

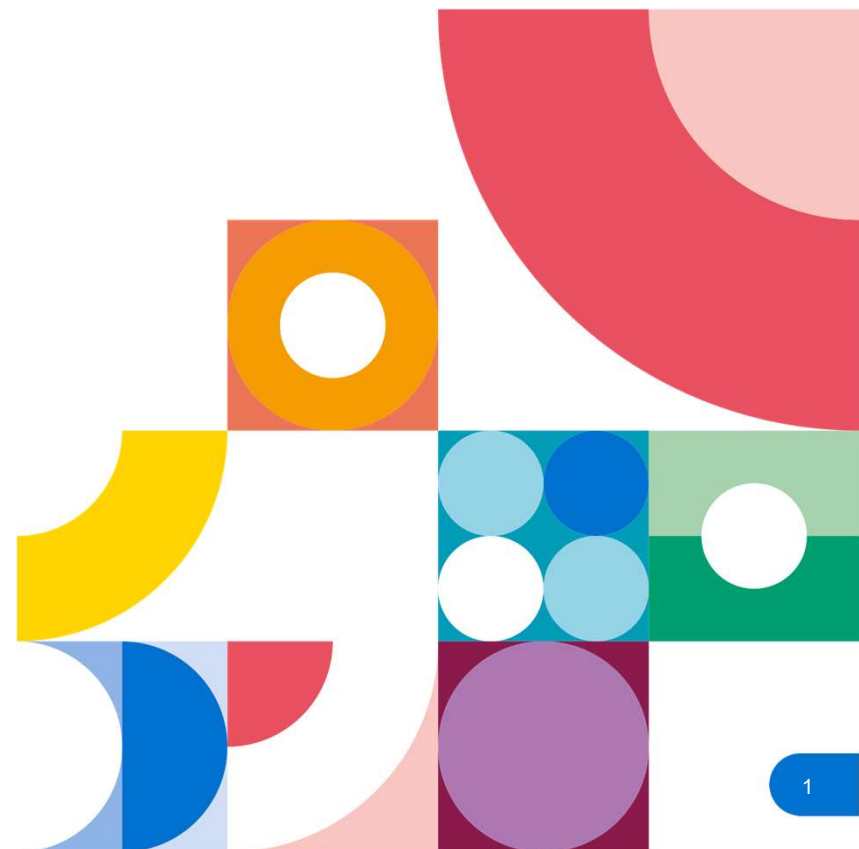
Sensibilisation aux tests

Pourquoi faire des tests ?

PNGT

Processus National de Gestion des Tests

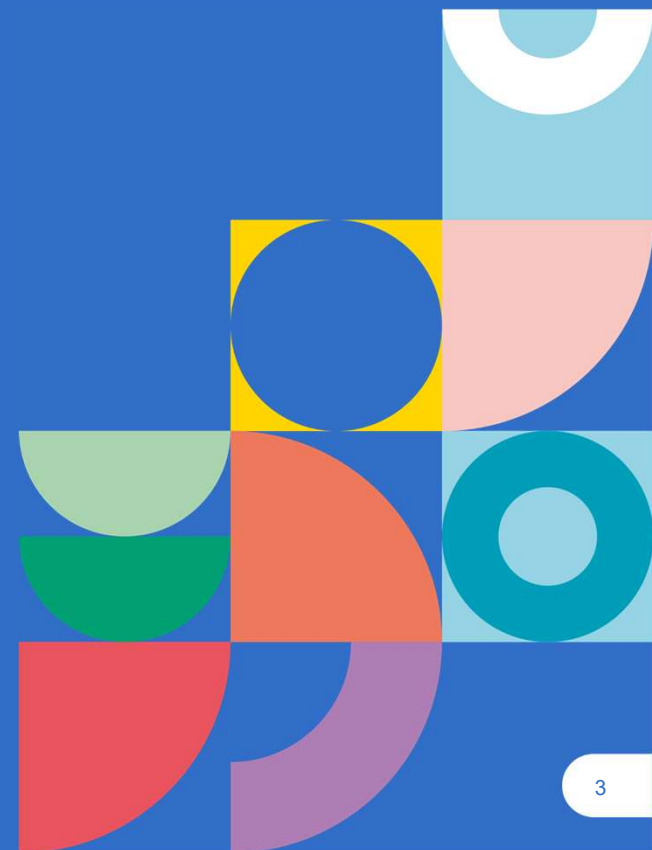
22/08/2025



1. Constat
2. Psychologie des tests
3. Les 7 principes du test
4. Notions de base
5. Les flots de contrôle
6. Scénarios, Cas de test, Etapes de test
7. Exigence et Couverture de test
8. Gestion des anomalies
9. Indicateurs de suivi
10. Livrables et traçabilité

