

# Support de formation

Version 2.09 – Juin 2019  
Sogeti Test Academy



---

Certification au niveau  
Fondation **version 2018**

# Plan du cours

ISTQB® - Niveau Fondation					
Fondamentaux des Tests	Tester pendant le cycle de vie du développement logiciel	Tests Statiques	Techniques de Tests	Gestion des tests	Outils de support aux tests
Que sont les tests ?	Les modèles de développement logiciel	Bases des tests statiques	Catégories de techniques de test	Organisation des tests	Introduction aux outils de test
Pourquoi les tests sont-ils nécessaires ?	Niveaux de tests	Processus de revue	Techniques de test boîte-noire	Planification et estimation des tests	Utilisation efficace des outils
7 principes sur les tests	Types de tests		Techniques de test boîte-blanche	Pilotage et contrôle des tests	
Processus de test	Tests de maintenance		Techniques de test basées sur l'expérience	Gestion de configuration	
La psychologie des tests				Risques et tests	
				Gestion des défauts	

# 1 Fondamentaux des tests

## 1.1 Que sont les tests ?



Certification au niveau  
Fondation **version 2018**

# Tests dynamiques vs statiques

Certains tests impliquent **l'exécution** du composant ou du système testé. Ces tests sont appelés **tests dynamiques**.



D'autres tests n'impliquent pas l'exécution du composant ou du système testé ; de tels tests sont appelés **tests statiques**.

Ainsi, les tests comprennent également la **revue de produits d'activités** amont tels que les exigences, les User Stories et le code source.



## 1.1.1 Objectifs habituels des tests

**Évaluer des produits d'activités** tels que les exigences, les User Stories, la conception et le code

**Vérifier** si toutes les exigences spécifiées ont été satisfaites

**Valider** si l'objet de test est complet et fonctionne comme attendu par les utilisateurs et autres parties prenantes

Construire la **confiance dans le niveau de qualité** de l'objet de test

**Prévenir** des défauts

Trouver des **défaillances et défauts**

**Fournir suffisamment d'information** aux parties prenantes pour leur permettre de **prendre des décisions éclairées**, en particulier en ce qui concerne le niveau de qualité de l'objet de test

**Réduire le niveau de risque** d'une qualité logicielle inadéquate

**Se conformer aux exigences ou aux normes** contractuelles, légales ou réglementaires, et/ou **vérifier la compatibilité de l'objet** de test avec de telles exigences ou normes

## 1.1.2 Test et débogage

L'exécution de **tests** peut **mettre en évidence des défaillances** causées par des défauts dans le logiciel.

Le **débogage** est l'activité de **développement** qui trouve (localise), analyse et corrige les **défauts** dans les composants concernés (cause des défaillances).

Par la suite, le **test de confirmation** vérifie si les corrections apportées ont résolu les défaillances.

Classiquement, les testeurs sont responsables du test initial et du test de confirmation final, tandis que les développeurs se chargent du débogage et du test des composants concernés.

Cependant, dans certains cycles de développement comme **Agile**, les testeurs peuvent être impliqués dans le débogage et le test des composants.

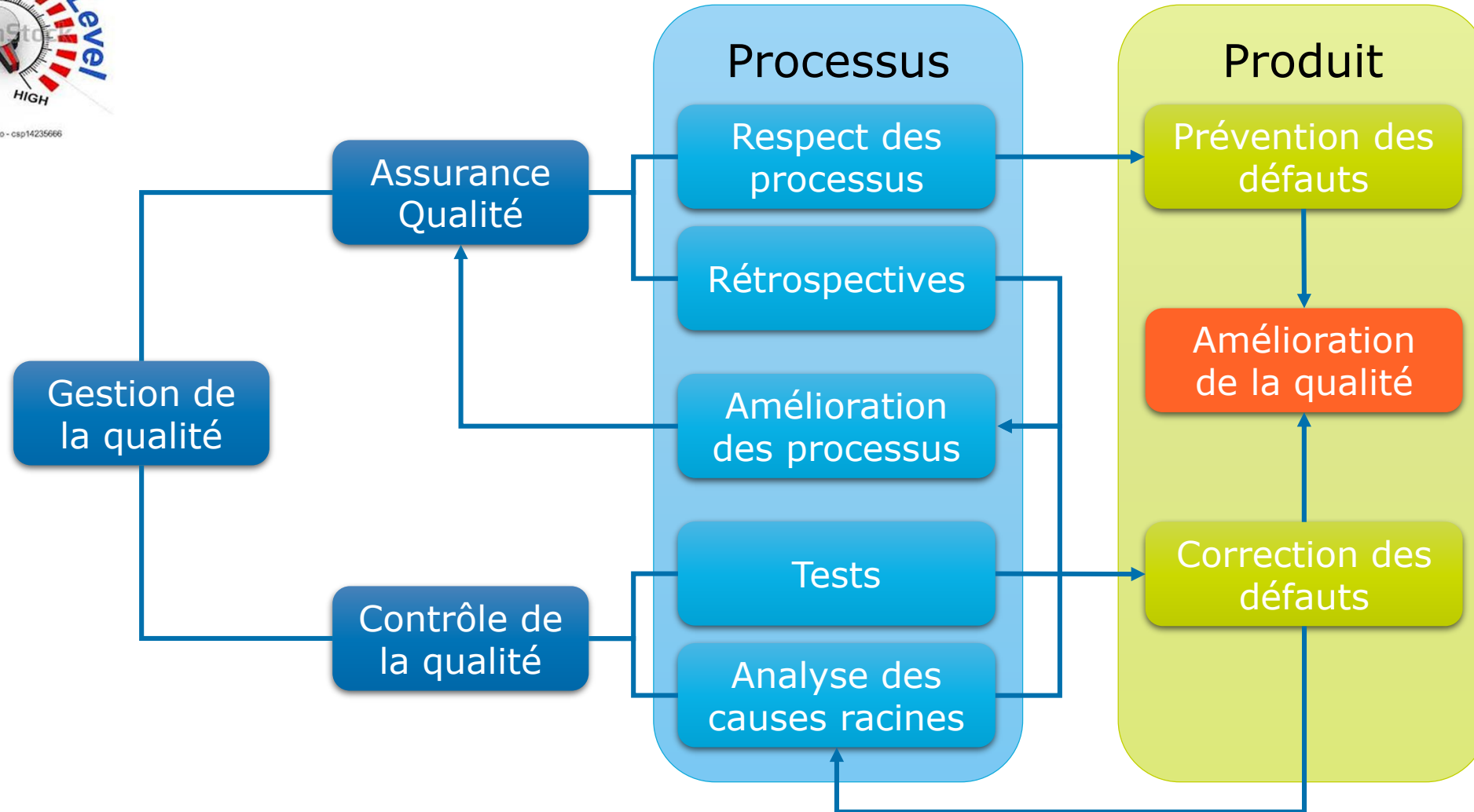
# 1 Fondamentaux des tests

## 1.2 Pourquoi les tests sont-ils nécessaires ?



Certification au niveau  
Fondation **version 2018**

## 1.2.2 Assurance qualité et test





## 1.2.3 Erreurs, défauts et défaillances

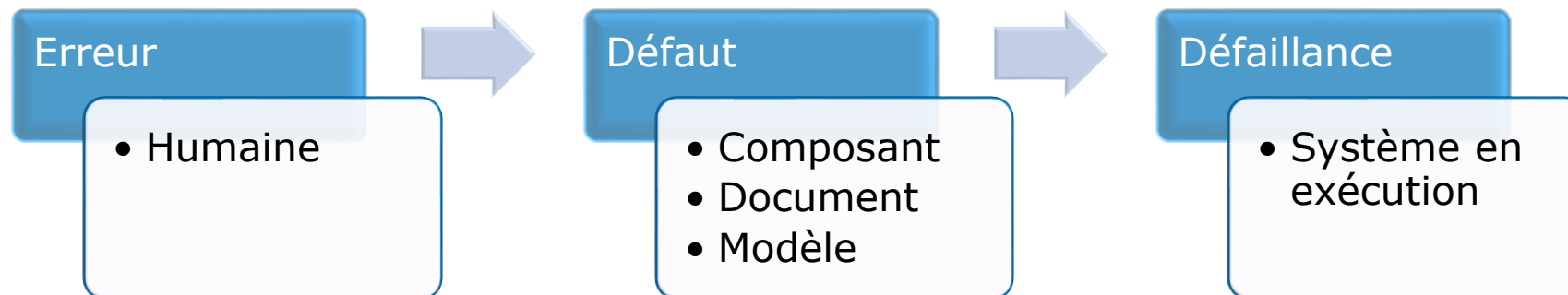
Un être humain peut faire une **erreur**, qui produit une anomalie (**défaut, bug**) dans le code d'un logiciel, ou dans un autre produit d'activités connexe (document, modèle...).

Une erreur qui conduit à l'introduction d'un défaut dans un produit d'activités peut déclencher l'introduction d'un défaut dans un produit d'activités dérivé.

- Par exemple, une erreur d'élucidation d'exigences peut conduire à un défaut au niveau des exigences, qui se répercute ensuite dans un défaut dans le code.

Si un code comportant un défaut est exécuté, le système peut ne pas faire ce qu'il aurait dû faire (ou peut faire ce qu'il n'aurait pas dû faire), générant une **défaillance**.

Des défauts dans les logiciels, systèmes ou documents peuvent générer des défaillances, **mais tous les défauts ne le font pas**.



## 1.2.3 Erreurs, défauts et défaillances

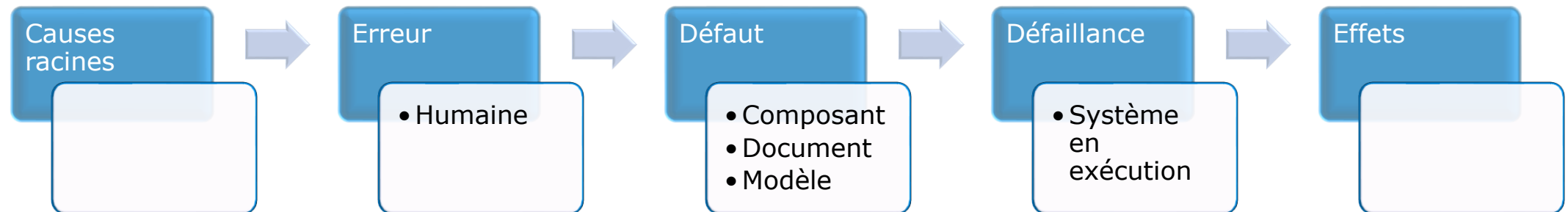
- **Faux positifs** : Défauts détectés qui n'en sont pas
- Les **faux négatifs** sont des tests qui ne détectent pas les défauts qu'ils auraient dû détecter.

## 1.2.4 Défauts, causes racines et effets

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 la probabilité d'apparition de défauts similaires à l'avenir.

En se concentrant sur les causes racines **les plus importantes, l'analyse des causes racines** peut conduire à des améliorations de processus qui **préviennent** l'introduction d'un nombre important de défauts futurs.



