit, la détection des défauts avant la livraison, ….



- L'union de ces points de vue permet d'atteindre un niveau plus élevé de qualité des produits.
- L'état d'esprit d'un testeur devrait inclure la curiosité, un pessimisme professionnel, un œil critique, l'attention aux détails et une motivation pour de bonnes et positives communications et relations.

APPROCHE ÉQUIPE INTÉGRÉE

- L'une des compétences importantes pour un testeur est la capacité à travailler efficacement dans un contexte d'équipe et à contribuer positivement aux objectifs de l'équipe.
- L'approche équipe intégrée – **une pratique issue de la programmation extrême (XP)** repose sur cette compétence.
- Dans l'approche intégrée, tout membre de l'équipe disposant des connaissances et des compétences nécessaires peut effectuer n'importe quelle tâche, et chacun est responsable de la qualité.
- Les membres de l'équipe partagent le même espace de travail (physique ou virtuel), car le regroupement facilite la communication et l'interaction.
- L'approche équipe intégrée améliore la dynamique d'équipe, la communication et la collaboration au sein de l'équipe, et crée une synergie en permettant aux différents ensembles de compétences au sein de l'équipe d'être exploités au profit du projet.

APPROCHE ÉQUIPE INTÉGRÉE

- Les testeurs travaillent en étroite collaboration avec les autres membres de l'équipe afin de garantir que les niveaux de qualité souhaités sont atteints. Ils collaborent notamment avec les représentants métier pour les aider à créer des tests d'acceptation adaptés et travaillent avec les développeurs pour convenir de la stratégie de test et décider des approches d'automatisation des tests. Les testeurs peuvent ainsi transférer leurs connaissances en matière de test aux autres membres de l'équipe et influencer le développement du produit.
- Selon le contexte, l'approche équipe intégrée n'est pas toujours appropriée. Par exemple, dans certaines situations, telles que les situations critiques pour la sécurité, un niveau élevé d'indépendance du test peut être nécessaire.

INDÉPENDANCE DES TESTS

- Un certain degré d'indépendance rend le testeur plus efficace pour trouver des défauts en raison des différences entre les biais cognitifs de l'auteur et ceux du testeur.
- L'indépendance ne remplace toutefois pas la proximité ; par exemple, les développeurs peuvent trouver efficacement de nombreux défauts dans leur propre code.
- Pour la plupart des projets, il est généralement préférable d'effectuer les tests avec plusieurs niveaux d'indépendance (par exemple, les développeurs effectuant les tests de composants et d'intégration de composants, l'équipe de test effectuant les tests de systèmes et d'intégration de systèmes, et les représentants métier effectuant les tests d'acceptation).

INDÉPENDANCE DES TESTS

Avantages et inconvénients

Avantages

- les testeurs indépendants sont susceptibles de repérer différents types de défaillances et de défauts par rapport aux développeurs en raison de leurs différents antécédents, perspectives techniques et biais
- un testeur indépendant peut vérifier, contester ou réfuter les hypothèses formulées par les parties prenantes lors de la spécification et de la mise en œuvre du système.