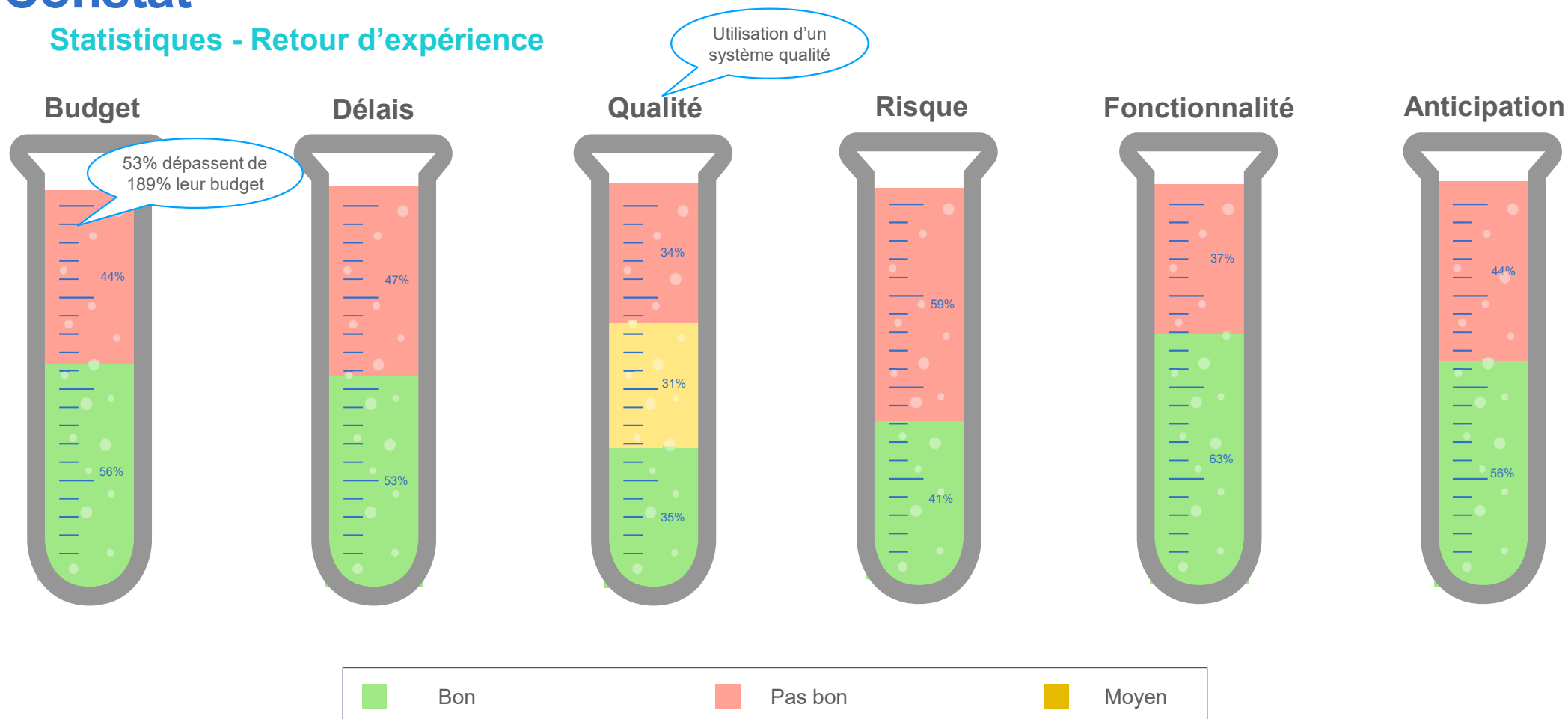


Constat



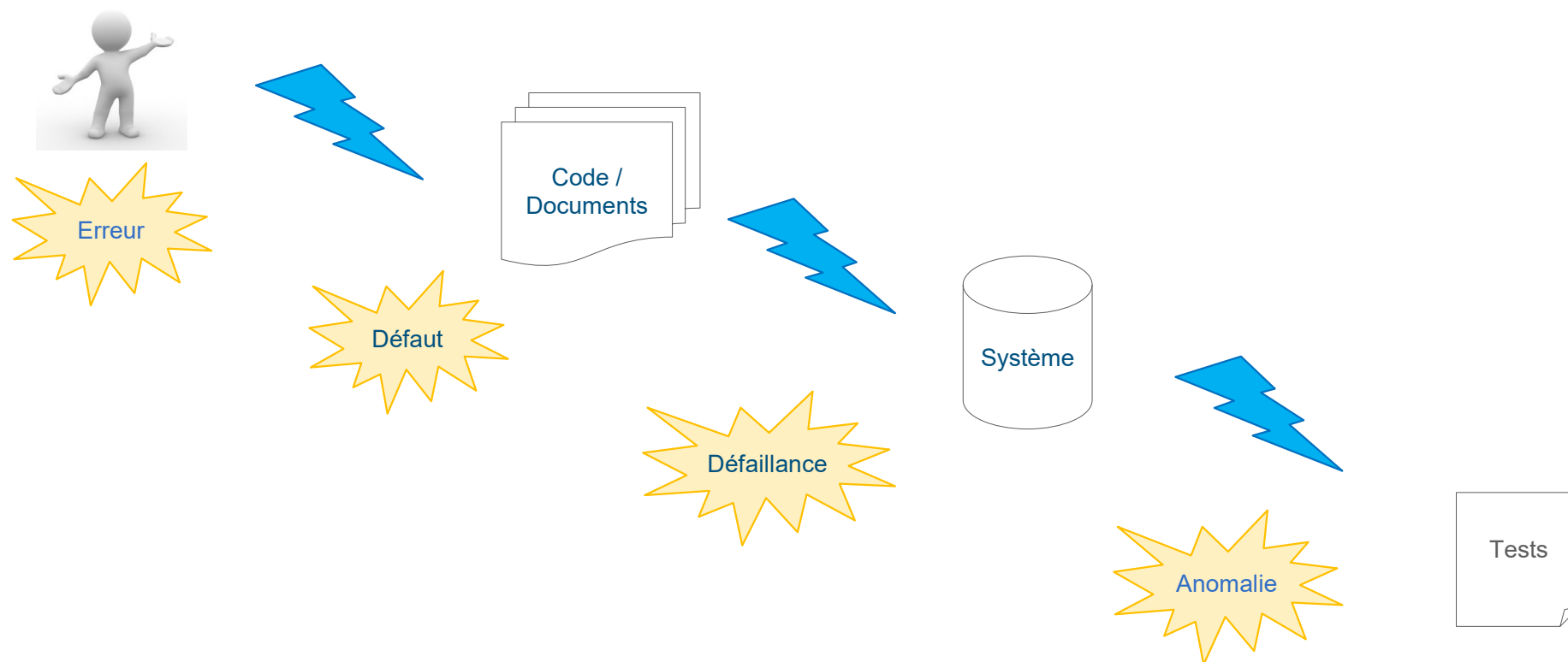
Constat

Statistiques - Retour d'expérience



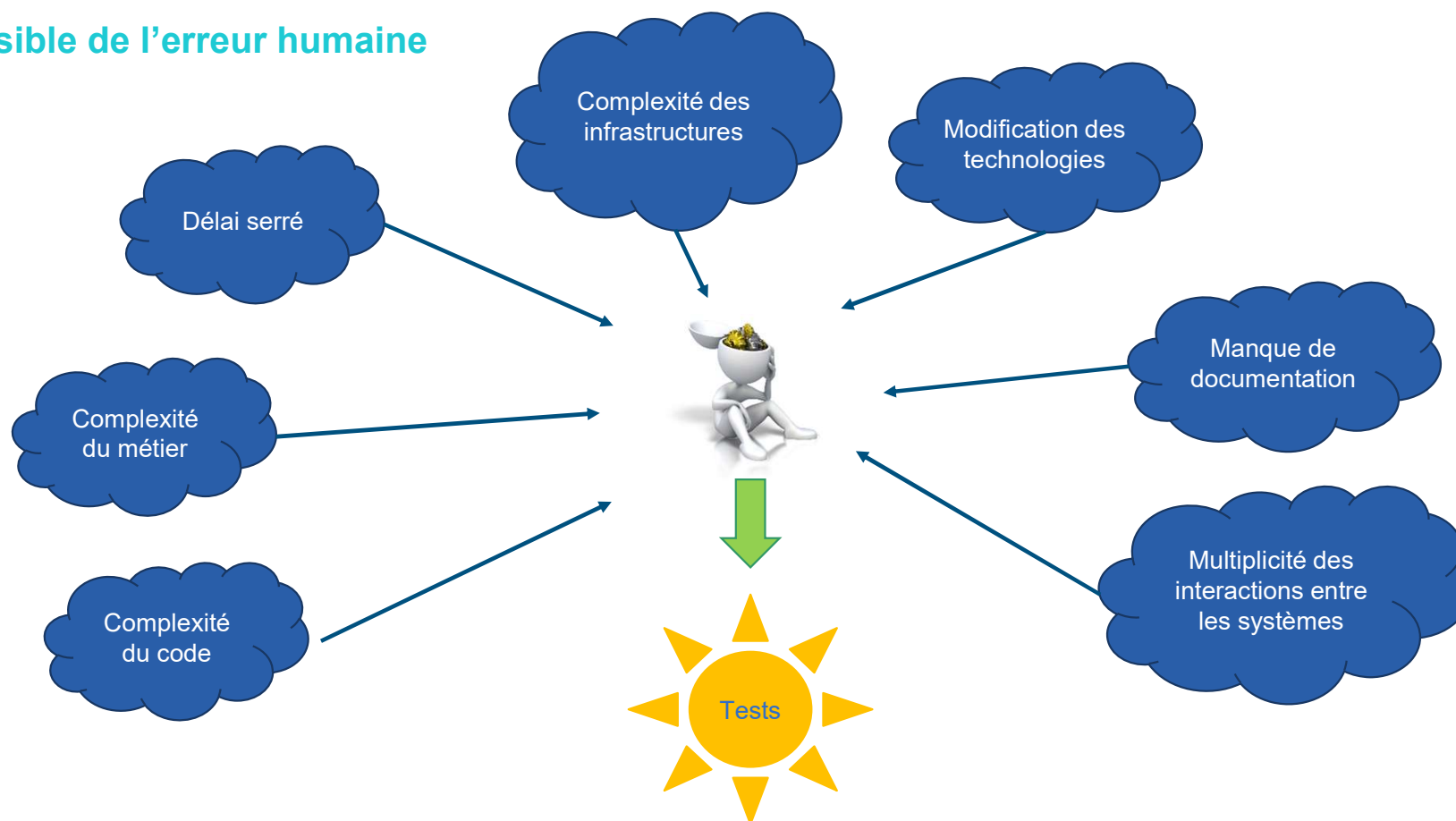
Constat

L'origine des anomalies



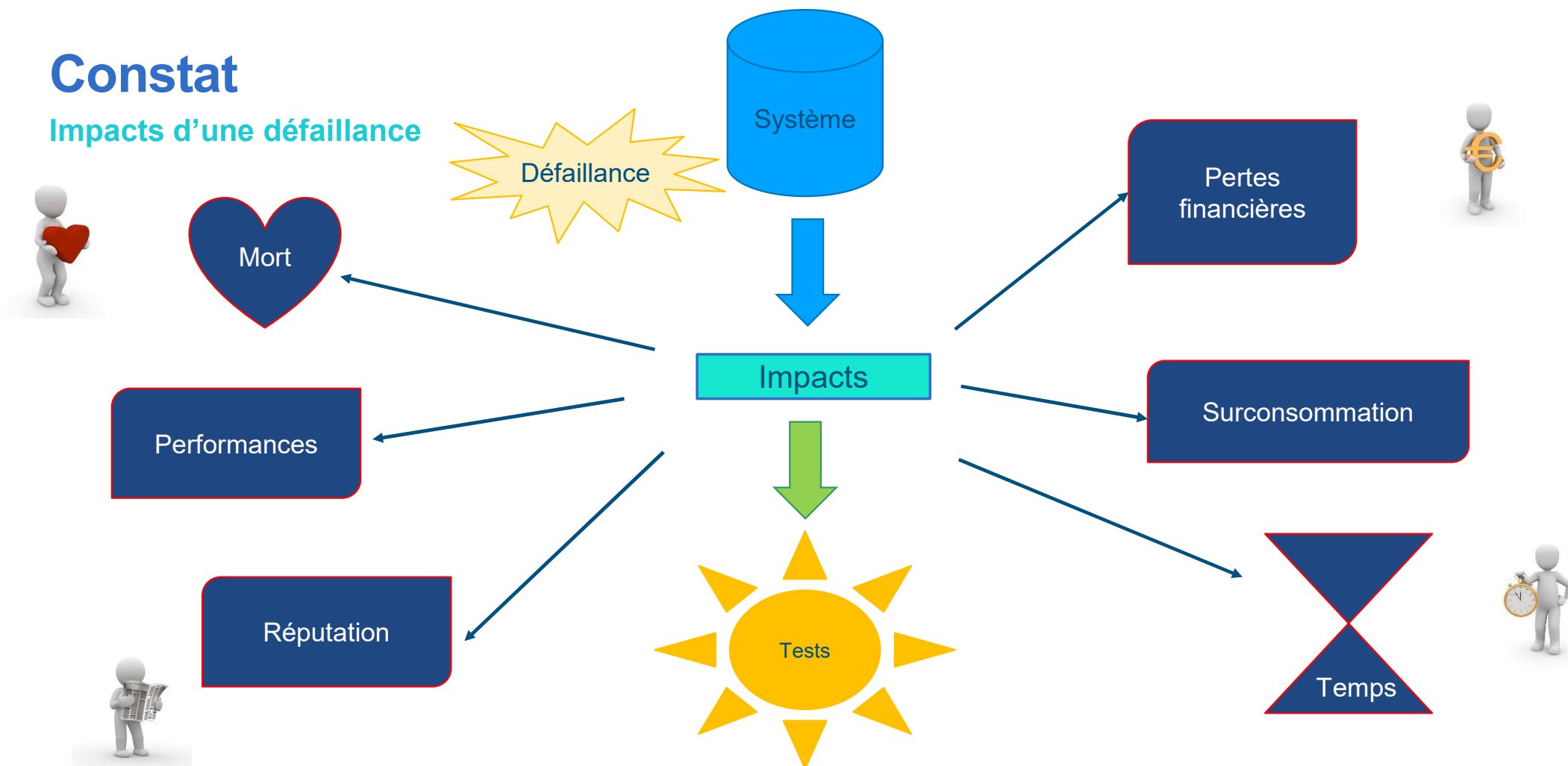