# 1 Fondamentaux des tests

## 1.3 7 principes sur les tests



Certification au niveau  
Fondation **version 2018**

# Les 7 principes généraux des tests

**Principe 1 – Les tests montrent la présence de défauts, pas leur absence**

**Principe 2 – Les tests exhaustifs sont impossibles**

**Principe 3 – Tester tôt économise du temps et de l'argent**

**Principe 4 – Regroupement des défauts**

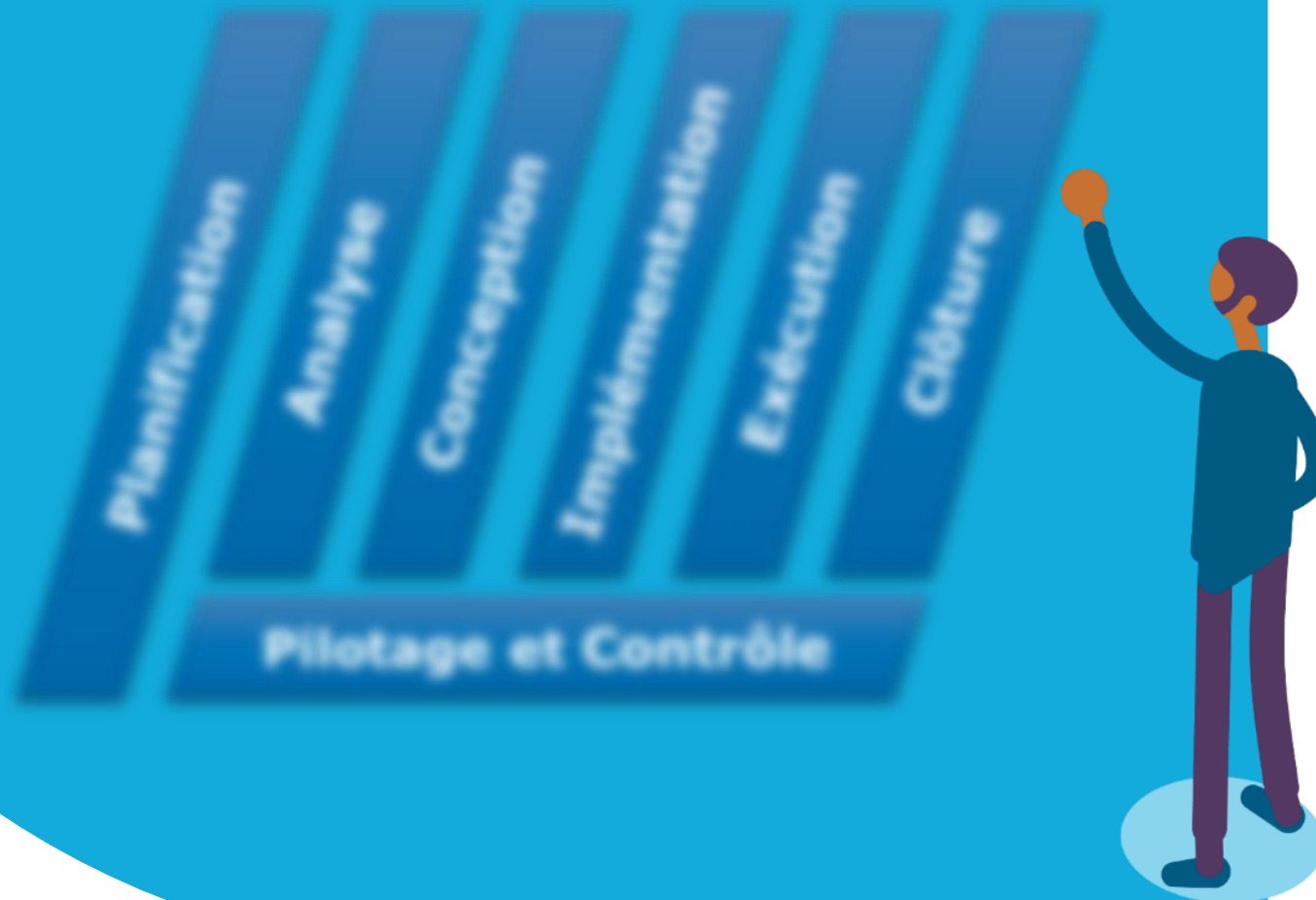
**Principe 5 – Paradoxe du pesticide**

**Principe 6 – Les tests dépendent du contexte**

**Principe 7 – L'absence d'erreurs est une illusion**

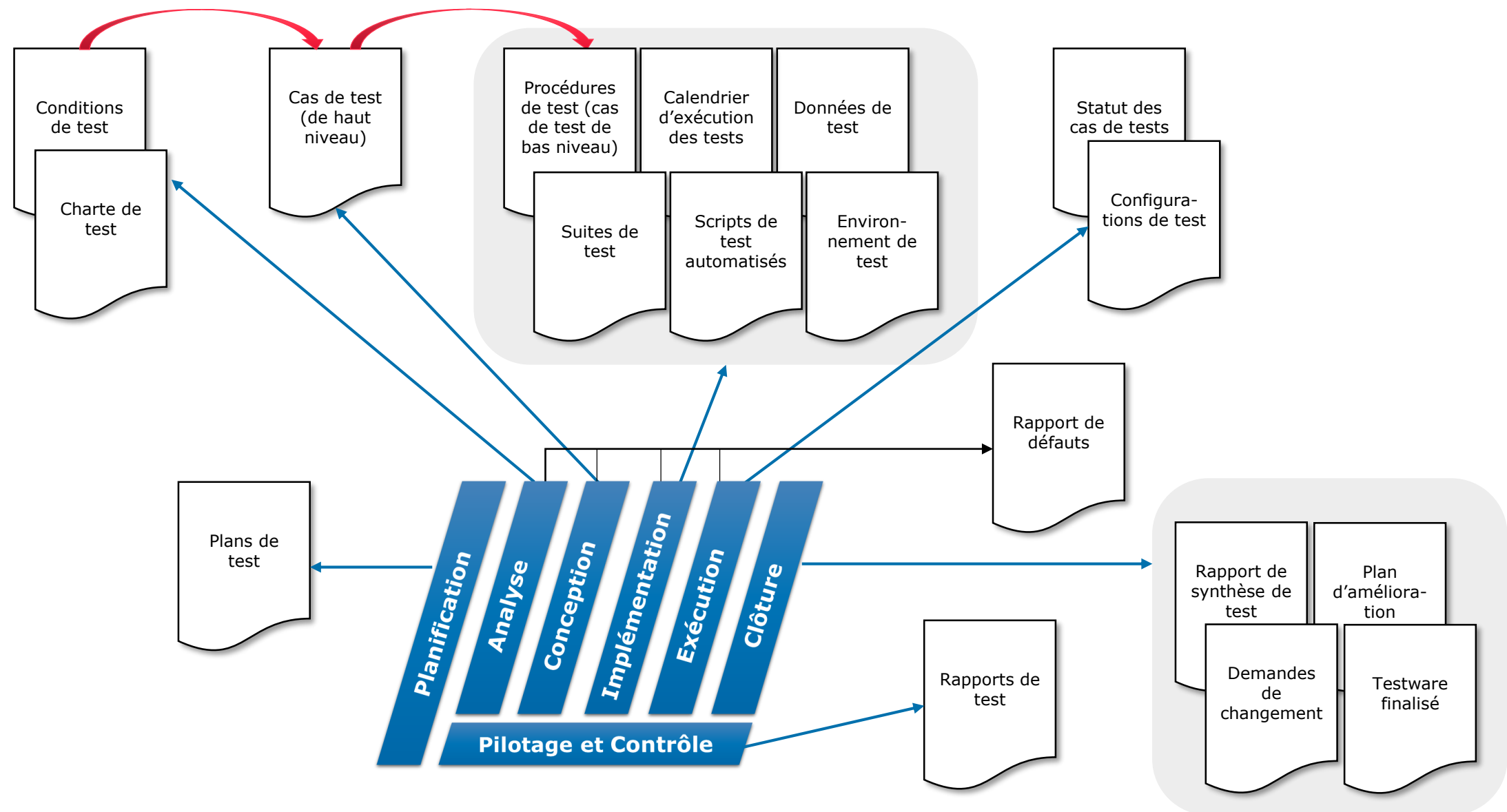
# 1 Fondamentaux des tests

## 1.4 Processus de test



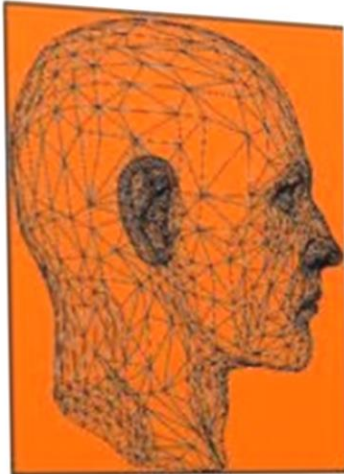
Certification au niveau  
Fondation **version 2018**

# 1.4.3 Les produits d'activités du test



# 1 Fondamentaux des tests

## 1.5 La psychologie des tests



**Software Developer**

*"Software testing is the process to prove that the software works correctly."*



**Software Tester**

*"Testing is the process to prove that the software doesn't work."*

*Test négatif*



Certification au niveau  
Fondation **version 2018**



## 1.5.2 Etat d'esprit des testeurs et des développeurs

Un état d'esprit reflète les hypothèses d'une personne et les méthodes qu'elle préfère pour la prise de décision et la résolution de problèmes.

**L'état d'esprit d'un testeur** devrait inclure :

- **Curiosité**
- **Pessimisme** professionnel (qui engendre la vigilance)
- Œil **critique**
- Attention aux **détails**
- Motivation pour les **bonnes relations**.

L'état d'esprit d'un testeur tend à grandir et à mûrir au fur et à mesure que le testeur acquiert de l'expérience.

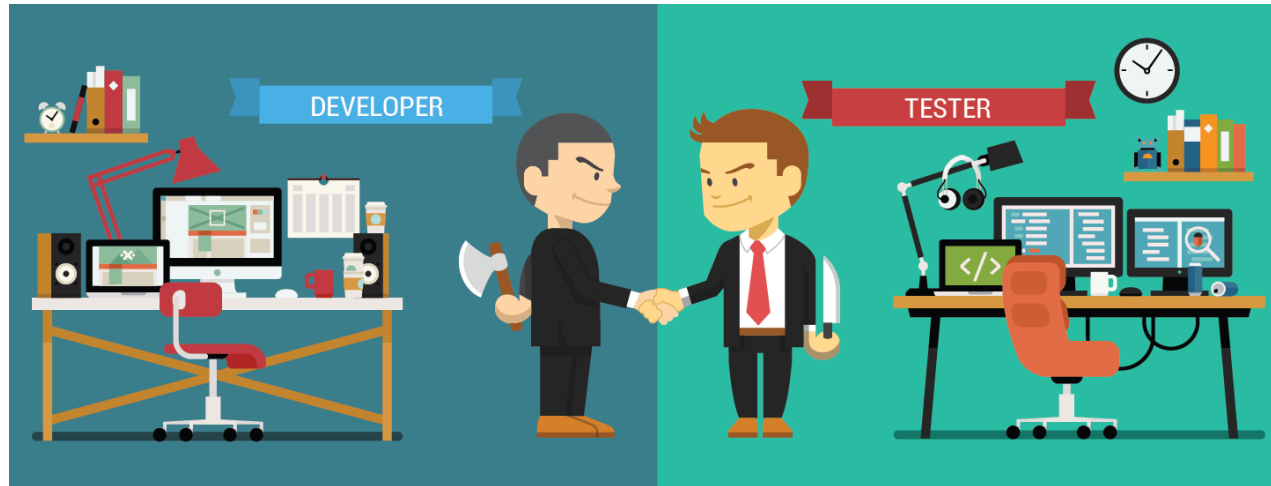
L'état d'esprit d'un développeur peut inclure certains éléments de l'état d'esprit d'un testeur, mais les développeurs sont souvent plus intéressés par la **conception** et la **construction** de solutions que par la recherche de ce qui pourrait être erroné.

## 1.5.2 Etat d'esprit des testeurs et des développeurs

Avec le bon état d'esprit, les développeurs sont capables de tester leur propre code.

Le fait que certaines des activités de test soient effectuées par des testeurs indépendants augmente **l'efficacité de la détection des défauts**, ce qui est particulièrement important pour les **systèmes de grande taille, complexes ou critiques** sur le plan de la sûreté.

Les testeurs indépendants apportent une **perspective différente** de celle des auteurs du produit d'activités testé (c.-à-d. les analystes métier, les Product Owners, les concepteurs et les développeurs), puisqu'ils ont des **biais cognitifs différents** de ceux des auteurs.



# 1 Fondamentaux des tests

## Termes à retenir



Certification au niveau  
Fondation **version 2018**

# Termes à retenir

**Système** : Ensemble d'éléments en interaction organisés pour accomplir une fonction ou un ensemble de fonctions spécifiques.

**(le) Test** : Processus consistant en toutes les activités du cycle de vie, statiques et dynamiques, concernant la planification et l'évaluation de produits logiciels et produits liés pour déterminer s'ils satisfont aux exigences et démontrer qu'ils sont conformes aux objectifs et détecter des anomalies.

**(un) Test** : un ensemble d'un ou plusieurs cas de tests.

**Testware** : Produits des activités réalisées au cours des différentes activités de test.

**Traçabilité** : La mesure dans laquelle une relation peut être établie entre deux ou plusieurs produits d'activités.

**Validation** : Confirmation par l'examen et la fourniture de preuves objectives que les exigences, pour un usage ou une application voulue, ont été satisfaites.

**Vérification** : Confirmation par l'examen et la fourniture de preuves objectives que des exigences spécifiées ont été satisfaites.

## 2 Tester pendant le cycle de vie du développement logiciel

### 2.1 Les modèles de développement logiciel



Certification au niveau  
Fondation **version 2018**

## 2.1.1 Développement de logiciel et tests logiciels

Un modèle de **cycle de vie de développement logiciel (SDLC)** décrit :

- les types d'activités réalisées à chaque étape d'un projet de développement logiciel,
- et la façon dont les activités sont reliées entre elles logiquement et chronologiquement.

Il existe un certain nombre de modèles de cycle de développement de logiciels différents, chacun d'entre eux **nécessitant des approches de test différentes** :

- Modèles de développement **séquentiel**
  - **Modèle en cascade**
  - **Modèle en V**
- Modèles de développement **itératif** et **incrémental**
  - **RUP**
  - **SCRUM**
  - **Kanban**
  - **Cycle en spirale**
  - **DevOps**