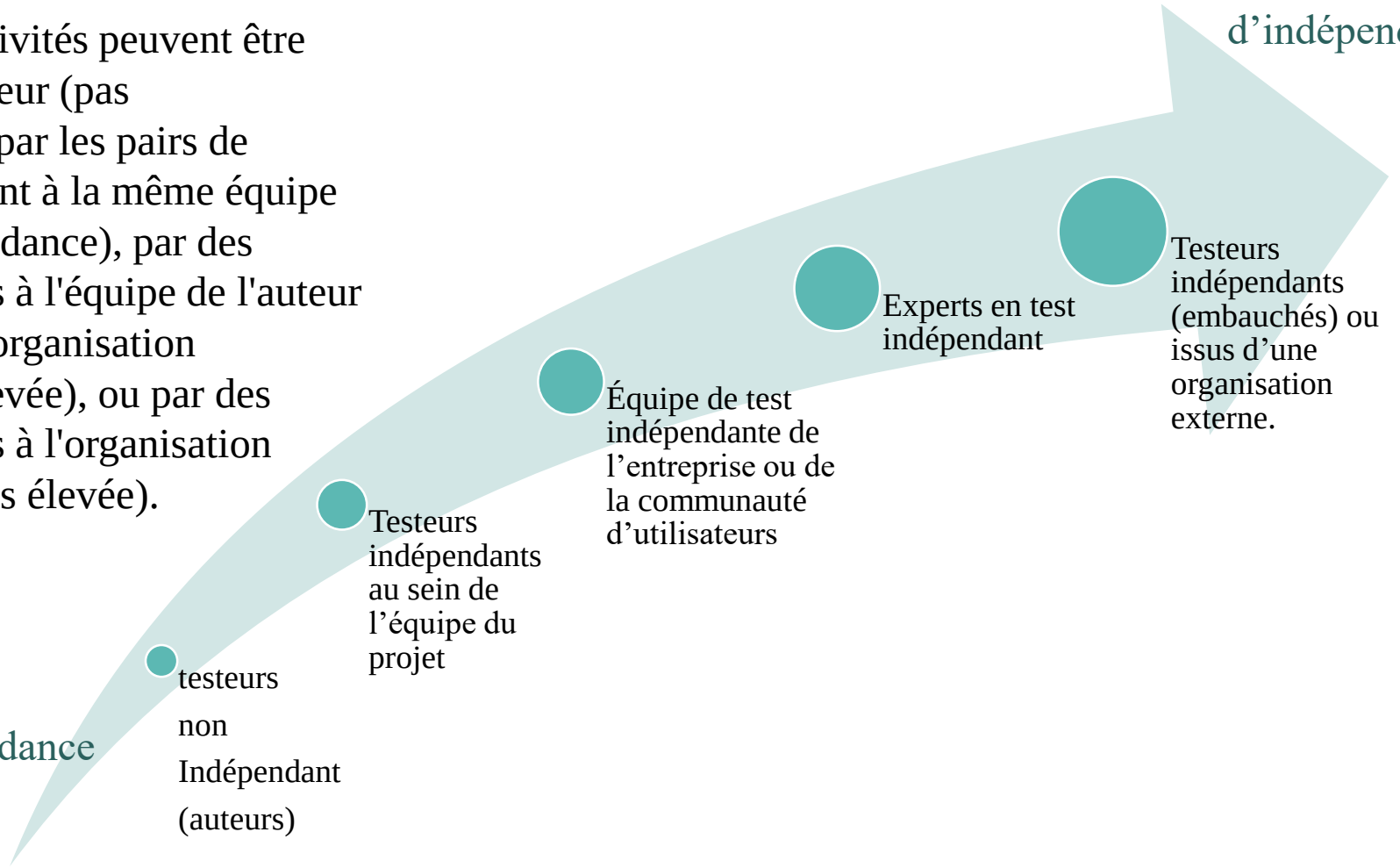
Inconvénients

- Les testeurs indépendants peuvent être isolés de l'équipe de développement, ce qui peut entraîner un manque de collaboration, des problèmes de communication ou une relation d'opposition avec l'équipe de développement.
- Les développeurs peuvent perdre le sens de la responsabilité en matière de qualité.
- Les testeurs indépendants peuvent être considérés comme un goulot d'étranglement ou être tenus pour responsables des retards dans la release(livraison du logiciel).