Constat

Cause possible de l'erreur humaine



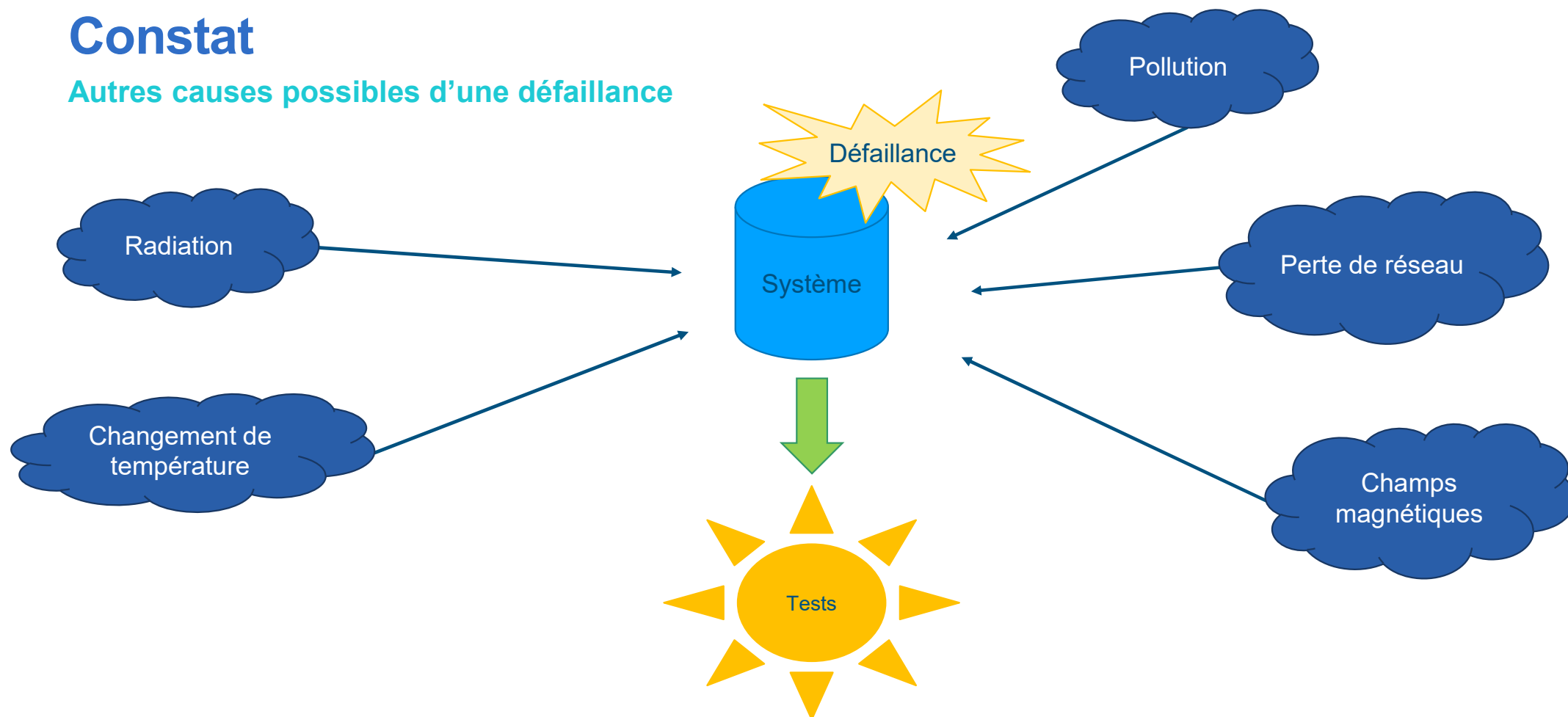
Constat

Impacts d'une défaillance



Constat

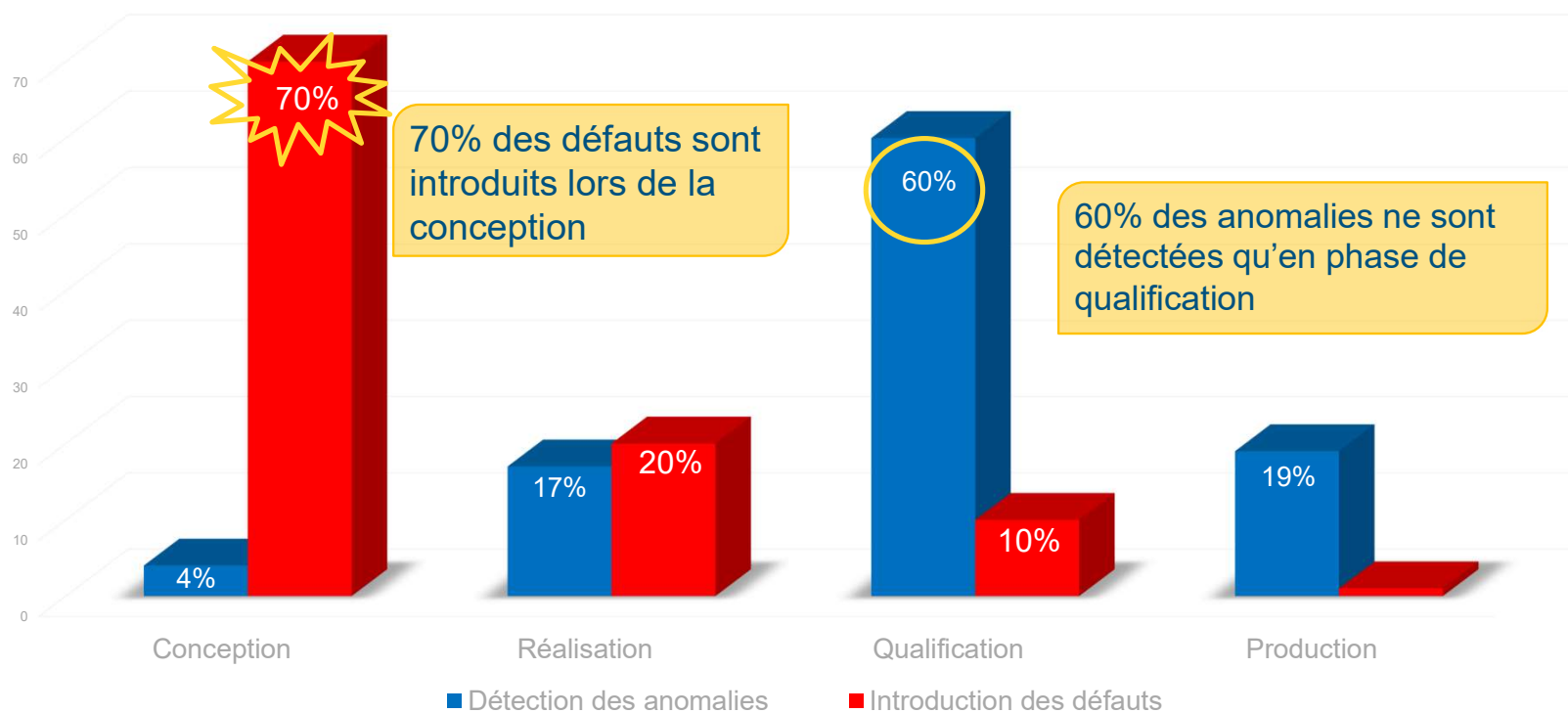
Autres causes possibles d'une défaillance