# Modèles de développement itératif et incrémental

## Développement incrémental :

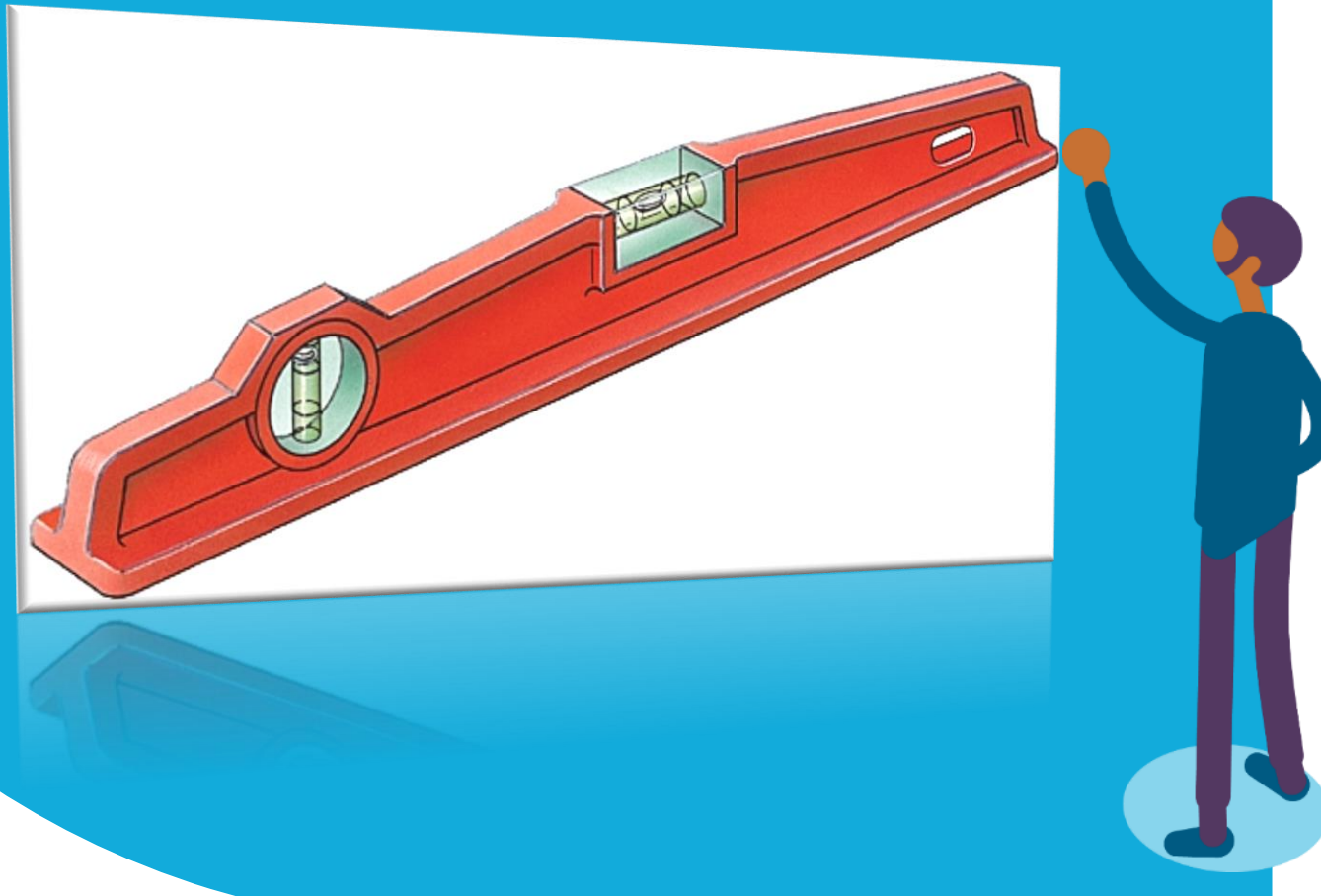


## Développement itératif :



# 2 Tester pendant le cycle de vie du développement logiciel

## 2.2 Niveaux de test



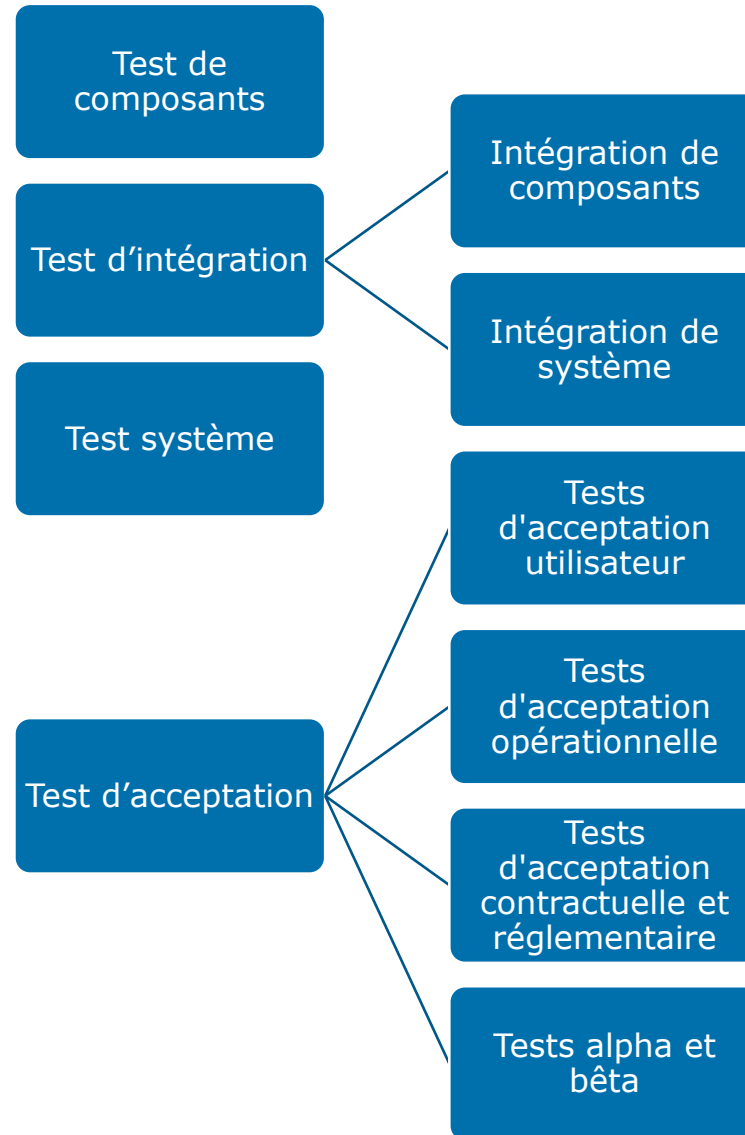
**CFTL**  
Comité Français  
des Tests Logiciels

**ISTQB**  
International Software  
Testing Qualifications Board

Certification au niveau  
Fondation **version 2018**

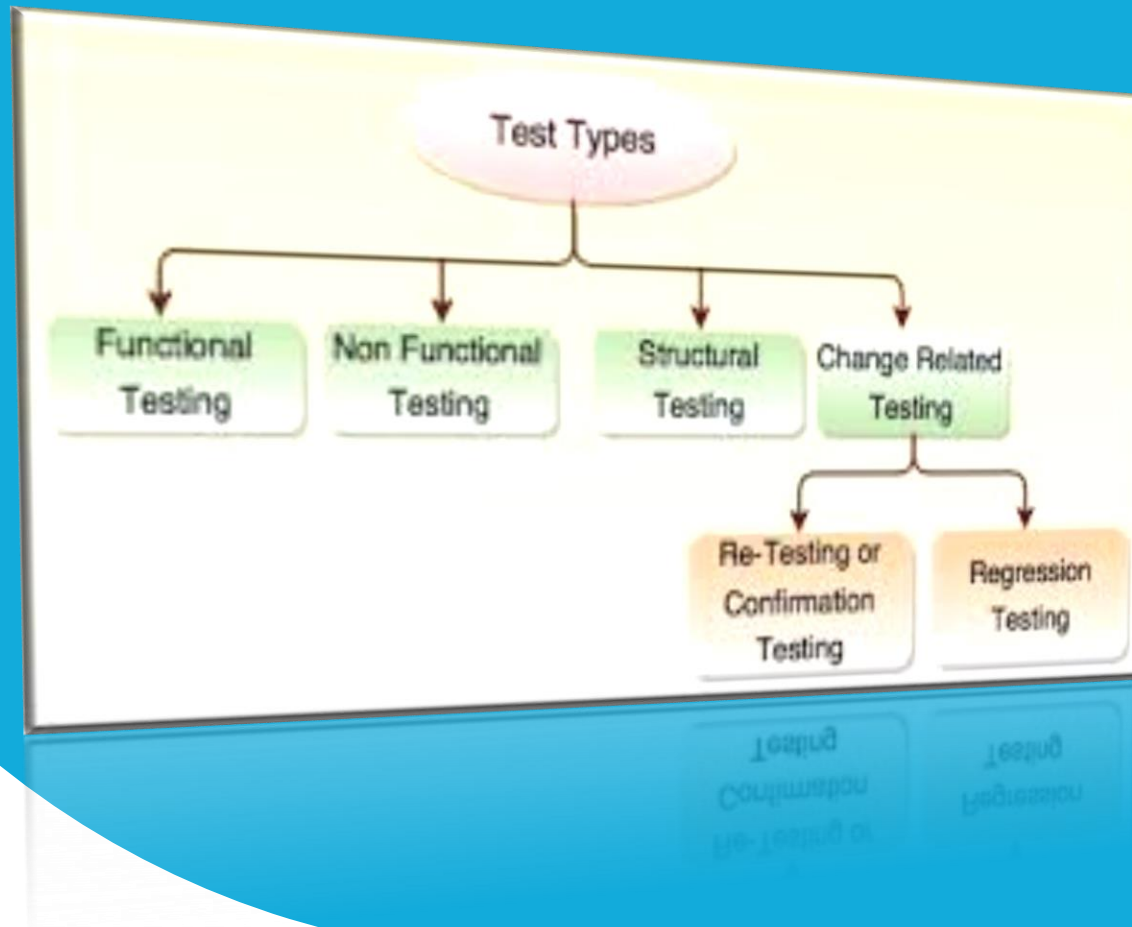


## 2.2 Niveaux de test



# 2 Tester pendant le cycle de vie du développement logiciel

## 2.3 Types de test

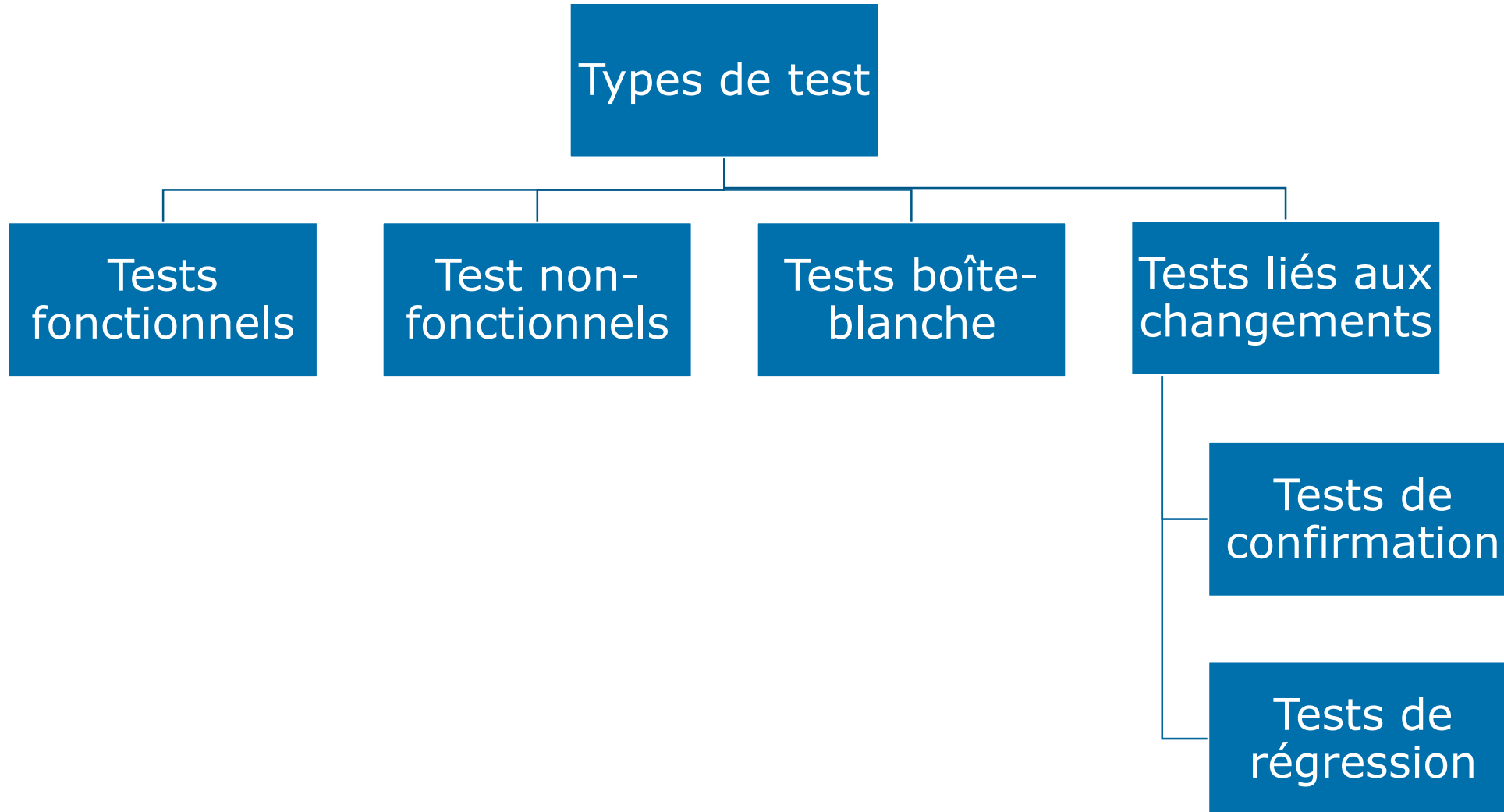


**CFTL**  
Comité Français  
des Tests Logiciels

**ISTQB**  
International Software  
Testing Qualifications Board

Certification au niveau  
Fondation **version 2018**

## 2.3 Types de test



# 2.3.5 Types de test et niveaux de test

	Tests fonctionnels	Tests non-fonctionnels	Tests boîte-blanche	Tests liés aux changements
Test de composants				
Test d'intégration				
Test système				
Test d'acceptation				

# 2 Tester pendant le cycle de vie du développement logiciel

## 2.4 Tests de maintenance



Certification au niveau  
Fondation **version 2018**

## 2.4.1 Facteurs déclencheurs pour la maintenance

- **Modification**

- améliorations planifiées (p. ex., basées sur des versions)
- changements correctifs et d'urgence
- changements de l'environnement opérationnel (OS, BD, COTS)

- **Migration**, par exemple d'une plate-forme à une autre, qui peut nécessiter

- des **tests opérationnels** du nouvel environnement ainsi que du logiciel migré
- des **tests de conversion** de données lorsque les données d'une autre application seront migrées dans le système en cours de maintenance

- **Déclassement**, lorsqu'une application arrive en fin de vie, qui peut nécessiter

- des **tests d'archivage et de restauration**
- des **tests de régression** pour s'assurer que les fonctionnalités non déclassées sont toujours opérationnelles

# 3 – Tests statiques

ISTQB® - Niveau Fondamental				
Fondamentaux des Tests	Tester pendant le cycle de développement logiciel	Techniques de conception de Tests	Gestion des tests	Outils de support aux tests
Que sont les tests ?	Les modèles de développement logiciel	Catégories de techniques de test	Organisation des tests	Introduction aux outils de test
Pourquoi les tests sont-ils nécessaires ?	Niveaux de test	Techniques de test boîte-noire	Planification et estimation des tests	Utilisation efficace des outils
7 principes sur les tests	Types de tests	Techniques de test te-blanche	Pilotage et contrôle des tests	
Processus de test	Tests de maintenance	Techniques de test basées sur l'expérience	Gestion de configuration	
La psychologie des tests			Risques et tests	
			Gestion des défauts	

Tests Statiques

Bases des tests statiques

Processus de revue



Certification au niveau  
Fondation **version 2018**

# 3 – Tests statiques

## 3.1 Bases des tests statiques

ISTQB® - Niveau Fondamental				
Fondamentaux des Tests	Tester pendant le cycle de développement logiciel	Techniques de conception de Tests	Gestion des tests	Outils de support aux tests
Que sont les tests ?	Les modèles de développement logiciel	Catégories de techniques de test	Organisation des tests	Introduction aux outils de test
Pourquoi les tests sont-ils nécessaires ?	Niveaux de test	Techniques de test boîte-noire	Planification et estimation des tests	Utilisation efficace des outils
7 principes sur les tests	Types de tests	Techniques de test boîte-blanche	Pilotage et contrôle des tests	
Processus de test	Tests de maintenance	Techniques de test basées sur l'expérience	Gestion de configuration	
La psychologie des tests			Risques et tests	
			Gestion des défauts	

Tests Statiques

Bases des tests statiques

Processus de revue



Certification au niveau  
Fondation **version 2018**



## 3.1.3 Différences entre les tests statiques et dynamiques

- Les deux ont pour objectifs l'évaluation de la qualité et l'identification des défauts.
- Les deux se complètent mutuellement en trouvant différents types de défauts

### ■ Tests statiques

- **Sans exécution** du logiciel, par **examen** des produits d'activité
- Détection directe des défauts
- Nécessite moins d'effort
- Permettent d'améliorer la qualité interne des produits d'activités

### ■ Tests dynamiques

- **Exécution** du logiciel
- Détection de défaillances
- Plus coûteux
- Se concentre généralement sur le comportement visible des produits