INDÉPENDANCE DES TESTS

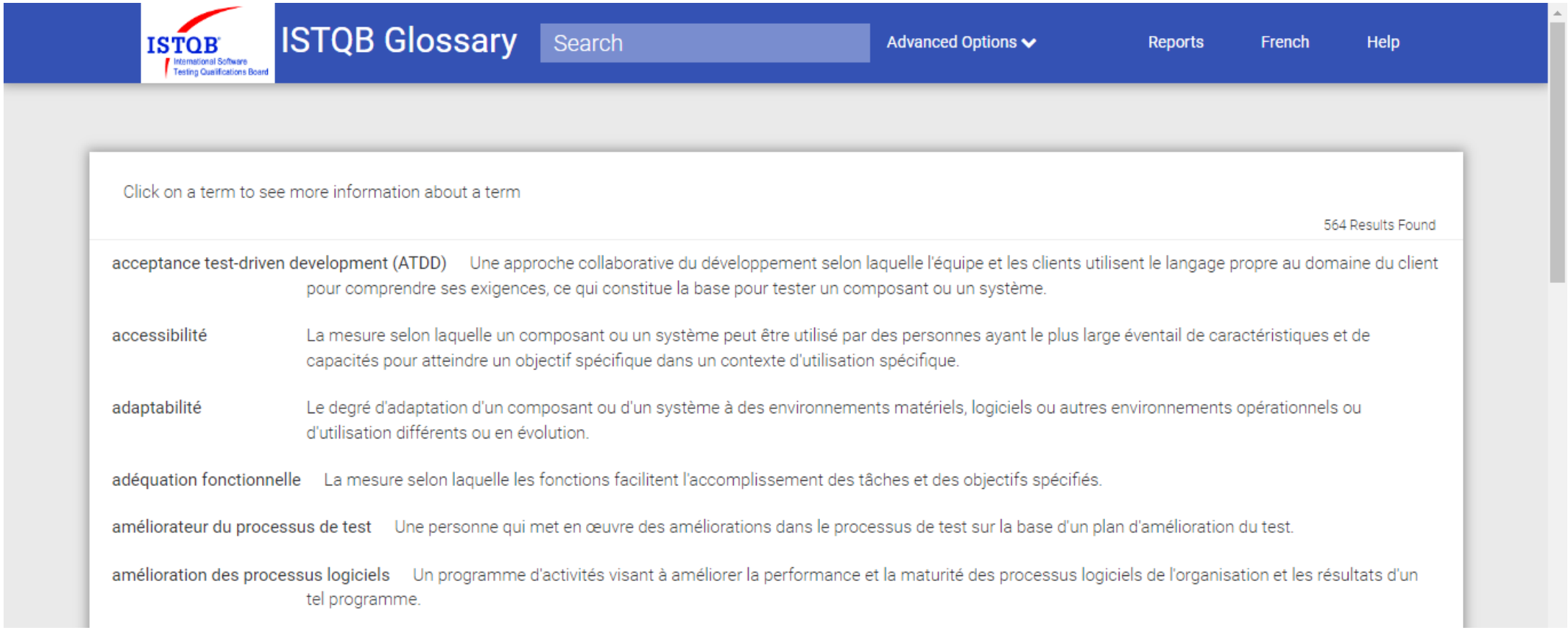
Les niveaux d'indépendance des tests

Les produits d'activités peuvent être testés par leur auteur (pas d'indépendance), par les pairs de l'auteur appartenant à la même équipe (un peu d'indépendance), par des testeurs extérieurs à l'équipe de l'auteur mais au sein de l'organisation (indépendance élevée), ou par des testeurs extérieurs à l'organisation (indépendance très élevée).