Constat

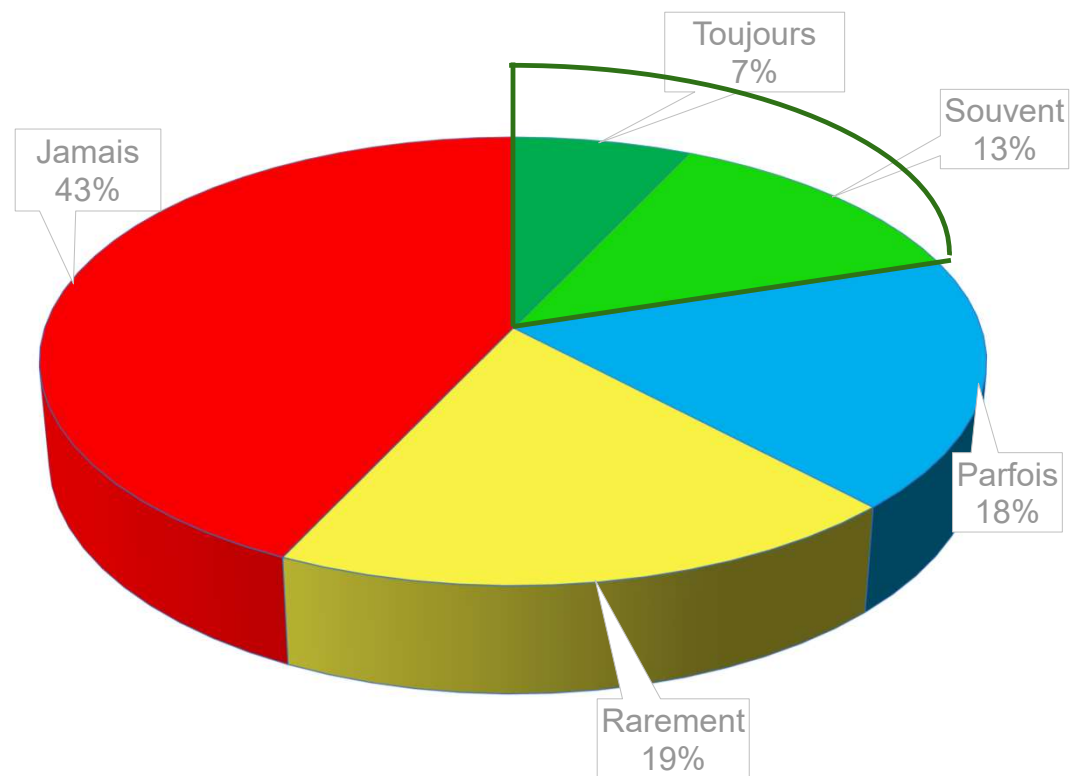
L'importance de tester tôt

Décalage entre détection des anomalies et introduction des défauts



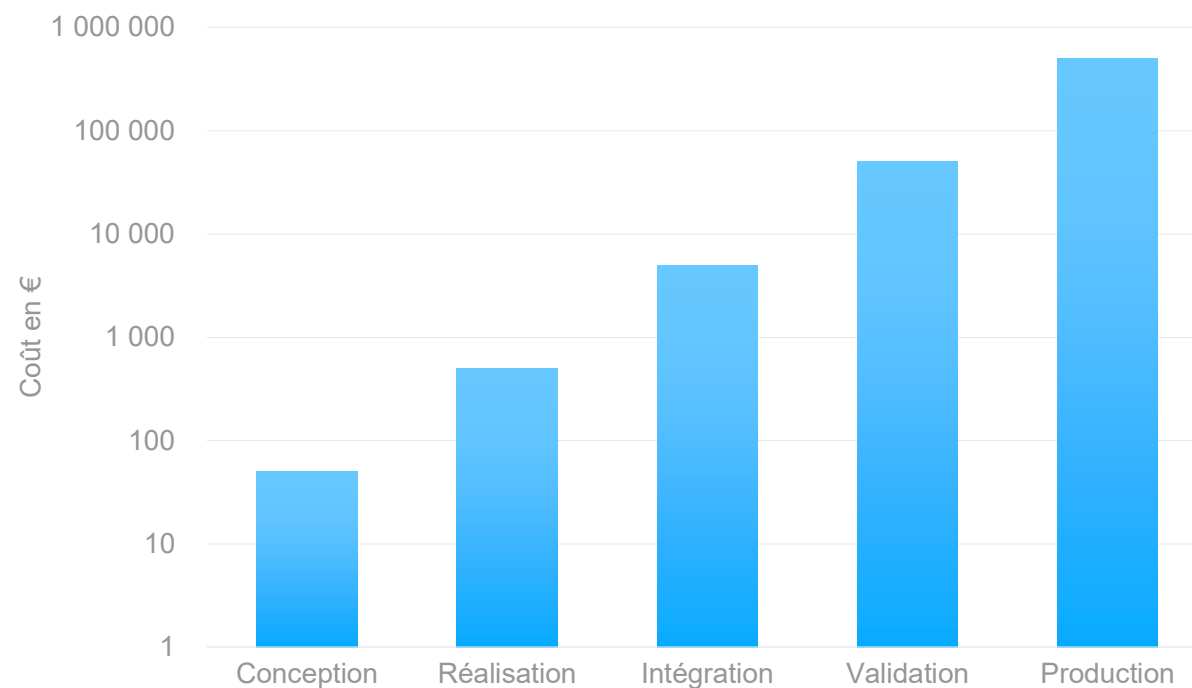
Constat

Identification des bonnes exigences



Constat

Tester tôt coûte moins cher



Plus tard = Plus cher

Constat

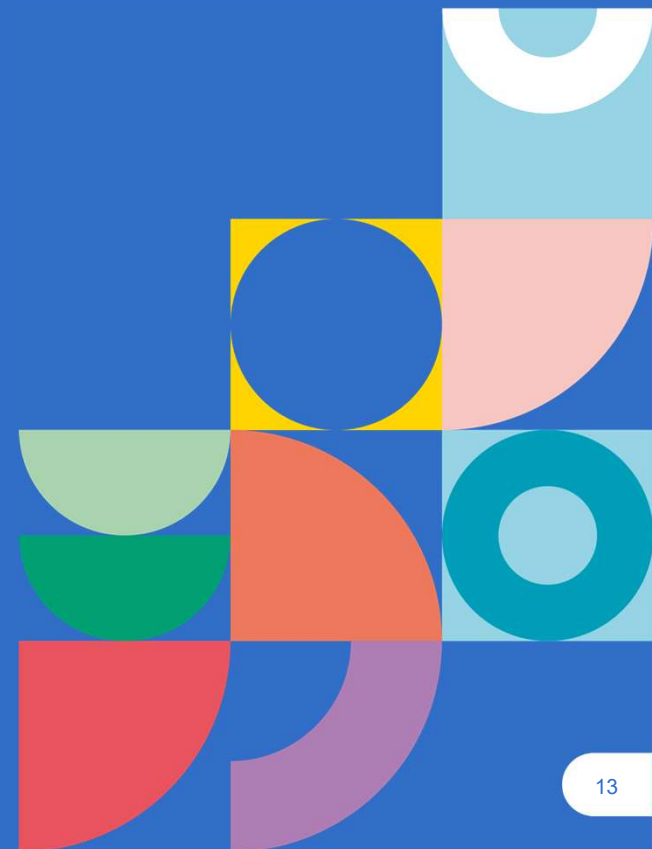
Evolution du coût de la correction d'une anomalie

« La qualité coûte cher,
mais il existe quelque chose de plus
coûteux que la qualité :
son absence »

P. Jocou (les enjeux économiques de la qualité)



Psychologie des tests



Idées reçues

Le test ça coute cher

Les testeurs sont des oiseaux de mauvais augure

Tu as été puni pour faire du test ?

Les testeurs n'y connaissent rien en technique et en développement

Tester c'est douter, compiler c'est gagné !

Pourquoi ce n'est pas automatisé ?

Le test c'est bien pour débiter

Les testeurs cassent le produit

Plus il y a de tests, mieux c'est !

Tester c'est exécuter

Si le projet est en retard c'est à cause des tests

Le test c'est à la fin du projet

Le test c'est le travail des testeurs



Psychologie des tests

Réalité des tests

- Objectif commun entre développeur et testeur
 - Le produit sort en production dans les délais impartis avec la meilleure qualité possible à moindre coût
- La qualification est une activité constructive notamment dans la gestion des risques du produit
 - Le meilleur moyen d'établir la confiance dans la qualité d'une solution, c'est d'essayer de la détruire
- Commencer par une collaboration plutôt que par des conflits
 - Rappeler à chacun l'objectif commun qui est de fournir des solutions de meilleure qualité
- Communiquer les anomalies, les progrès, les risques de façon factuelle, constructive et neutre, sans jugement.
- S'assurer que l'autre personne a bien compris ce que l'on a dit et vice et versa
- Quand un problème survient, c'est le problème de l'équipe : L'important c'est le résultat!



Psychologie des tests

Les qualités d'un « bon » testeur

- Curiosité
- Œil critique
- Adaptabilité
- Rigueur
- Remise en question
- Persuasion
- Force de proposition
- Esprit d'équipe
- Bonne humeur



- Esprit de synthèse
- Acceptation de l'échec
- Factuel en toutes circonstances
- Diplomatie
- Empathie
- Sens du détail
- Créativité
- Expérience
- Communication

Psychologie des tests

Indépendances des tests

- Evite le parti pris de l'auteur
- + efficace si les tests sont effectués par une équipe indépendante de l'équipe de développement
 - Equipe de développement fusionnelle, difficulté de remise en question, confiance entre les membres
 - Equipe de test peut prendre plus de recul, moins impactée émotionnellement, remise en question constante, persévérance, professionnalisme au niveau des tests
- Niveau de dépendance de test du plus faible ou plus fort:
 - Tests conçus par l'auteur
 - Tests croisés dans la même équipe : Celui qui exécute n'est pas celui qui a rédigé les tests
 - Tests effectués par une équipe de test indépendante ou par des spécialistes du test
 - Tests effectués par une organisation ou société différente



3

Les 7 principes du test



Les 7 principes du test

Principe 1 : Les tests montrent la présence des défauts

Principe 1 : Les tests montrent la présence des défauts

Les tests peuvent prouver la présence de défauts, mais ne peuvent pas en prouver l'absence.
Les tests réduisent la probabilité que des défauts restent cachés dans le logiciel mais, même si aucun défaut n'est découvert, ce n'est pas une preuve d'exactitude.

**Il y aura toujours des anomalies
même après 20 ans de production et d'utilisation**



Les 7 principes du test

Principe 2 : Les tests exhaustifs sont impossibles

Principe 2 : Les tests exhaustifs sont impossibles

Tout tester (toutes les combinaisons d'entrées et de pré-conditions) n'est pas faisable sauf pour des cas triviaux. Plutôt que des tests exhaustifs, utilisons l'analyse des risques et des priorités pour focaliser les efforts de tests.