# 3 – Tests statiques

## 3.2 Processus de revue

ISTQB® - Niveau Fondamental					
Fondamentaux des Tests	Tester pendant le cycle de développement logiciel	Tests Statiques	Techniques de conception de Tests	Gestion des tests	Outils de support aux tests
Que sont les tests ?	Les modèles de développement logiciel	Bases des tests statiques	Catégories de techniques de test	Organisation des tests	Introduction aux outils de test
Pourquoi les tests sont-ils nécessaires ?	Niveaux de test		Techniques de test boîte-noire	Planification et estimation des tests	Utilisation efficace des outils
7 principes sur les tests	Types de tests	Processus de revue	Techniques de test boîte-blanche	Pilotage et contrôle des tests	
Processus de test	Tests de maintenance		Techniques de test basées sur l'expérience	Gestion de configuration	
La psychologie des tests				Risques et tests	
				Gestion des défauts	



Certification au niveau  
Fondation **version 2018**

## 3.2 – Processus de revue

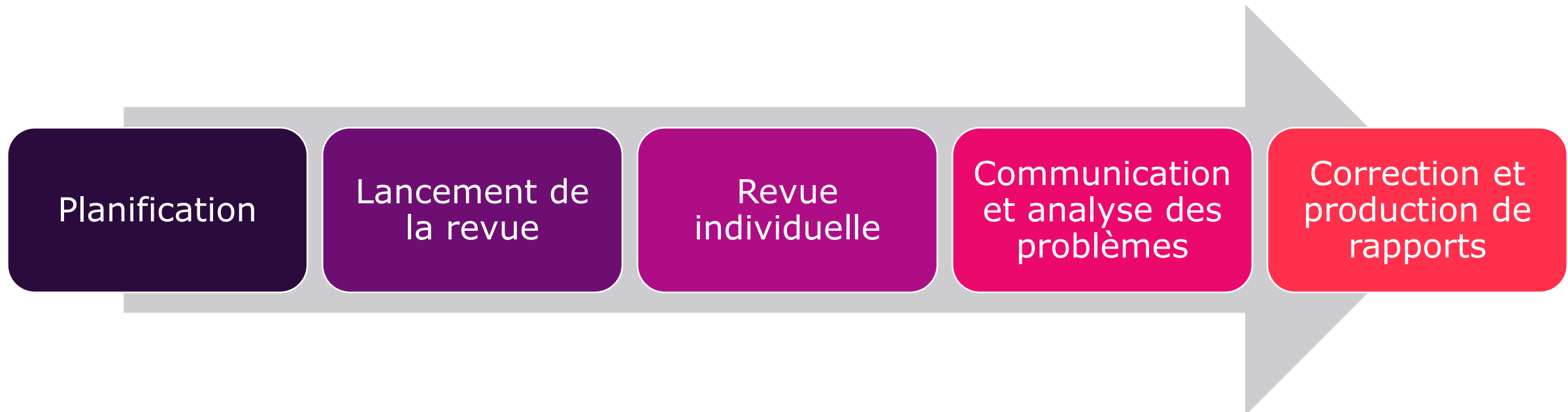
- Le processus des revues peut varier de très informel à très formel.

### ▪ Revue informelle

- Pas de processus défini
- Pas de résultats formels documentés

### ▪ Revue formelle

- Documentation des procédures de conduite de la revue.
- Documentation des résultats de la revue
- Participation de l'équipe



## 3.2.2 Rôles et responsabilités dans une revue formelle

### Auteur

- Crée le produit d'activités revu
- Corrige les défauts du produit d'activités revu (si nécessaire)

### Manager

- Est responsable de la planification de la revue
- Décide de la mise en oeuvre des revues
- Affecte le personnel, le budget et le temps
- Vérifie le rapport coût-efficacité en continu
- Met en œuvre les mesures appropriées en cas de résultats inadéquats

### Facilitateur (ou modérateur)

- Assure le bon déroulement des réunions de revue (quand elles ont lieu)
- Fait la médiation, si nécessaire, entre les différents points de vue
- Est souvent la personne dont dépend le succès de la revue

### Responsable de la revue

- Prend la responsabilité générale de la revue
- Décide qui sera impliqué et organise quand et où elle aura lieu

### Réviseurs

- Il peut s'agir d'experts du domaine, de personnes travaillant sur le projet, d'intervenants ayant un intérêt pour le produit d'activités et/ou de personnes ayant des compétences techniques ou métier spécifiques
- Ils identifient les défauts potentiels du produit d'activités à revoir
- Ils peuvent représenter différentes perspectives (p. ex. testeur, programmeur, utilisateur, opérateur, analyste métier, expert en utilisabilité, etc.).

### Scribe (ou rapporteur)

- Recueille les défauts potentiels découverts au cours de l'activité de revue individuelle
- Enregistre les nouveaux défauts potentiels, les points en suspens et les décisions prises lors de la réunion de revue (durant son déroulement)

## 3.2.3 Types de revue

- 4 types de revues courants :



- Tous les types de revue peuvent aider à la détection des défauts
- Un même produit d'activités peut faire l'objet de plus d'un type de revue
- Le type de revue sélectionné doit être basé sur
  - les besoins du projet
  - les ressources disponibles
  - le type de produit et les risques
  - le domaine d'activité et la culture de l'entreprise
  - ...

# 3.2.3 Types de revue



Tableau récapitulatif

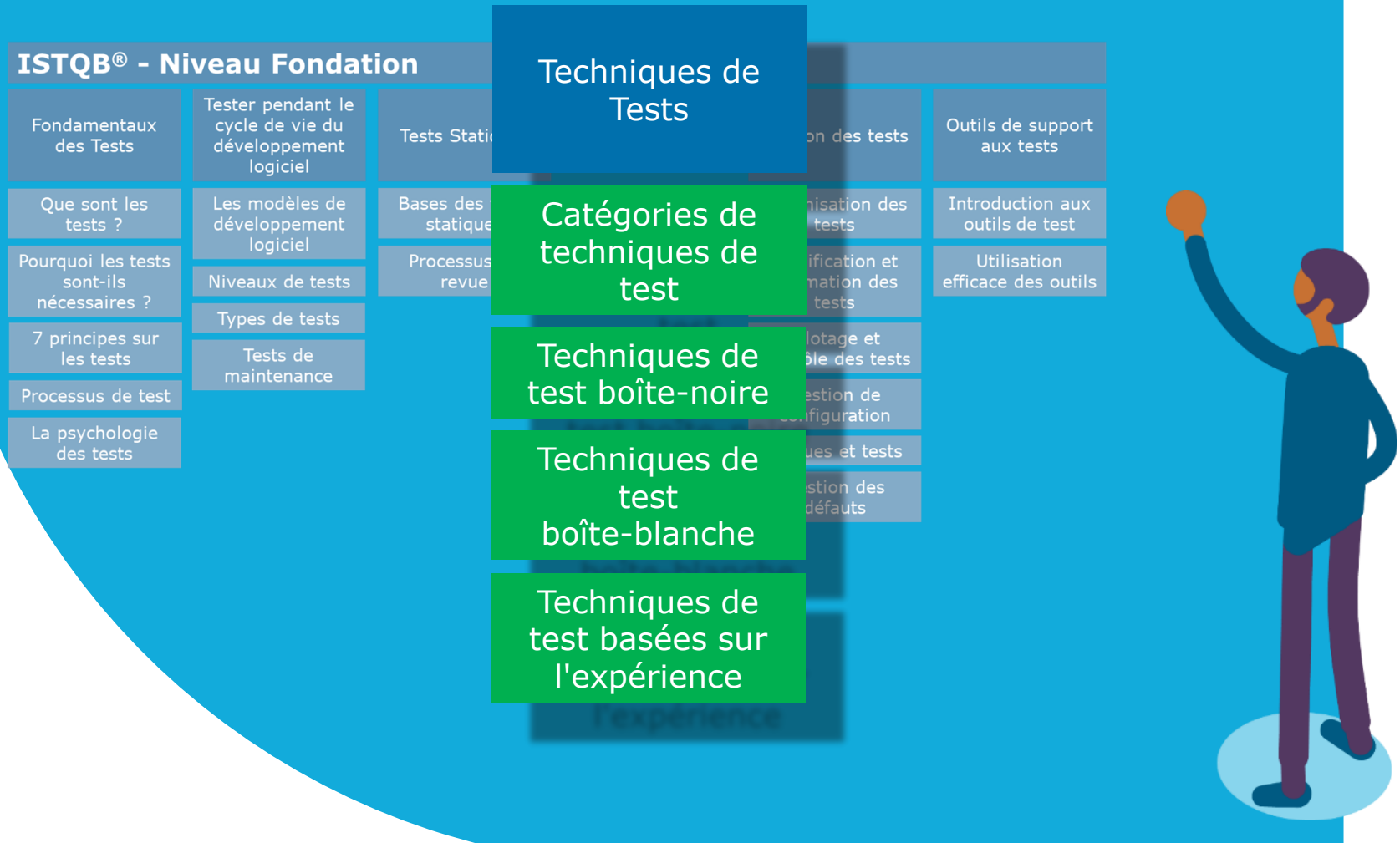
	Revue informelle	Relecture technique	Revue technique	Inspection
Processus formel		X	X	X
Réunion de revue	F	X (dirigée par l’auteur)	F	X (dirigée par un facilitateur formé (pas l’auteur)
Résultats documentés	F	F	X	X
Check-lists	F	F	F	X
Préparation individuelle		F	X	X
Scribe		X	X (pas l’auteur)	X
Réviseurs techniques			X	X
Critères entrées/sorties				X
Métriques				X
Responsable revue				X (pas l’auteur)

X : obligatoire  
F : facultatif

## 3.2.4 Application des techniques de revue

- **Ad hoc**
- **Basée sur les checklists**
- **Scénarios et essais à blanc**
- **Basée sur les rôles**
- **Basée sur la perspective**

# 4 Techniques de test

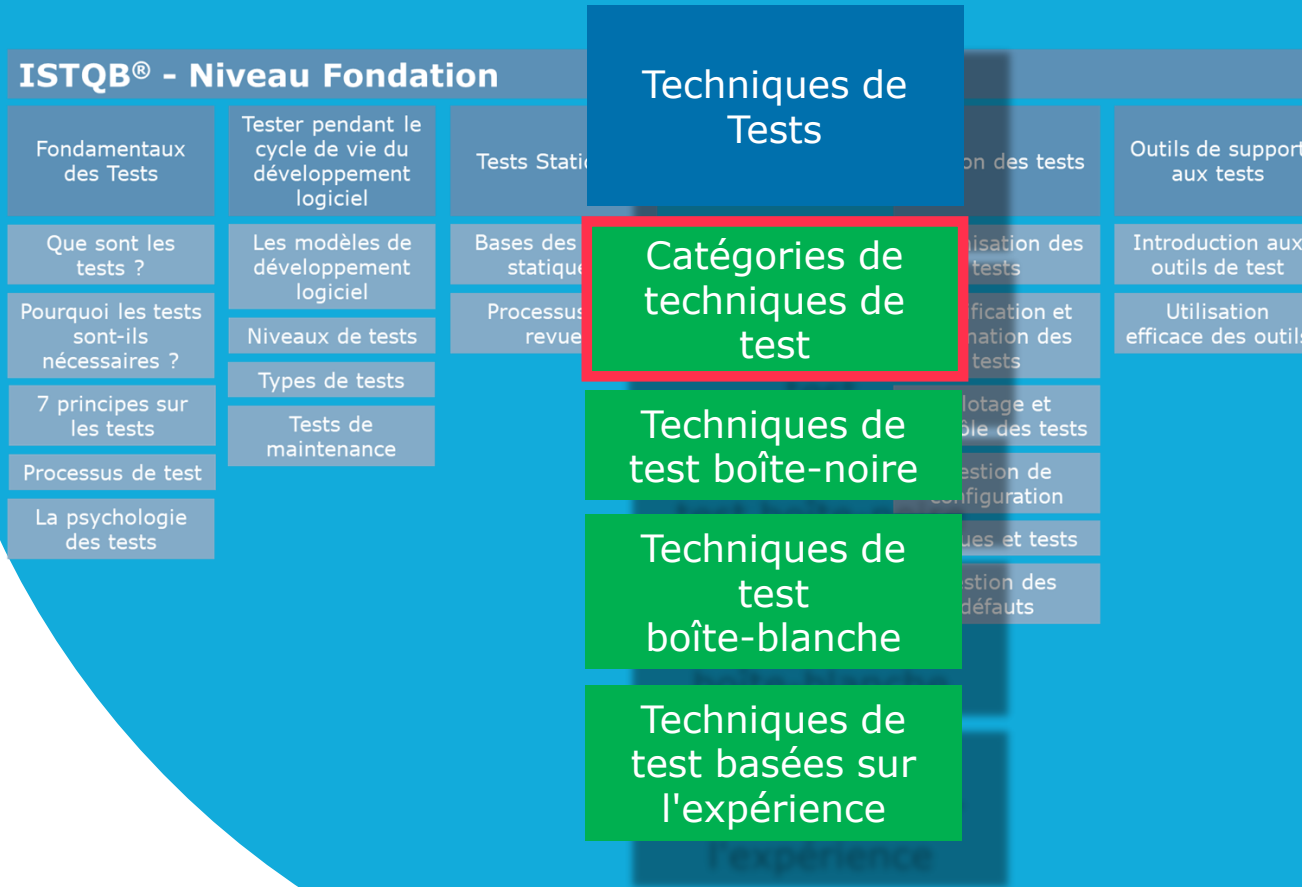


Certification au niveau  
Fondation **version 2018**



# 4 Techniques de test

## 4.1 Catégories de techniques de test



Certification au niveau  
Fondation **version 2018**



## 4.1.2 Catégories de techniques de test et leurs caractéristiques

### Techniques boîte-noire

- Basées sur le comportement
- Basées sur l'analyse de la base de test appropriée
- Applicables aux tests fonctionnels et non-fonctionnels.
- Se concentrent sur les entrées et les sorties de l'objet de test.
- Ne font pas référence à la structure interne de l'objet de test.

### Techniques boîte-blanche

- Basées sur la structure
- Basées sur l'analyse de la base de test appropriée
- Se concentrent sur la structure interne de l'objet de test.
- Se concentrent sur le traitement opéré à l'intérieur de l'objet de test.