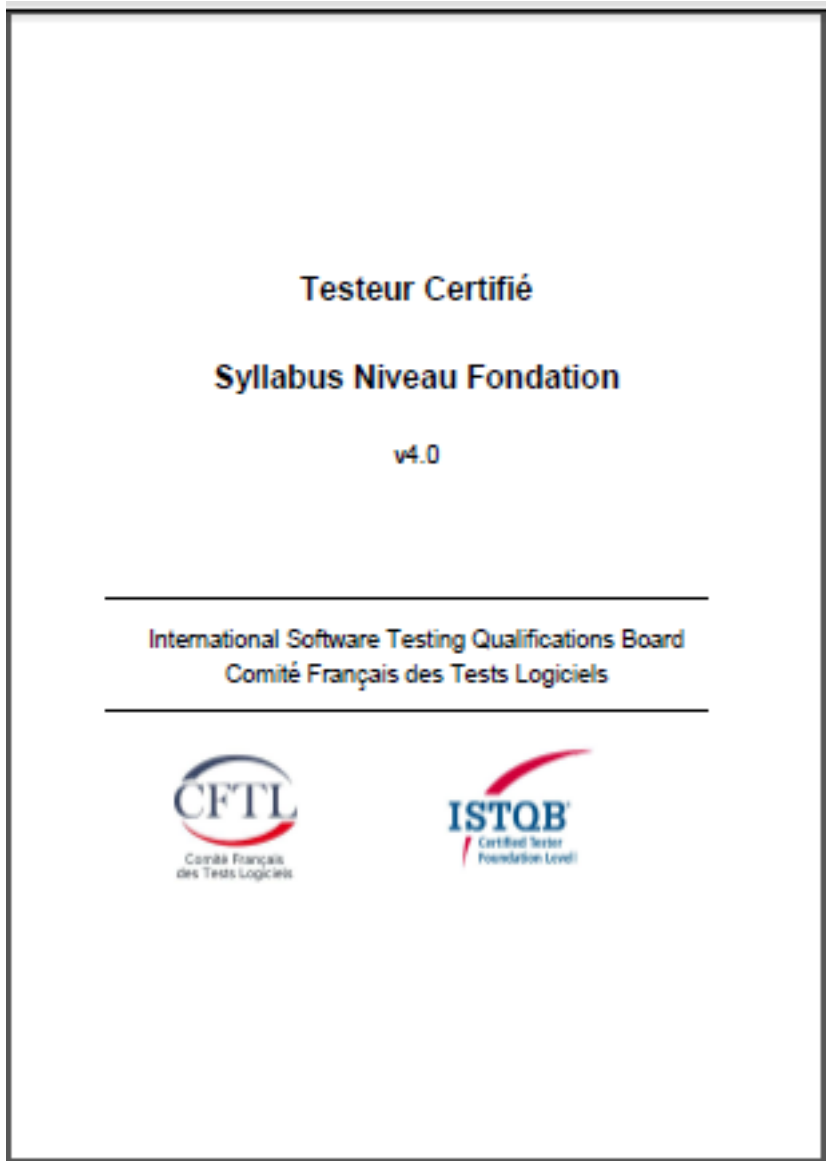


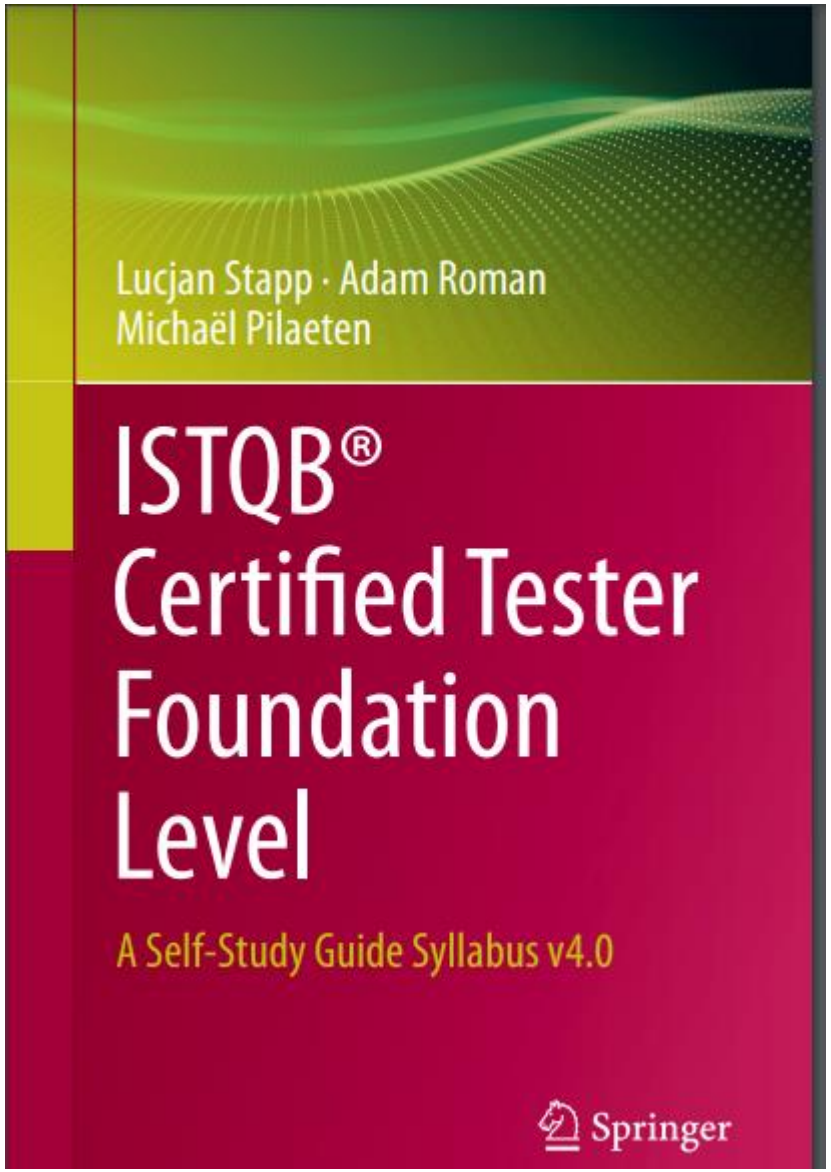
GLOSSAIRE - ISTQB

https://glossary.istqb.org/fr/search



RÉFÉRENCES


Testeur Certifié
Syllabus Niveau Fondation
v4.0
International Software Testing Qualifications Board
Comité Français des Tests Logiciels
CFTL
Comité Français des Tests Logiciels
ISTQB
Certified Tester Foundation Level


Lucjan Stapp · Adam Roman
Michaël Pilaeten
ISTQB®
Certified Tester
Foundation
Level
A Self-Study Guide Syllabus v4.0
Springer