- Tester une application simple qui comprend :
 - 20 écrans
 - 4 menus avec 3 options chacun
 - 10 champs
 - 2 types de données
 - 100 valeurs possibles
- Valeurs courantes d'exécution :
 - Expert = 10s
 - Moyen = 1min
 - Débutant = 10min
- Temps pour effectuer des tests exhaustifs :
 - $20 \times 4 \times 3 \times 10 \times 2 \times 100 = 480\,000$ cas de tests
 - $480\,000 \times 10 = \underline{\underline{55 \text{ jours}}}$ pour un expert !

EXEMPLE



Les 7 principes du test

Principe 3 : Tester tôt

Principe 3 : Tester tôt

Pour trouver des défauts tôt, les activités de tests devraient commencer aussi tôt que possible dans le cycle de développement du logiciel ou du système, et devraient être focalisées vers des objectifs définis.

- Effectuer des revues de spécification ou si besoin co-construire les spécifications avec l'équipe de test
 - Revue Statique en collaboration avec les équipes
- Réduire le coût des corrections en détectant des anomalies au plus tôt
 - Eviter le plus tard plus cher
- Commencer par les fonctionnalités les plus critiques pour trouver les anomalies les plus importantes en premier
 - Plus de temps pour corriger et vérifier
- Tester tôt ne demande rien d'autre que de l'organisation

Plus tard = Plus cher



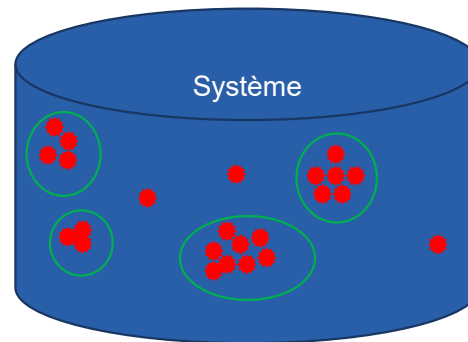
Les 7 principes du test

Principe 4 : Regroupement des défauts

Principe 4 : Regroupement des défauts

L'effort de test devrait être fixé proportionnellement à la densité des défauts prévus et constatés dans les différents modules.

Un petit nombre de modules contiennent généralement la majorité des défauts détectés lors des tests pré-livraison, ou affichent le plus de défaillances en opération



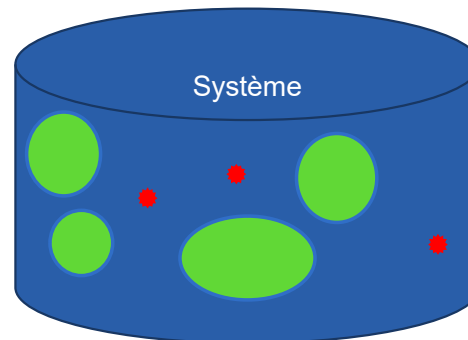
Les 7 principes du test

Principe 5 : Paradoxe du pesticide

Principe 5 : Paradoxe du pesticide

Si les mêmes tests sont répétés de nombreuses fois, il arrivera que le même ensemble de cas de tests ne trouvera plus de nouveaux défauts.

Pour prévenir ce “paradoxe du pesticide”, les cas de tests doivent être régulièrement revus et révisés, et de nouveaux tests, différents, doivent être écrits pour couvrir d’autres chemins dans le logiciel ou le système de façon à permettre la découverte de nouveaux défauts.



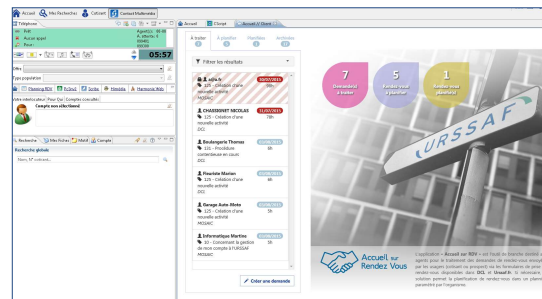
Les 7 principes du test

Principe 6 : Les tests dépendent du contexte

Principe 6 : Les tests dépendent du contexte

Les tests sont effectués différemment dans des contextes différents.

Par exemple, les logiciels de sécurité critique seront testés différemment d'un site de commerce électronique.

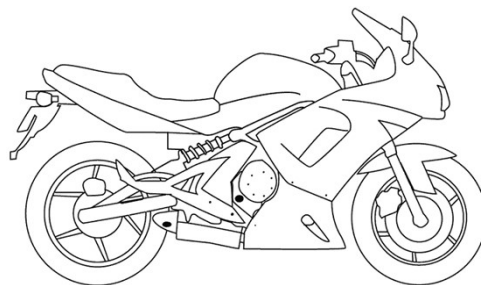


Les 7 principes du test

Principe 7 : L'illusion de l'absence d'erreur

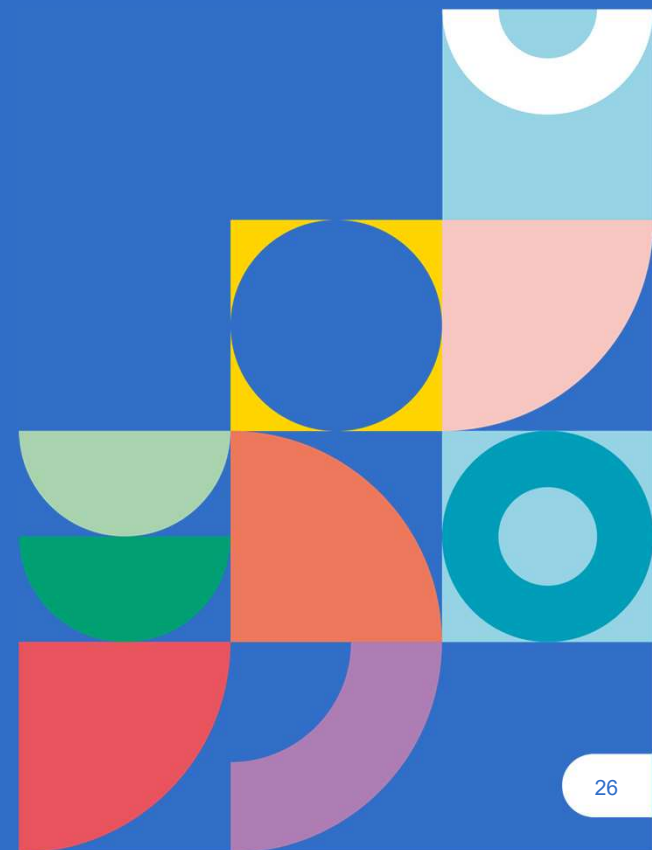
Principe 7 : L'illusion de l'absence d'erreur

Trouver et corriger des défauts n'aide pas si le système conçu est inutilisable et ne comble pas les besoins et les attentes des utilisateurs.





Notions de base



Notions de base

Les objectifs globaux du test

- Réduire le niveau de risque sur le système
- Trouver et anticiper des anomalies
- Produire des informations pertinentes à la prise de décision
- Augmenter la confiance sur la qualité produite
- Réduire les coûts post Mise En Production (MEP)