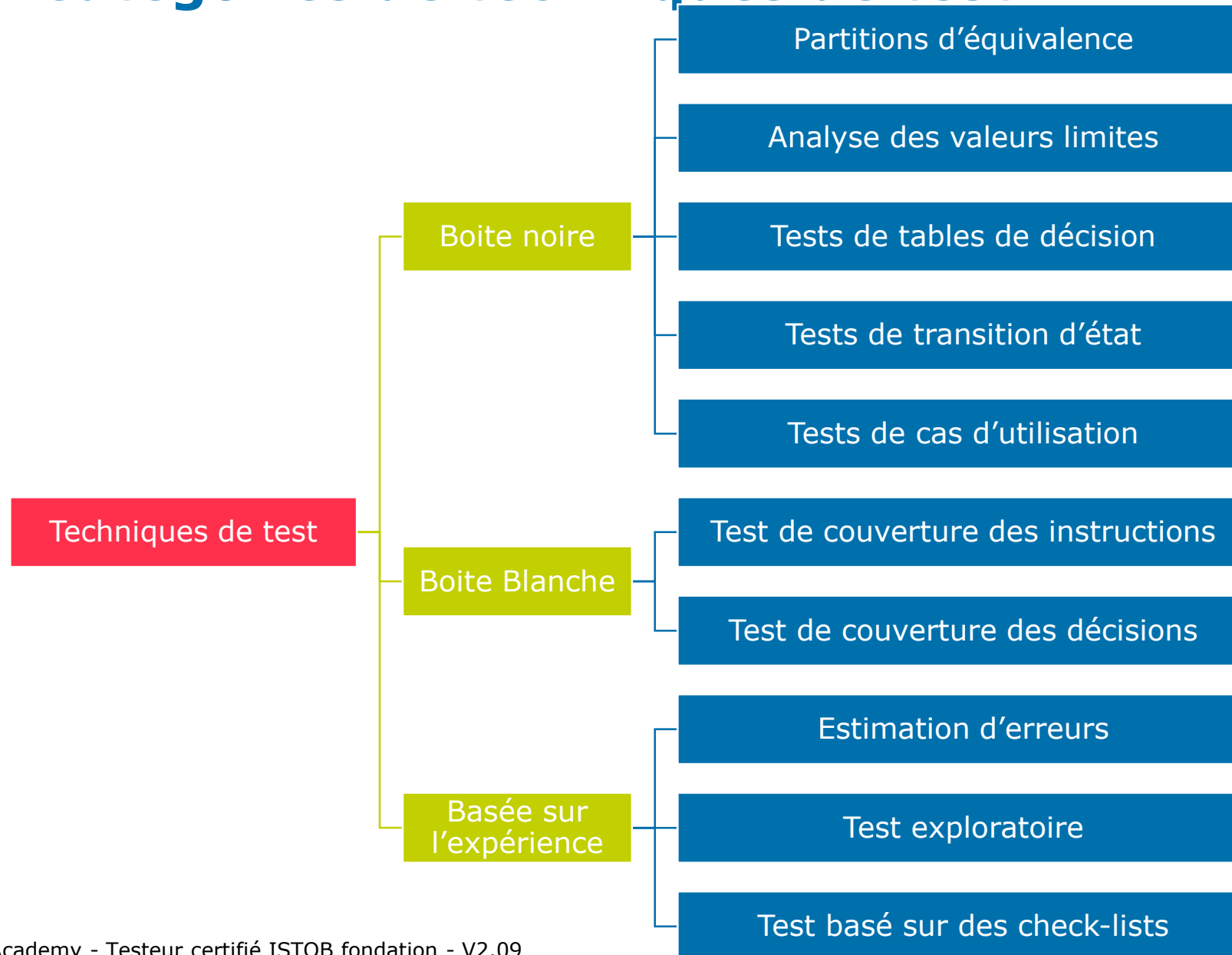
### Techniques basées sur l'expérience

- Tirent parti de l'expérience des développeurs, des testeurs et des utilisateurs pour concevoir, implémenter et exécuter des tests
- Souvent combinées à des techniques de test boîte-noire et boîte blanche.

*Les techniques de test de **boîte-noire** sont aussi appelées **techniques comportementales***

*Les techniques de test de **boîte blanche** sont aussi appelées **techniques structurelles***

## 4.1.2 Catégories de techniques de test



# 4 Techniques de test

## 4.2 Techniques de test boîte-noire



Certification au niveau  
Fondation **version 2018**

## 4.2.1 Partitions d'équivalence

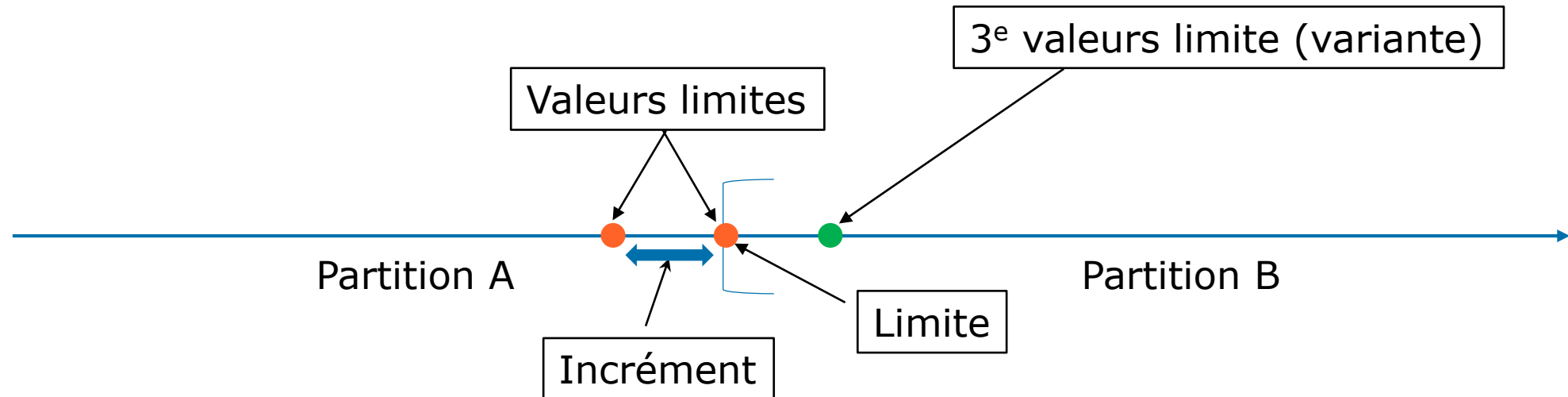
- **Tous les éléments d'une partition donnée sont supposés être traités de la même manière..**
- **Les valeurs valides** sont des valeurs qui doivent être acceptées par le composant ou le système.
- **Les valeurs invalides** sont des valeurs qui doivent être rejetées par le composant ou le système.
- **Les partitions sont disjointes** : chaque valeur doit appartenir à une et une seule partition d'équivalence.
- Lorsque des **partitions d'équivalence invalides** sont utilisées dans des cas de test, elles doivent être **testées individuellement**, **afin de s'assurer que des défaillances** ne sont pas masquées.
- Pour obtenir une couverture de 100% avec cette technique, les cas de test doivent couvrir **toutes les partitions identifiées (y compris les partitions invalides)** en utilisant **une valeur pour chaque partition**.
- Les partitions d'équivalence sont **applicables à tous les niveaux de tests**.

## 4.2.2 Analyse des valeurs limites

L'analyse des valeurs limites est une **extension des partitions d'équivalence**, mais ne peut être utilisée **que lorsque la partition est ordonnée**, composée de données numériques ou séquentielles.

Les valeurs minimale et maximale (ou première et dernière valeurs) d'une partition sont ses valeurs limites.

Exemple :



# 4.2.3 Test de tables de décision

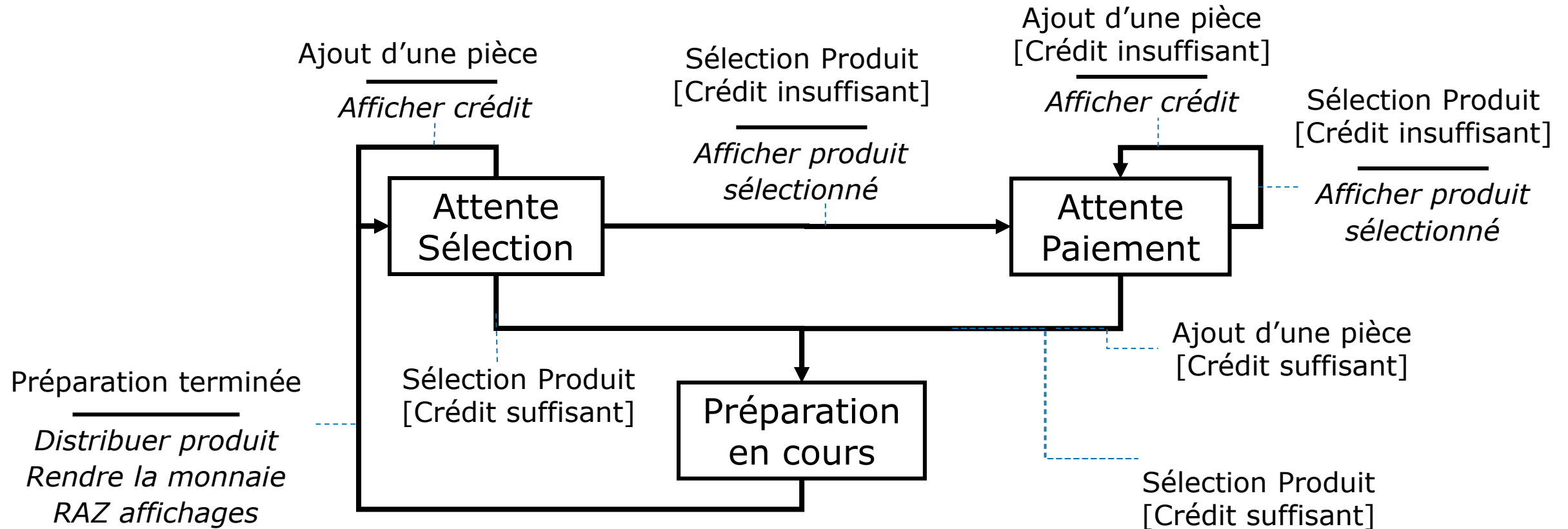
## Exemple d'assurance maladie

2^n cas test, n étant le nombre de condition

Cas de Test	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
C1 Age > 60	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
C2 Santé = bonne	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0
C3 Rembts < 150	0	0	1	1	1	1	0	0	0	0	1	1	1	1	0	0
C4 Age ≤ 50	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
R1 Prime:= 1.200									X	X	X	X	X	X	X	X
R2 Prime:= 950	X	X	X	X	X	X	X	X								
R3 Reduction:= 10%			X			X	X			X	X			X		
R4 Pas de réduction	X	X		X	X			X	X			X	X		X	X
Pas possible										X	X			X	X	
Total	950	950	855	950	950	855	855	950	1200	N/A	N/A	1200	1200	N/A	N/A	1200

## 4.2.4 Test des transitions d'état

### Diagramme de transitions d'état





# 4.2.4 Test des transitions d'état

Tableau de transitions d'état

Evénement		Ajout d'une pièce		Sélection Produit		Préparation terminée
Condition de garde		[Crédit insuffisant]	[Crédit suffisant]	[Crédit insuffisant]	[Crédit suffisant]	
Etat de départ	Attente Sélection	Attente Sélection		Attente Paiement	Préparation en cours	
	Attente Paiement	Attente Paiement	Préparation en cours	Attente Paiement	Préparation en cours	
	Préparation en cours					Attente Sélection

*Les cases à fond blanc représentent l'état suivant*

*Certains événements sous certaines conditions peuvent déclencher une action mais pas de transition*

*Les cases laissées vides représentent les transitions non valides*

## 4.2.4 Test des transitions d'état

### Couverture 0-switch

Cas de Test	1	2	3	4	5	6	7
Etat initial	Attente Sélection Crédit=0	Attente Sélection Crédit=0	Attente Sélection Crédit=0,5	Attente Paiement Crédit=0 Produit à 1€ sélectionné	Attente Paiement Crédit=0,5 Produit à 1€ sélectionné	Attente Paiement Crédit=0,5 Produit à 1€ sélectionné	Attente Paiement Crédit=0,5 Produit à 1€ sélectionné
Evénement	Ajout 50cts	Sélection Produit à 1€	Sélection Produit à 50cts	Ajout 50cts	Ajout 50cts	Sélection Produit à 1€	Sélection Produit à 50cts
Etat suivant	Attente Sélection Crédit=0,5	Attente Paiement Crédit=0 Produit à 1€ sélectionné	Préparation en cours	Attente Paiement Crédit=0,5 Produit à 1€ sélectionné	Préparation en cours	Attente Paiement Crédit=0,5 Produit à 1€ sélectionné	Préparation en cours
Evénement (implicite)			Préparation terminée		Préparation terminée		Préparation terminée
Etat suivant			Attente Sélection Crédit=0		Attente Sélection Crédit=0		Attente Sélection Crédit=0

# 4.2.4 Test des transitions d'état

## Couverture 1-switch

Cas de Test	Etat initial			Evénement	Etat suivant			Evénement	Etat suivant			Evénement (implicite)	Etat suivant	
1	Attente Sélection	Crédit=0		Ajout 50cts	Attente Sélection	Crédit=0,5		Ajout 50cts	Attente Sélection	Crédit=1				
2	Attente Sélection	Crédit=0		Ajout 50cts	Attente Sélection	Crédit=0,5		Sélection Produit à 1€	Attente Paiement	Crédit=0,5	Produit à 1€ sélectionné			
3	Attente Sélection	Crédit=0,5		Ajout 50cts	Attente Sélection	Crédit=1		Sélection Produit à 50cts	Préparation en cours			Préparation terminée	Attente Sélection	Crédit=0
4	Attente Sélection	Crédit=0		Sélection Produit à 1€	Attente Paiement	Crédit=0	Produit à 1€ sélectionné	Ajout 50cts	Attente Paiement	Crédit=0,5	Produit à 1€ sélectionné			
5	Attente Sélection	Crédit=0		Sélection Produit à 50cts	Attente Paiement	Crédit=0	Produit à 50cts sélectionné	Ajout 50cts	Préparation en cours			Préparation terminée	Attente Sélection	Crédit=0
6	Attente Paiement	Crédit=0,5	Produit à 1€ sélectionné	Sélection Produit à 1€	Attente Paiement	Crédit=0,5	Produit à 1€ sélectionné	Sélection Produit à 1€	Attente Paiement	Crédit=0,5	Produit à 1€ sélectionné			
7	Attente Paiement	Crédit=0,5	Produit à 1€ sélectionné	Sélection Produit à 1€	Attente Paiement	Crédit=0,5	Produit à 1€ sélectionné	Ajout 50cts	Préparation en cours			Préparation terminée	Attente Sélection	Crédit=0
8	Préparation en cours			Préparation terminée	Attente Sélection	Crédit=0		Ajout 50cts	Attente Sélection	Crédit=0,5				
9	Préparation en cours			Préparation terminée	Attente Sélection	Crédit=0		Sélection Produit à 1€	Attente Paiement	Crédit=0	Produit à 1€ sélectionné			

## 4.2.5 Tests de cas d'utilisation

Un cas d'utilisation peut inclure des **variations** possibles de son **comportement de base**, y compris un comportement **exceptionnel** ainsi que la **gestion des erreurs**.

Les tests sont conçus pour exercer les comportements définis :

- comportement de base,
- comportement exceptionnel ou alternatif,
- traitement des erreurs.

La couverture peut être mesurée par le pourcentage des comportements testés divisé par le nombre total des comportements du cas d'utilisation.

# 4 Techniques de test

## 4.3 Techniques de test boîte-blanche

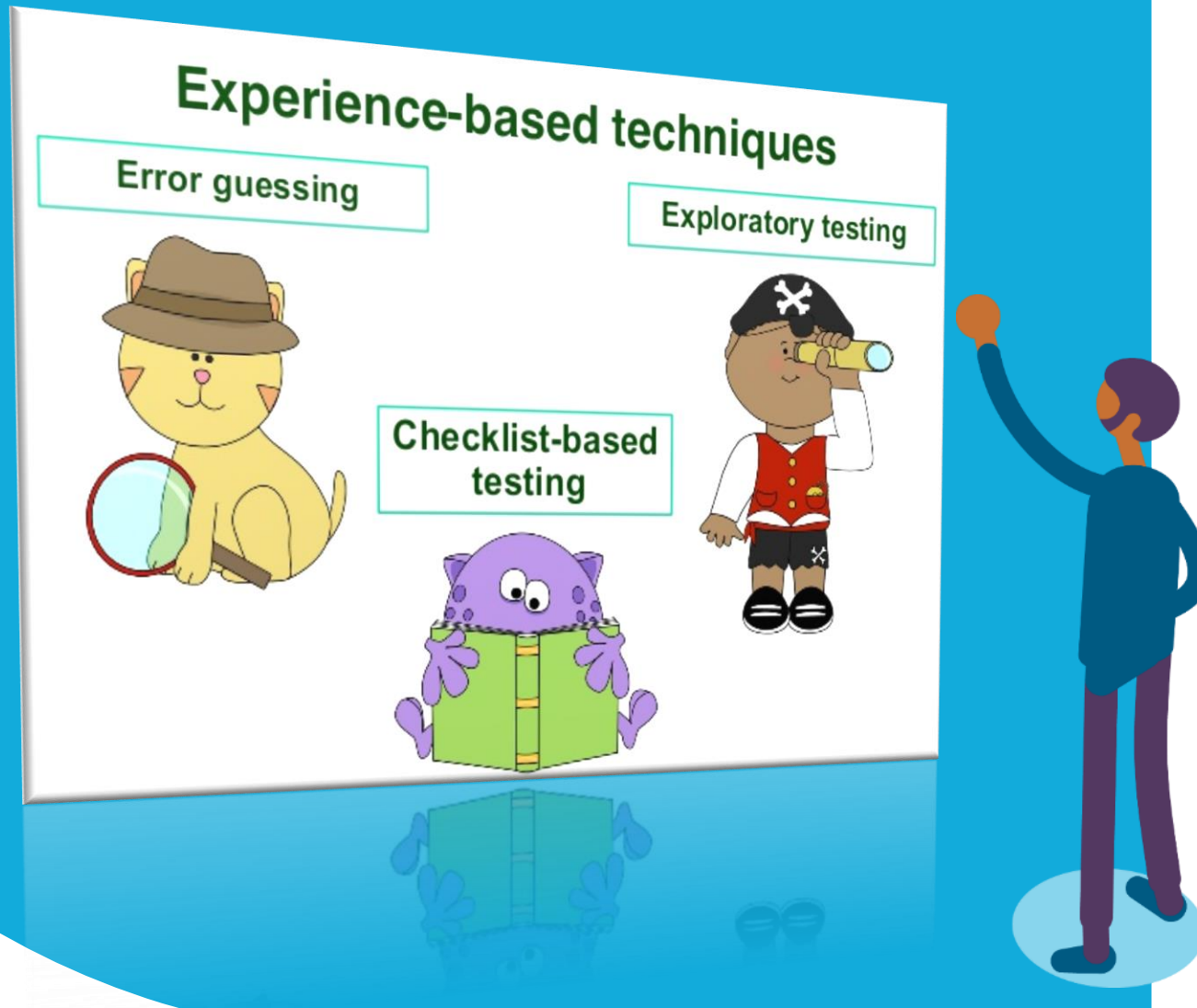


Certification au niveau  
Fondation **version 2018**

## 4.3 Techniques de test boîte-blanche

Couverture des	Objectifs	Couverture	
Instructions	Exercer les instructions du code	Nb instructions exercées / Nb instructions totales	une couverture de 100% des décisions garantit une couverture à 100% des instructions, mais l'inverse n'est pas vrai
Décisions	Exercer les décisions du code (les différents résultats des décisions)	Nb de résultats de décisions exercés / Nb total de résultats de décisions	

## 4.4 Techniques de test basées sur l'expérience



Certification au niveau  
Fondation **version 2018**

# 5 Gestion des tests

## 5.1 Organisation des tests



Certification au niveau  
Fondation **version 2018**



## 5.1.1 Indépendance des tests

Les degrés d'indépendance dans les tests comprennent les niveaux suivants (allant d'un niveau d'indépendance faible à un niveau élevé) :

- **Pas de testeurs indépendants** : les développeurs testent leur propre code
- Des développeurs ou des **testeurs indépendants au sein des équipes** de développement ou de l'équipe de projet ; il peut s'agir de développeurs qui testent les produits de leurs collègues
- Une **équipe de test indépendante** ou un groupe de testeurs au sein de l'organisation, rapportant à la direction du projet ou à la direction générale
- Des **testeurs indépendants appartenant à l'organisation** métier ou à la communauté d'utilisateurs, ou spécialisés dans des types de tests spécifiques tels que l'utilisabilité, la sécurité, la performance, la conformité réglementaire ou la portabilité
- Des **testeurs indépendants externes** à l'organisation, travaillant soit sur site (**insourcing**) soit hors site (**outsourcing**)

## 5.1.1 Indépendance des tests

### Avantages de l'indépendance :

- Les testeurs indépendants sont susceptibles de détecter des **types de défaillances différents** par rapport aux développeurs en raison de leurs **connaissances propres**, de leurs **approches techniques** et de **biais différents**
- Un testeur indépendant peut **vérifier, contester ou réfuter les hypothèses** formulées par les parties prenantes au cours de la spécification et de l'implémentation du système

### Inconvénients de l'indépendance :

- Les testeurs peuvent être **isolés de l'équipe de développement**, ce qui peut entraîner
  - un **manque de collaboration**,
  - des **retards dans la communication** des retours d'information à l'équipe de développement
  - ou une **relation conflictuelle** avec l'équipe de développement
- Les **développeurs** peuvent perdre le sens des **responsabilités vis-à-vis de la qualité**
- Les testeurs indépendants peuvent être considérés comme un **goulot d'étranglement** ou être rendus responsables des retards dans la sortie de la version
- Les testeurs indépendants peuvent **ne pas disposer de certaines informations importantes** (par exemple, sur l'objet de test)

## 5.1.2 Tâches d'un Test Manager et d'un testeur

### Tâches habituelles du Test Manager

- Développer ou revoir une **politique de test** et une stratégie de test pour l'organisation
- **Planifier les activités de test :**
- **Lancer et suivre l'avancement des activités de test et vérifier le statut des critères de sortie** en fonction des résultats des tests
- Préparer et fournir les **rapports d'avancement des tests et de synthèse** des tests
- Adapter la planification en fonction des résultats et de l'avancement des tests et prendre toutes les mesures nécessaires au **contrôle des tests**
- Aider la mise en place du système de **gestion des défauts** et à la **gestion adéquate de la configuration des testware**
- Introduire les **métriques** appropriées pour mesurer l'avancement des tests et évaluer la qualité des tests et du produit

## 5.1.2 Tâches d'un Test Manager et d'un testeur

### Tâches habituelles du Test Manager (suite)

- Accompagner la sélection et la mise en œuvre des **outils** pour soutenir le processus de test, **allouer du temps** et des ressources **pour des projets pilotes** et apporter un **accompagnement** continu à l'utilisation du ou des outils
- Décider de l'implémentation du ou des **environnements de test**
- **Promouvoir et soutenir les testeurs**, l'équipe de test et le métier du test au sein de l'organisation
- **Développer les compétences et les carrières des testeurs** (par exemple, par le biais de plans de formation, d'évaluations de performance, de coaching, etc.).

## 5.1.2 Tâches d'un Test Manager et d'un testeur

### Tâches habituelles des testeurs

- **Revoir et contribuer aux plans de test**
- Analyser, revoir et évaluer les **exigences**, les User Stories et les critères d'acceptation, les spécifications et les modèles (c.-à-d. les bases de test) vis à vis de leur **testabilité**
- Identifier et documenter les **conditions de test**, et saisir la **traçabilité** entre les cas de test, les conditions de test et les bases de test.
- **Concevoir, configurer et vérifier le ou les environnement(s) de test**, souvent en se coordonnant avec l'administration système et réseau
- **Concevoir et implémenter** les cas de test et les procédures de test
- **Préparer et acquérir des données de test**
- Créer le **planning détaillé d'exécution des tests**
- **Exécuter** des tests, évaluer les résultats et documenter les écarts par rapport aux résultats attendus
- **Utiliser les outils** appropriés pour faciliter le processus de test
- **Automatiser** les tests selon les besoins (peut être porté par un développeur ou un expert en automatisation des tests)
- Évaluer les **caractéristiques non-fonctionnelles** telles que la performance, la fiabilité, l'utilisabilité, la sécurité, la compatibilité et la portabilité
- Revoir des **tests développés par d'autres**

## 5.1.2 Tâches d'un Test Manager et d'un testeur

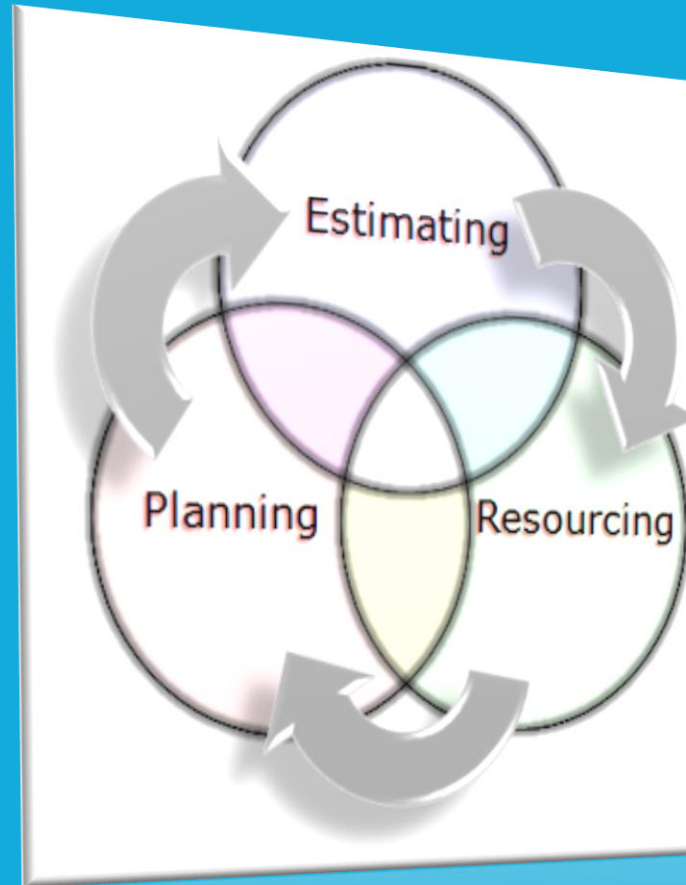
Les personnes qui travaillent à l'analyse des tests, à la conception des tests, à des types de tests spécifiques ou à l'automatisation des tests peuvent être des **spécialistes** dans ces rôles.

En fonction des risques produit et projet, et du modèle de cycle de vie du développement logiciel choisi, **différentes personnes peuvent assumer le rôle de testeur** à différents niveaux de test.

- Par exemple, au niveau du test des **composants** et du test d'intégration des composants, le rôle de testeur est **souvent** porté par des **développeurs**.
- Au niveau du test **d'acceptation**, le rôle de testeur est **souvent** porté par des analystes **métier**, des experts du **domaine** et des **utilisateurs**.
- Aux niveaux du test du système et du test d'intégration de systèmes, le rôle de testeur est **souvent** porté par une **équipe de test indépendante**.
- Au niveau des tests d'acceptation **opérationnelle**, le rôle de testeur est **souvent** porté par **l'équipe d'exploitation et/ou d'administration**.

# 5 Gestion des tests

## 5.2 Planification et estimation des tests



**CFTL**  
Comité Français  
des Tests Logiciels

**ISTQB®**  
International Software  
Testing Qualifications Board

Certification au niveau  
Fondation **version 2018**

## 5.2.1 Objet et contenu d'un plan de test

Un plan de test décrit les activités de test pour des projets de développement et de maintenance.