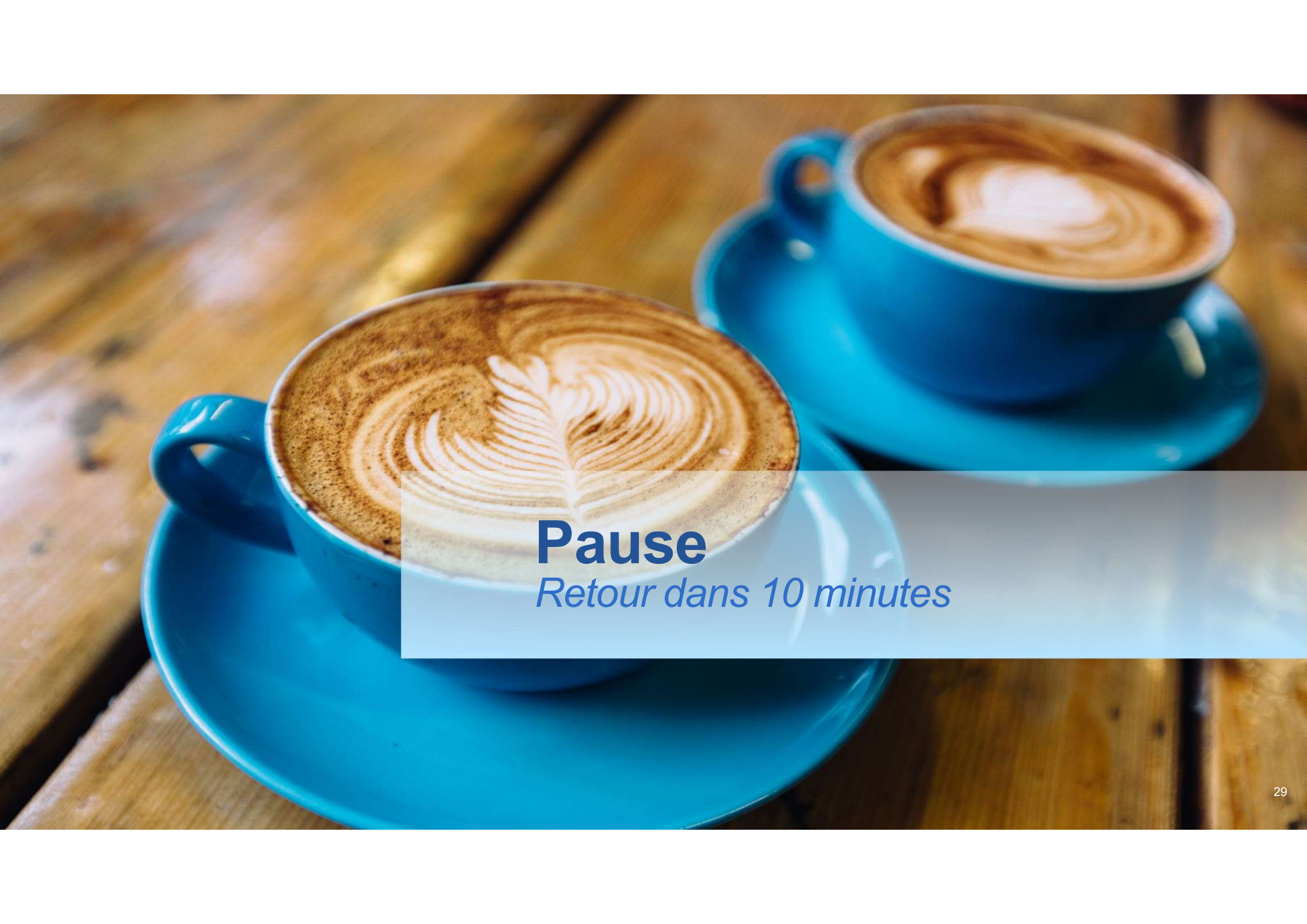


Notions de base

Les objectifs globaux du test

- Vérifier que le résultat correspond aux exigences utilisateurs (Vérification)
- Vérifier que le résultat correspond aux besoins utilisateurs (Validation)



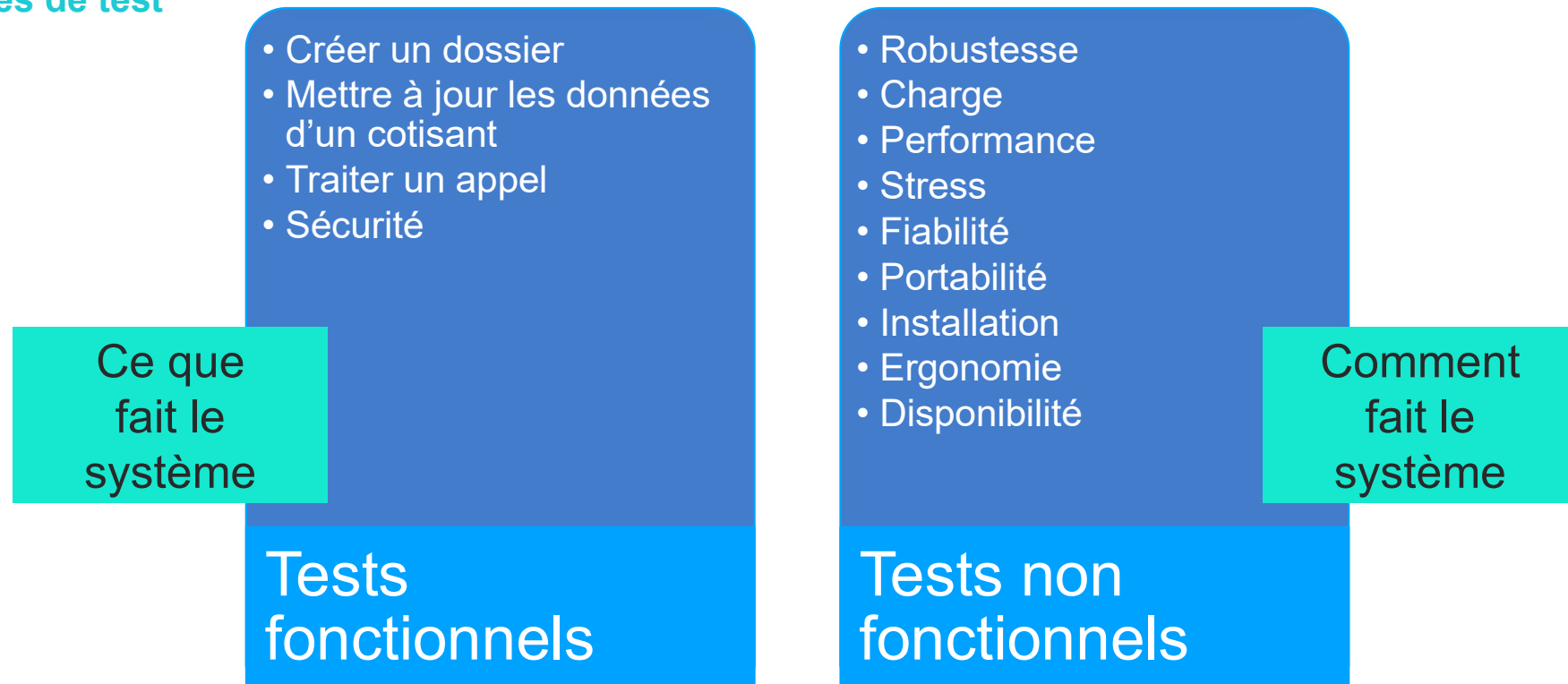


Pause

Retour dans 10 minutes

Notions de base

Les types de test



Notions de base

La pyramide des tests

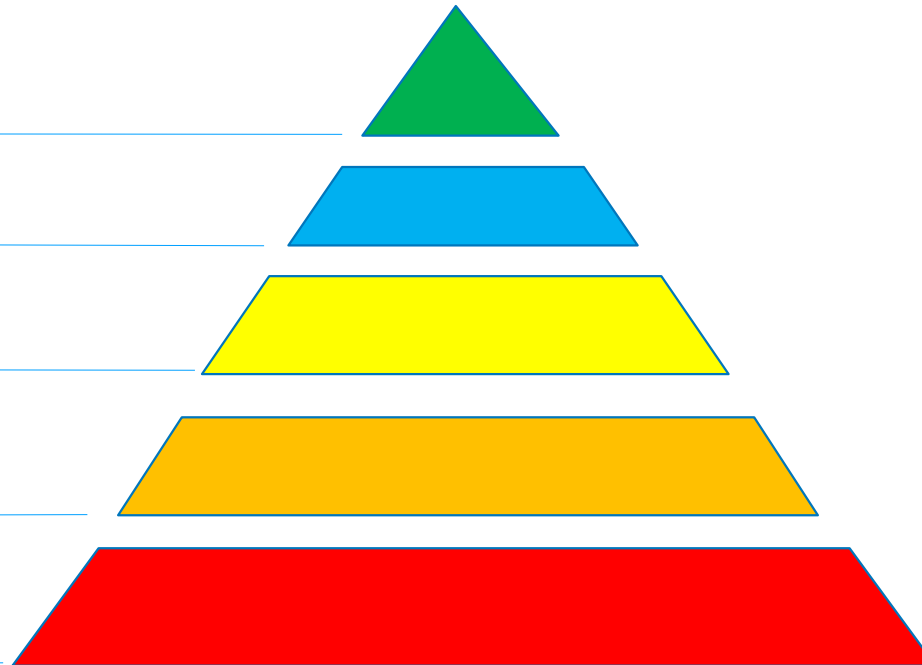
Tests d'Acceptation

Tests d'Intégration Système

Tests Système

Tests Intégration Composant

Tests Unitaire



Notions de base

Objectifs par niveaux de tests

- Tests unitaires
 - Chercher des défauts dans des fonctions du programme qui sont testables séparément
- Tests d'intégration composant
 - Tester les interactions et interfaces entre composants pour détecter des anomalies
- Tests système
 - Tester que le système respecte les exigences spécifiées
- Tests d'intégration système
 - Vérifier que les interactions entre systèmes fonctionnent correctement et répondent aux exigences spécifiées.
- Tests d'acceptation
 - Tester que le logiciel respecte les besoins utilisateurs exprimés
 - Tester que le logiciel est prêt à être déployé et utilisé
- Tests de confirmation
 - Tester que la correction d'une anomalie est effective
- Tests de régression
 - Tester que des anomalies n'ont pas été introduites durant le développement des évolutions ou corrections d'anomalies
- Tests de migration
 - Tester que les données et le fonctionnement sont identiques avant et après la migration
- Tests fumigatoires
 - S'assurer que les fonctionnalités vitales du système n'ont pas subi de régression

Notions de base

Tests unitaires

- Diverses appellations
 - TU
 - Tests Composant
 - Tests de module
 - Tests de programme
- Bases de tests
 - Conception détaillée
 - Conception du logiciel
 - Modèle de données
- Objets habituels de tests
 - Composants
 - Programmes
 - Conversion de données / utilitaires
 - Programme de migration



- But
 - Vérifier que les fonctions du programme qui sont testables séparément respectent les exigences
- Tests fonctionnels
 - Contrat de services
 - Opérations
- Tests non fonctionnels
 - Comportement des ressources (fuite mémoire,...)
 - Robustesse
 - Performance
 - Tests structurels (couverture des branches,...)
- Réalisateur
 - Equipe de développement (tests fonctionnels et non fonctionnels)
 - Et/ou CSPP (tests techniques)

Notions de base

Exemple d'un test unitaire

```
@Test
public void savePieceJointeAvecFichier() throws Exception {

    //on enregistre la pièce jointe
    savePieceJointe(savePieceJointe, null);

    GetPiecesJointesRequest requestPieceJointe = new GetPiecesJointesRequest();
    requestPieceJointe.setIdBibliotheque(bibliothequeNationale.getId());

    //on vérifie que l'enregistrement s'est bien passé
    ResponseMatcherToJAXB<GetPiecesJointesResponse> response = new ResponseMatcherToJAXB<GetPiecesJointesResponse>();
    executeRequest(requestPieceJointe).andExpect(response);

    List<InfosPieceJointe> infosPieceJointe = response.getJAXBObject().getInfosPieceJointe();
    Assert.assertTrue(infosPieceJointe.size() == 1);
    Assert.assertTrue(infosPieceJointe.get(0).getTitre().equals(savePieceJointe.getTitre()));
}
```

Notions de base

Tests d'intégration Composant



- Diverses appellations
 - TIC
 - Tests d'intégration Unitaire
- Bases de tests
 - Conception du logiciel et du Système
 - Architecture
 - Cas d'utilisation
- Objets habituels de tests
 - Interfaces
 - Infrastructure
- But
 - Vérifier que les interfaces entre les composants respectent les exigences
- Tests fonctionnels
 - Contrat de services
 - Opérations
 - Interfaces
- Tests non fonctionnels
 - Charge
 - Performance
 - Stress
- Réalisateur
 - Equipe de développement (Tests d'intégration des composants)
 - CSPP (tests non fonctionnels)

Notions de base

Exemple d'un test d'intégration composant



Notions de base

Tests système

- Diverses appellations
 - TS
 - Qualification interne
- Bases de tests
 - Spécifications fonctionnelles
 - Exigences
 - Cas d'utilisation
 - Rapport d'analyse des risques
- Objets habituels de tests
 - Manuels système, utilisateur et opérationnels
 - Configuration système et données de configuration
 - Le système



- But
 - Vérifier que le système respecte les exigences spécifiées
- Tests fonctionnels
 - Cas d'utilisation
 - Procédures métiers
- Tests non fonctionnels
 - Exigences non fonctionnelles
- Réalisateur
 - Equipe de test indépendante de l'équipe de développement (tests fonctionnels et techniques)
 - et/ou CSPP (tests techniques)

Notions de base

Exemple d'un test système avec un distributeur de billet de banque

- Actions : Introduire la carte bancaire CB n°23546789322 dans le distributeur Caisse d'Epargne
 - Résultat attendu : L'application affiche l'écran de saisie de code de carte bleue
- Actions : Saisir le code « 23456 » dans le champ « Code ». Cliquer sur le bouton « Valider »
 - Résultat attendu : Le système affiche le message « Patienter... » avec un sablier qui tourne puis affiche l'écran de saisie du montant désiré
- Actions : Saisir le montant « 100,00 » puis cliquer sur le bouton « Valider ».
 - Résultat attendu : Le système affiche le message « Patienter... » avec un sablier qui tourne puis affiche l'écran de souhait d'un ticket
- Actions : Cliquer sur le bouton « Non »
 - Résultat attendu : Le système affiche le message « Patienter... » avec un sablier qui tourne puis affiche le message « Reprenez votre carte » et éjecte la carte
- Actions : Prendre la carte bancaire
 - Résultat attendu : Le système affiche le message « Patienter... » avec un sablier qui tourne puis affiche le message « Prenez vos billets » et ouvre la trappe avec les billets.
- Actions : Prendre les billets
 - Résultat attendu : La somme des billets correspond à la somme demandée. Le système affiche le message « Patienter... » avec un sablier qui tourne puis affiche le menu d'accueil

Notions de base

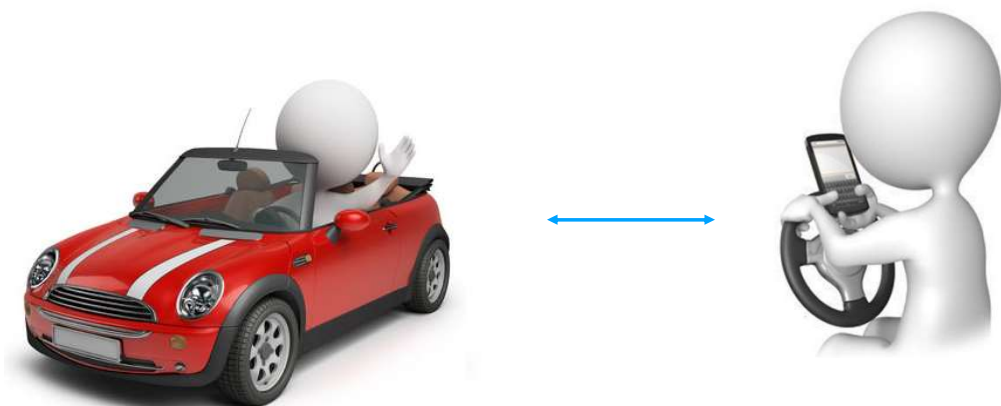
Tests d'intégration Système



- Diverses appellations
 - TIS
 - Tests d'intégration Système
- Bases de tests
 - Spécifications fonctionnelles
 - Exigences
 - Cas d'utilisation
 - Rapport d'analyse des risques
- Objets habituels de tests
 - Manuels système, utilisateur et opérationnels
 - Configuration système et données de configuration
 - Le système
- But
 - Vérifier que les interfaces entre les systèmes respectent les exigences
- Tests fonctionnels
 - Cas d'utilisation
 - Procédures métiers
- Tests non fonctionnels
 - Exigences non fonctionnelles
- Réalisateur
 - DAOS (tests d'intégration Système)
 - CSPP (tests non fonctionnels)

Notions de base

Exemple d'un test d'intégration système entre un téléphone et une voiture



Notions de base

Tests d'acceptation

- Diverses appellations
 - Validation Métier
 - Qualification externe
 - Recette utilisateur
- Bases de tests
 - Exigences utilisateur
 - Exigences du système
 - Cas d'utilisation
 - Processus métier
 - Réglementation
 - Législation
- Objets habituels de tests
 - Procédures utilisateur
 - Processus métier sur l'intégralité du système
 - Le système



- But
 - Valider que le système respecte les besoins utilisateurs exprimés
 - Valider que le système est prêt à être déployé et utilisé
- Tests fonctionnels
 - Tests d'acceptation réglementaire
 - Procédures métiers
 - Utilisabilité du système
- Tests non fonctionnels
 - Tests des backup et restaurations
 - Tâches de maintenance
 - Gestion des utilisateurs
- Réalisateur
 - CNV, URSSAF de proximité
 - MOA, Equipe de test utilisateur
 - et/ou CSPP (tests techniques)

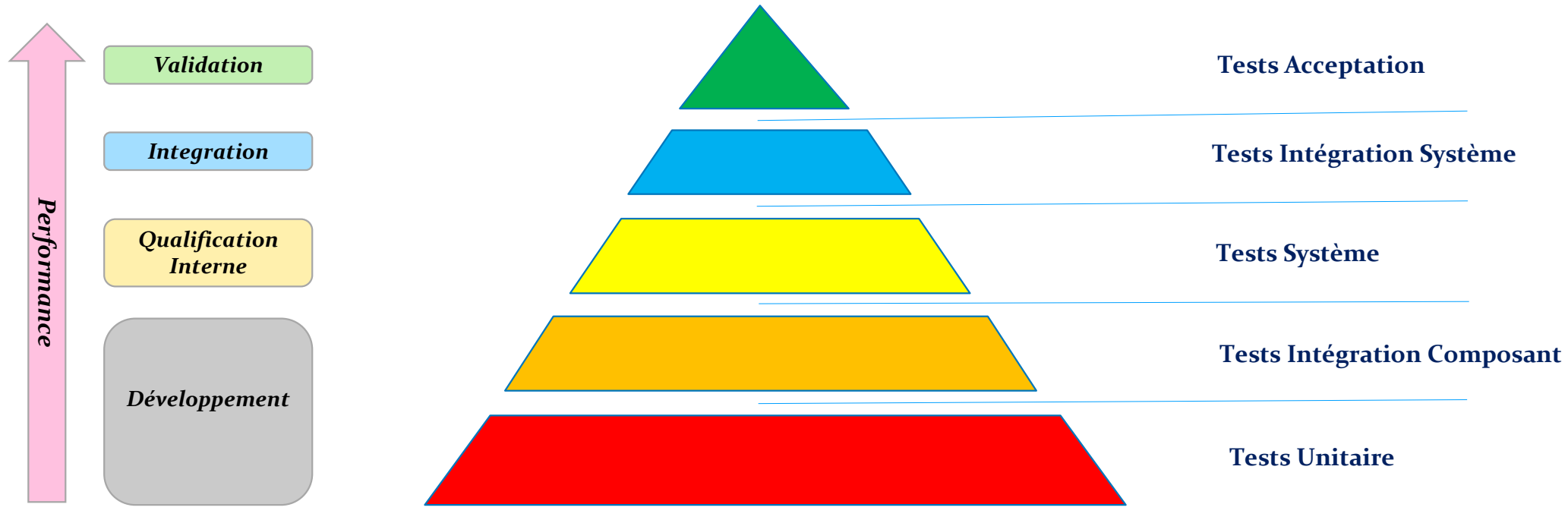
Notions de base

Exemple d'un test d'acceptation avec un distributeur de billet de banque

- Pré Requis : Avoir un client ayant un compte avec un solde > 5 000 000€
- Actions : Introduire la carte bancaire GOLD n°23501289322 dans le distributeur Caisse d'Epargne
Saisir le code « 21236 » dans le champs « Code ». Cliquer sur le bouton « Valider »
Saisir le montant « 3 000,00 » puis cliquer sur le bouton « Valider ».
 - Résultat attendu : Le système affiche le message « Patienter... » avec un sablier qui tourne puis affiche l'écran de souhait d'un ticket. Il n'y a pas de message d'erreur d'affiché car le client a le droit de demander une somme supérieure à 1 000€ au distributeur au vu du statut particulier.
- Actions : Cliquer sur le bouton « Non »
Prendre la carte bancaire
Prendre les billets
 - Résultat attendu : La somme des billets correspond à la somme demandée. Le système affiche le message « Patienter... » avec un sablier qui tourne puis affiche le menu d'accueil
- Actions : Observer les écritures comptables
 - Résultat attendu : Les écritures comptables ont débités le compte courant lié à la carte n°23501289322 avec le code « Débit »
- ...