La planification est influencée par :

- **la politique et la stratégie de test de l'organisation,**
- **les cycles de vie du développement et les méthodes utilisées (voir section 2.1),**
- **l'étendue des tests (le périmètre),**
- **les objectifs,**
- **les risques,**
- **les contraintes éventuelles,**
- **la criticité,**
- **la testabilité,**
- **et la disponibilité des ressources.**



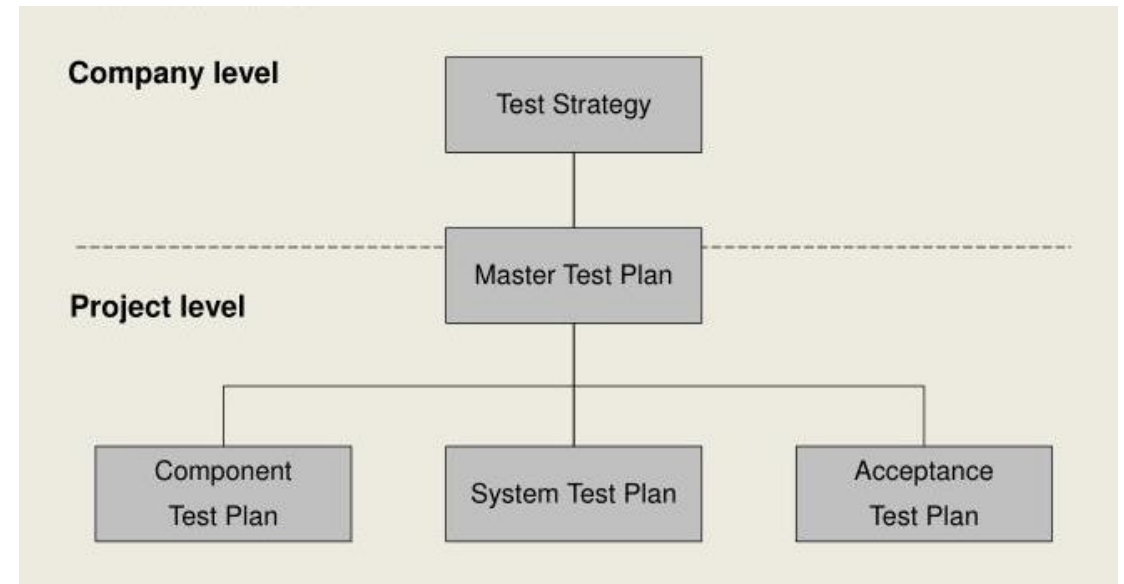
## 5.2.1 Objet et contenu d'un plan de test

Au fur et à mesure que le projet progresse, davantage d'informations deviennent disponibles et plus de détails peuvent être inclus dans le plan de test.

La planification des tests est une **activité continue** et s'effectue **tout au long du cycle de vie** du produit (y/c la phase de maintenance).

En fonction des retours d'information sur les activités de test, **les risques peuvent évoluer** et la **planification** être **ajustée**.

La planification peut être documentée dans un **plan de test maître** et dans des plans de test séparés pour des **niveaux de test**, ou pour des **types de test** séparés, tels que les tests d'utilisabilité et les tests de performance.



## 5.2.1 Objet et contenu d'un plan de test

### Activités de planification des tests

- Déterminer le **périmètre, les objectifs et les risques** des tests
- Définir **l'approche** générale des tests
- Intégrer et coordonner les activités de test dans les activités du cycle de vie du logiciel
- **Prendre des décisions** sur ce qu'il faut tester, les **personnes et les autres ressources** nécessaires pour effectuer les diverses activités de test et la façon dont les activités de test seront effectuées
- **Planifier les activités** d'analyse, de conception, d'implémentation, d'exécution et d'évaluation des tests,
- Sélectionner les **métriques** pour le pilotage et le contrôle des tests
- **Budgéter** les activités de test
- Déterminer le niveau de détail et la structure de la **documentation** des

## 5.2.2 Stratégie de test et approche de test

Une stratégie de test fournit une **description générale du processus de test**, généralement au niveau du **produit** ou de **l'organisation**. Les types les plus courants de stratégies de test sont les suivants :

- **Analytique** : basé sur **l'analyse** d'un facteur (p. ex., exigences ou risques)
- **Basée sur des modèles**
- **Méthodique** : repose sur l'utilisation systématique d'un ensemble prédéfini de tests ou de conditions de test
- **Conforme à un processus (ou à une norme)** : basés sur des règles et normes externes, telles que des normes spécifiques à l'industrie, ou tout processus ou norme imposée à l'organisation ou par l'organisation.
- **Dirigée (ou consultative)** : dicté par les recommandations, conseils ou instructions de parties prenantes, experts du domaine métier ou experts techniques
- **Anti-régressions** : vise à éviter la régression en réutilisant des testware existants
- **Réactive** : les tests sont conçus, implémentés et exécutés immédiatement en réaction aux événements se produisant pendant l'exécution des tests antérieurs.

## 5.2.2 Stratégie de test et approche de test

Une stratégie de test pertinente est souvent créée en combinant plusieurs de ces types de stratégies de test.

- Par exemple, le test basé sur les risques (une stratégie **analytique**) peut être combiné avec des tests exploratoires (une stratégie **réactive**)

Alors que la stratégie de test fournit une description générale du processus de test, **l'approche de test ajuste la stratégie de test pour un projet ou une version particulière.**

L'approche de test est le **point de départ** :

- pour sélectionner les **niveaux, types et techniques de test**,
- pour définir les critères **d'entrée et de sortie**

L'ajustement de la stratégie est basé sur les décisions prises en tenant compte de facteurs tels que les risques, la sécurité, les ressources et les compétences disponibles, la technologie, la nature du système (p. ex., construit sur mesure versus sur étagère), les objectifs des tests et les réglementations.

## 5.2.3 Critères d'entrée et de sortie (Définition du prêt et définition du terminé)

*Définition du prêt = Definition of Ready (**DoR**)*

*Définition du terminé = Definition of Done (**DoD**)*

**Objet** : définir quand une activité de test doit commencer (**DoR**) et se terminer (**DoD**)

**Finalité** : contrôle efficace sur la qualité du logiciel et des tests

Même si les critères de sortie ne sont pas satisfaits, il est courant que les activités de test soient réduites en raison du budget dépensé, du temps prévu et/ou de la pression pour mettre le produit sur le marché.

Il peut être acceptable de mettre fin aux tests dans de telles circonstances, si les parties prenantes du projet et les responsables métier ont revu et **accepté le risque** de mettre en production sans autres tests.

## 5.2.4 Calendrier d'exécution des tests

Une fois que les différents cas de test et procédures de test sont produits (avec des procédures de test potentiellement automatisées) et assemblés en suites de test,

- les suites de test peuvent être organisées selon un calendrier d'exécution des tests qui définit **l'ordre dans lequel elles doivent être exécutées**.

Idéalement, les cas de test devraient être exécutés en fonction de leur **niveau de priorité**, généralement en exécutant en premier les cas de test ayant la priorité la plus élevée.

Cependant, si les cas de test ou les caractéristiques testées ont des **dépendances**, il faut les ordonner de façon appropriée,

- Si un cas de test avec une priorité plus élevée dépend d'un cas de test avec une priorité plus faible, le cas de test de priorité plus faible doit être exécuté en premier.

Les tests de confirmation et de régression doivent également être priorisés, en fonction de l'importance d'une remontée d'information rapide sur les changements réalisés, mais là encore, des dépendances peuvent s'appliquer.

## 5.2.5 Facteurs influençant l'effort de test

### Caractéristiques du produit

- Les risques associés au produit
- La taille du produit
- La complexité du domaine métier du produit
- Les exigences relatives aux caractéristiques de qualité (p. ex., sécurité, fiabilité)
- Les exigences en matière de conformité juridique et réglementaire

### Caractéristiques liées aux personnes

- Les compétences et l'expérience des personnes impliquées, en particulier avec des projets et des produits similaires (par exemple, connaissance du domaine)
- La cohésion et le leadership dans l'équipe

### Caractéristiques du processus de développement

- La pression des délais
- La stabilité et la maturité de l'organisation
- Le modèle de développement utilisé
- La qualité des bases de test
- L'approche de test
- Les outils utilisés
- Le niveau de détail requis pour la documentation des tests

### Résultats des tests

- Le nombre et la sévérité des défauts trouvés
- La quantité nécessaire de retest ou rejeu

## 5.2.6 Techniques d'estimation des tests

### Techniques les plus couramment utilisées :

- La technique basée sur les **métriques** : estimer l'effort de test sur la base de métriques d'anciens projets similaires ou sur la base de valeurs représentatives
  - En séquentiel, la productivité de chacune des phases (selon le nombre d'exigences ou de cas de test) est enregistrée sous forme **d'abaques** pour estimer les projets futurs de nature similaire ; de même, les volumes de défauts et le temps nécessaire pour les éliminer sont recueillis et enregistrés
  - En Agile, les **burndown charts** enregistrés pour une itération sont utilisés pour déterminer la quantité de travail que l'équipe peut faire dans la prochaine itération
- La technique basée sur **l'expertise** : estimation de l'effort de test sur la base de l'expérience des responsables des tâches de test ou par des experts
  - Dans les projets séquentiels, la technique d'estimation **Wideband Delphi** est un exemple d'approche basée sur l'expertise dans laquelle des groupes d'experts fournissent des estimations basées sur leur expérience
  - En Agile, au cours d'un **planning poker**, les membres de l'équipe estiment l'effort pour livrer une fonctionnalité en se basant sur leur expérience



# 5 Gestion des tests

## 5.3 Pilotage et contrôle des tests



Certification au niveau  
Fondation **version 2018**

## 5.3 Pilotage et contrôle des tests

### Pilotage des tests :

- recueillir (manuellement ou automatiquement) des informations sur les activités de test
- fournir un retour et de la visibilité sur ces activités (reporting)
- évaluer l'avancement du test vis-à-vis des critères de sortie

### Contrôle des tests :

- mesures correctives ou d'orientation prises à la suite de l'information et des métriques recueillies et (éventuellement) rapportées.  
Les actions peuvent couvrir **toutes les activités de test** et peuvent affecter **toute autre activité** du cycle de vie du logiciel. Exemples :
  - Re-prioriser les tests lorsqu'un risque identifié se produit (p. ex. logiciel livré en retard)
  - Modifier le calendrier des tests en raison de la disponibilité ou de l'indisponibilité d'un environnement de test ou d'autres ressources
  - Réévaluer si un élément de test répond à un critère d'entrée ou de sortie à cause d'une modification

## 5.3.2 Buts, contenu et destinataires des rapports de test

### But :

- **Synthétiser et communiquer** les informations sur l'activité de test, à la fois **pendant et à la fin** d'une activité de test (par exemple, un niveau de test)

Le rapport de test préparé **pendant** une activité de test peut être appelé **rapport d'avancement** de test,

Le rapport de test préparé **à la fin** d'une activité de test (lorsque les **critères de sortie** sont atteints) peut être appelé **rapport de synthèse** de test.

## 5.3.2 Buts, contenu et destinataires des rapports de test

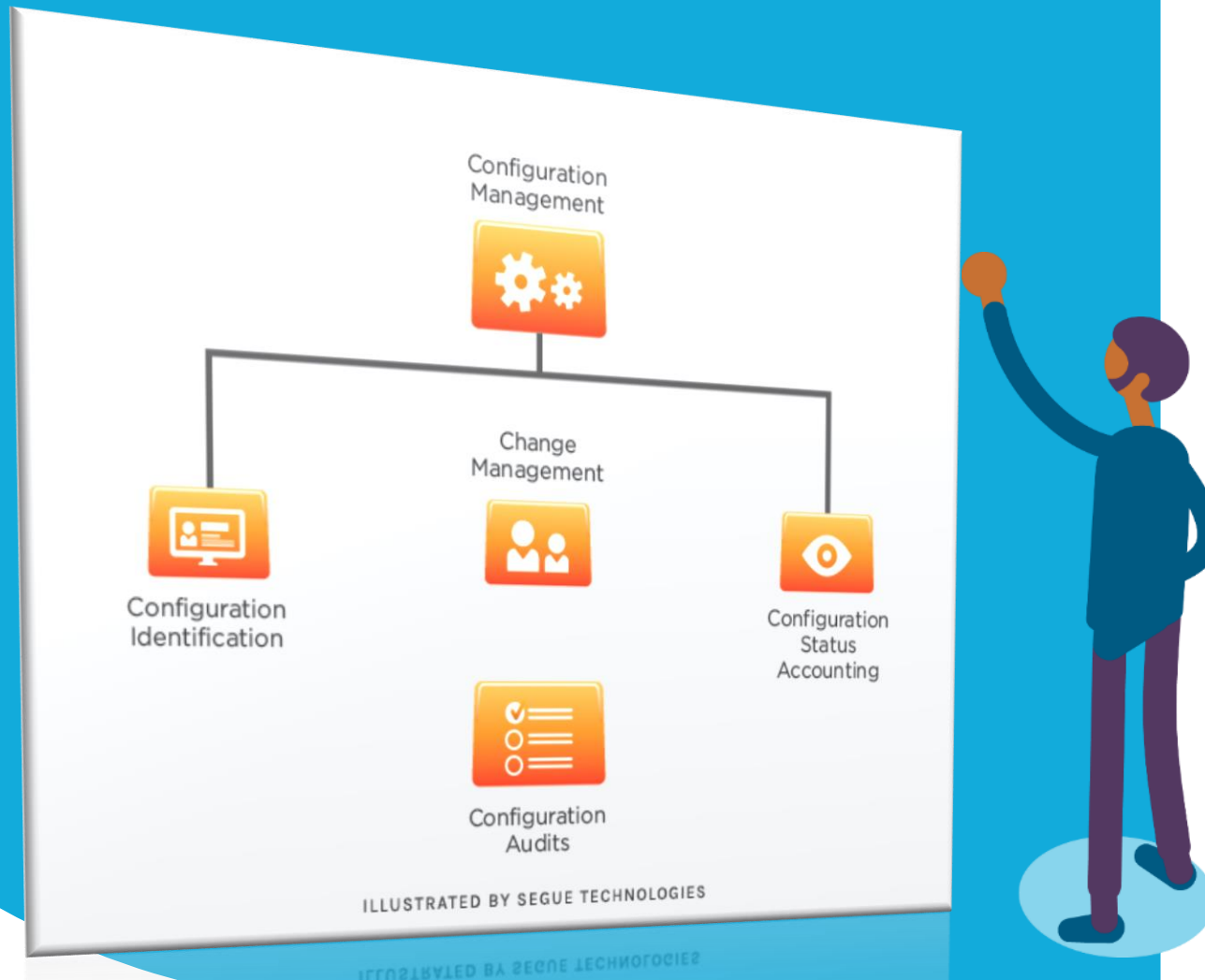
### Contenu :

Les rapports d'avancement et de synthèse de test peuvent comprendre les éléments suivants :

- Informations sur le **déroulement de la période** de test et activités prévues pour la prochaine période
- **Récapitulatif des tests réalisés**
- **Écarts** par rapport au plan, y compris les écarts par rapport au calendrier, à la durée ou à l'effort consacré aux activités de test
- Facteurs ayant **freiné** ou freinant l'avancement
- **Statut des tests et de la qualité du produit** par rapport aux **critères de sortie** ou à la définition du terminé
- **Métriques** sur les défauts, les cas de test, la couverture de test, l'avancement de l'activité et la consommation de ressources.
- **Risques** résiduels (voir section 5.5)
- **Produits d'activités** de test réutilisables

# 5 Gestion des tests

## 5.4 Gestion de configuration



Certification au niveau  
Fondation **version 2018**

## 5.4 Gestion de configuration

Le but de la gestion de la configuration est d'établir et de **maintenir l'intégrité** du **système**, du **testware** et de leurs **relations mutuelles tout au long du cycle de vie** du **projet** et du **produit**.

Pour supporter les tests, la gestion de la configuration doit s'assurer de ce qui suit :

- Tous les **éléments de test** sont **identifiés** de façon unique, **versionnés**, **suivis pour les changements** et **reliés les uns aux autres**
- Tous les **éléments du testware** sont identifiés de manière unique, versionnés, suivis pour les changements, liés les uns aux autres **et liés aux versions des éléments de test** afin que la **traçabilité** puisse être maintenue tout au long du processus de test
- Tous les documents et logiciels identifiés sont **référéncés sans ambiguïté** dans la documentation de test

**Au cours de la planification des tests**, les procédures de gestion de la configuration et l'infrastructure (outils) devraient être identifiées et implémentées.

# 5 Gestion des tests

## 5.5 Risques et tests



Certification au niveau  
Fondation **version 2018**

## 5.5.1 Définition du risque

### Définition

Un risque peut être défini comme la possibilité qu'un événement, un danger, une menace ou une situation arrive, et que les conséquences indésirables qui en découlent constituent un problème potentiel.

Le **niveau de risque** sera déterminé par la **probabilité** qu'un **événement redouté** arrive et par **l'impact** de ce dernier (les dommages résultant de cet événement).

$$\text{RISQUE} = \text{PROBABILITE} * \text{IMPACT}$$



## 5.5.2 Risques produit et risques projet

### Risque produit

Le risque produit implique la possibilité qu'un **produit** d'activités (par exemple, une spécification, un composant, un système ou un test) puisse **ne pas satisfaire les besoins de ses utilisateurs et/ou parties prenantes**.

Lorsque les risques produit sont **associés à des caractéristiques de qualité** spécifiques d'un produit (adéquation fonctionnelle, fiabilité, performance, utilisabilité, sécurité, compatibilité, maintenabilité et portabilité), les risques produit sont également appelés **risques qualité**.

### Risque projet

Le risque projet implique des situations qui, si elles se produisent, peuvent avoir un effet négatif sur **la capacité d'un projet à atteindre ses objectifs**.

**Problèmes liés au projet**

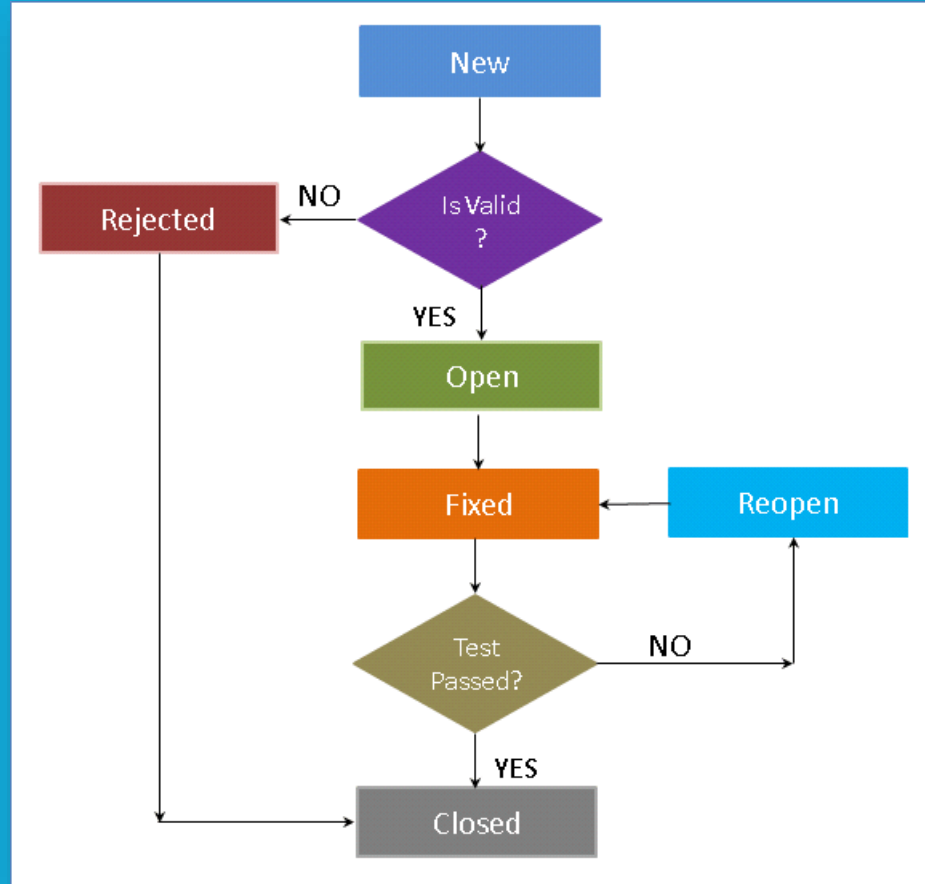
**Problèmes organisationnels**

**Problèmes politiques**

**Problèmes liés aux fournisseurs**

# 5 Gestion des tests

## 5.6 Gestion des défauts



Certification au niveau  
Fondation **version 2018**

## 5.6 Gestion des défauts

Un rapport de défaut comprend généralement les éléments suivants :

- **Un identifiant**
- **Un titre et un bref résumé du défaut signalé**
- La **date** du rapport de défaut, l'organisation émettrice et l'**auteur**
- **L'identification de l'élément de test** et de **l'environnement**
- La ou les phases du cycle de développement au cours desquelles le défaut a été observé
- Une description du défaut pour permettre sa reproduction et sa résolution (logs, captures)
- Les **résultats attendus et obtenus**
- La **portée ou le degré d'impact** (sévérité) du défaut pour les parties prenantes concernées
- L'urgence/priorité de la correction
- Le statut du défaut (p. ex. ouvert, en double, en attente de correction, en attente de test de confirmation, ré-ouvert, fermé)
- Les conclusions, recommandations et approbations
- Les enjeux généraux, tels que les autres domaines qui peuvent être affectés
- L'historique des modifications, notamment la séquence des mesures prises pour isoler, réparer le défaut et confirmer qu'il est corrigé
- Les références, y compris le cas de test qui a révélé le problème

# 6 Outils de support aux tests

## 6.1 Introduction aux outils de test



**CFTL**  
Comité Français  
des Tests Logiciels

**ISTQB**  
International Software  
Testing Qualifications Board

Certification au niveau  
Fondation **version 2018**

## 6.1.1 Classification des outils de test

### **Outils pour la gestion des tests et du testware**

- Outils de gestion des tests et outils de gestion du cycle de vie des applications
- Outils de gestion des exigences (p. ex. traçabilité des objets de test)
- Outils de gestion des défauts
- **Outils de gestion de la configuration**
- **Outils d'intégration continue (D)**

### **Outils pour les tests statiques**

- Outils de support aux revues
- Outils d'analyse statique (D)

### **Outils pour la conception et l'implémentation des tests**

- Outils de Model-Based Testing
- Outils de préparation des données de test
- Outils d'ATDD ou de BDD
- Outils de TDD (D)

### **Outils pour l'exécution et l'enregistrement des résultats (les logs de tests)**

- Outils d'exécution de tests (p. ex., pour exécuter des tests de régression)
- **Outils de couverture (p. ex., couverture des exigences, couverture du code (D))**
- Harnais de test (D)
- Framework de tests unitaires (D)

## 6.1.1 Classification des outils de test

### Outils pour la mesure de la performance et l'analyse dynamique

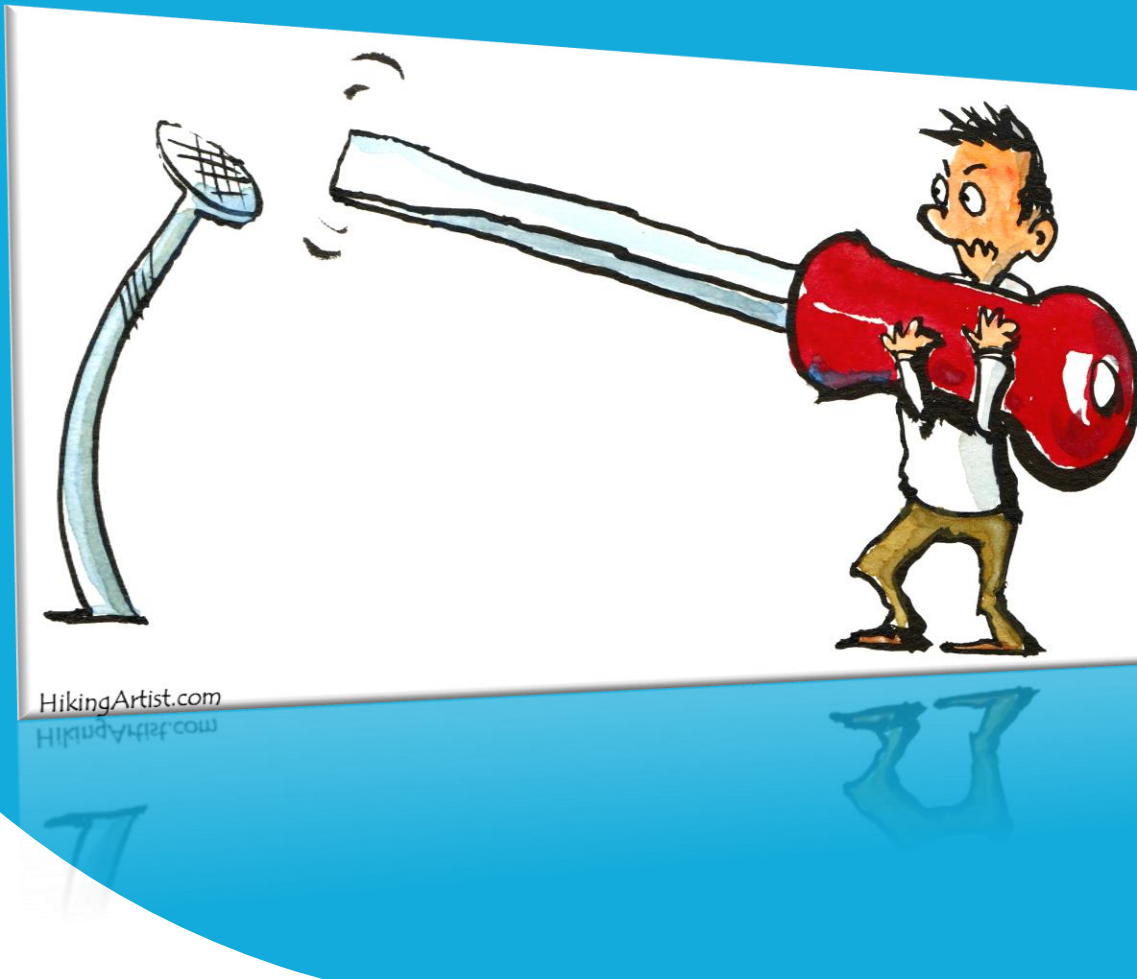
- Outils de test de performance
- **Outils de monitoring**
- Outils d'analyse dynamique (D)

### Outils pour des besoins de test spécifiques

- Évaluation de la qualité des données
- **Conversion et migration des données**
- Tests d'utilisabilité
- Tests d'accessibilité
- Tests de localisation
- Tests de sécurité
- Tests de portabilité (p. ex. tests de logiciels sur plusieurs plates-formes prises en charge)

# 6 Outils de support aux tests

## 6.2 Utilisation efficace des outils



**CFTL**  
Comité Français  
des Tests Logiciels

**ISTQB**  
International Software  
Testing Qualifications Board

Certification au niveau  
Fondation **version 2018**



## 6.2.1 Principes de base pour la sélection des outils

- Évaluer la **maturité de l'organisation**, de ses forces et de ses faiblesses
- Identifier ce qu'un outil pourrait **améliorer**
- Sélectionner un outil **compatible** avec
  - les technologies utilisées par les objets de test
  - les outils de build et d'intégration continue déjà utilisés au sein de l'organisation
- Évaluation de l'outil par rapport à des **exigences précises et à des critères objectifs**
- Vérifier si l'outil est disponible pour une **période d'essai** gratuit (et pour combien de temps)
- **Évaluation du fournisseur** (y compris la formation, le support et les aspects commerciaux) ou du support pour les outils non commerciaux (p. ex., open source)
- Évaluer les besoins de **formation**
- Identifier les besoins internes de **coaching et de support à l'utilisation**
- Prendre en compte les avantages et des inconvénients des différents **modèles de licence** (p. ex., commercial ou open source)
- Estimation d'un **rapport coût-bénéfice** basé sur une analyse de rentabilité concrète
- **Faire un POC** (Proof-of-Concept) pour
  - déterminer si **l'outil fonctionne efficacement avec le logiciel à tester et au sein de l'infrastructure actuelle**
  - ou, si nécessaire, identifier les **changements nécessaires à cette infrastructure** pour utiliser efficacement l'outil.



## 6.2.3 Facteurs de succès pour les outils

- Déployer l'outil au reste de l'organisation de façon **incrémentale**
- **Adapter et améliorer les processus** en fonction de l'utilisation de l'outil
- Apporter de la **formation, du coaching et du conseil** aux utilisateurs d'outils
- Définir des **recommandations d'usage** de l'outil (p. ex. normes internes pour l'automatisation)
- Mettre en place un moyen de collecte d'informations sur **l'utilisation réelle de l'outil**
- **Suivre l'utilisation de l'outil et ses bénéfices**
- **Accompagner** les utilisateurs
- Recueillir les **leçons tirées de l'expérience** de **tous** les utilisateurs
- S'assurer que l'outil est **intégré au niveau technique et organisationnel** dans le **cycle de vie du développement du logiciel**, ce qui peut impliquer des organisations différentes, responsables de l'exploitation et/ou des fournisseurs tiers.

# And now...



## About Sogeti

Sogeti is a leading provider of technology and engineering services. Sogeti delivers solutions that enable digital transformation and offers cutting-edge expertise in Cloud, Cybersecurity, Digital Manufacturing, Digital Assurance & Testing, and emerging technologies. Sogeti combines agility and speed of implementation with strong technology supplier partnerships, world class methodologies and its global delivery model, Rightshore®. Sogeti brings together more than 25,000 professionals in 15 countries, based in over 100 locations in Europe, USA and India. Sogeti is a wholly-owned subsidiary of Capgemini SE, listed on the Paris Stock Exchange.

Learn more about us at  
[www.sogeti.com](http://www.sogeti.com)



This message contains information that may be privileged or confidential and is the property of the Capgemini Group.

Copyright© 2018 Sogeti. All rights reserved.