Notions de base

En résumé



Optimisation d'effort de tests : Ne pas tester à nouveau ce qui a été testé dans les niveaux inférieurs =
Complémentarité, amélioration de la couverture

NB : Les tests fonctionnels et techniques, ainsi que les tests de régression, de performances et de sécurité peuvent être effectués si nécessaire à chaque niveau de test

Notions de base

En résumé

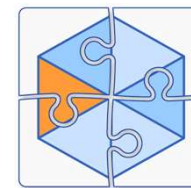
« Pour faire simple, une application est comme un puzzle. Ce n'est pas parce qu'une **pièce** est correcte (**tests unitaires**)



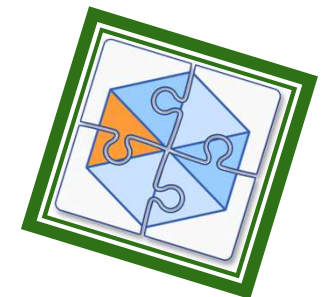
qu'elle s'emboîte bien avec ses **pièces voisines** (**tests d'intégration de composants**).



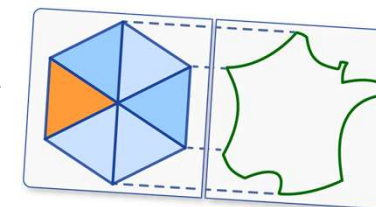
Ce n'est pas parce que le **puzzle complet** semble correct (**tests système**)



qu'il s'assemble bien **avec un cadre** (**tests d'intégration système**).



Et ce n'est pas parce que le puzzle s'intègre bien dans son environnement que c'est vraiment **celui que les utilisateurs voulaient** (**tests d'acceptation**). »



Notions de base

En résumé

- Exemple imagé l'analogie du puzzle avec une pyramide.
- Si le client est une entreprise d'Amérique du sud dont le business est le tourisme et si le puzzle livré représente la pyramide de Gizeh, alors ce puzzle ne sera pas vendable et ce même si les caractéristiques suivantes sont respectées :

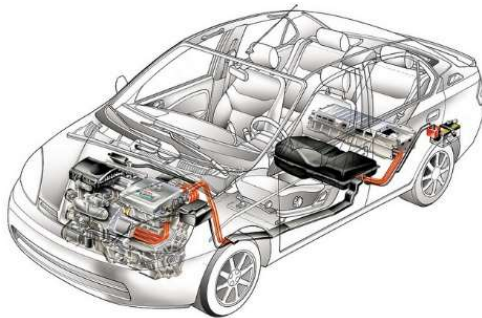
- ✓ Représentation d'une pyramide
- ✓ Nombre de pièce
- ✓ Coût de fabrication
- ✓ Qualité de l'image



Notions de base

Les techniques de Test

- « **Boite blanche** »
 - Tests basés sur une analyse de la structure interne du composant ou système
- « **Boite noire** »
 - Tests, fonctionnels ou non fonctionnels, sans référence aux structures internes du composant ou du système.



Notions de base

Les modes d'exécution

- Statiques

Tests effectués sans l'exécution de l'élément à tester

- Exemples :

- Revue des spécifications
- Revue de code (manuel ou automatisé)
- Revue de cahier de test
- Mesure de complexité
- Compilation
- Contrôle des normes de programmation



- Dynamiques

Tests qui nécessitent l'exécution de l'élément à tester.

- Exemples :

- Manuel
- Automatisés avec des outils (QTP,...)
- Tests de performances
- Tests de charge
- Facilité d'utilisation

Notions de base

Les modes d'exécution - Statistiques

**85 % des défauts peuvent être détectés
lors de la revue de spécification**



Notions de base

Automatisation

- Avantages
 - Réduit les coûts et les délais pour les tests
 - Accroît la couverture des tests
 - Gains de temps
 - Détection des régressions plus tôt
 - Tests de performance, charge, stress, volume,...
- Inconvénients
 - Maintenance
 - Réutilisabilité
 - Analyse des résultats
- Préconisation
 - Tests de régression de fonctionnalités stables et stratégiques
 - Tests avec des informations précises
 - Efficience : rentabilité investissement / coût
 - Tests systématiques, répétitifs et automatisables

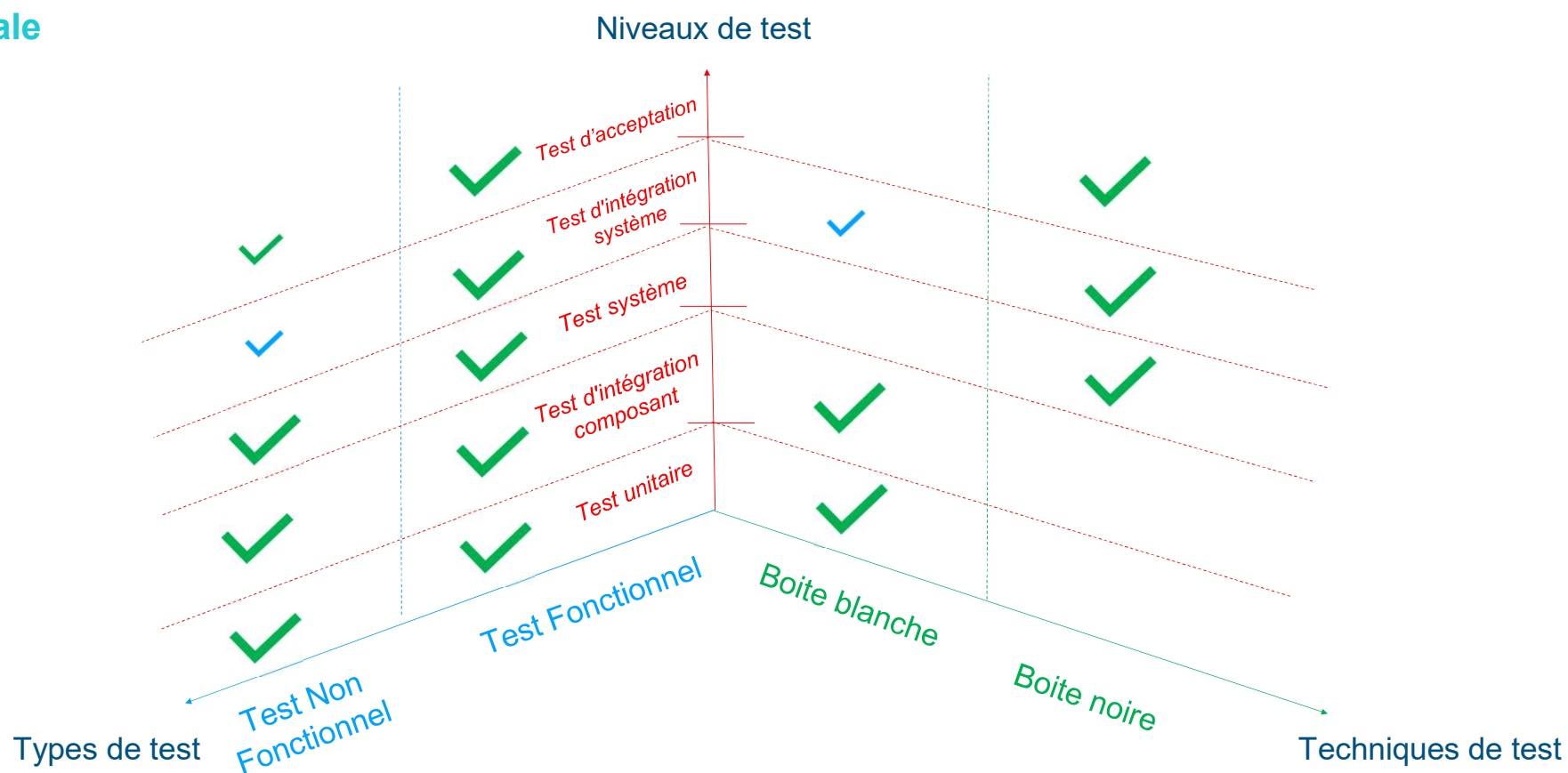
Non automatisables :



- Tests vérifiant la facilité d'utilisation
- Tests ponctuels
- Tests des impressions
- Tests sur des fonctionnalités non stables
- Tests complexes

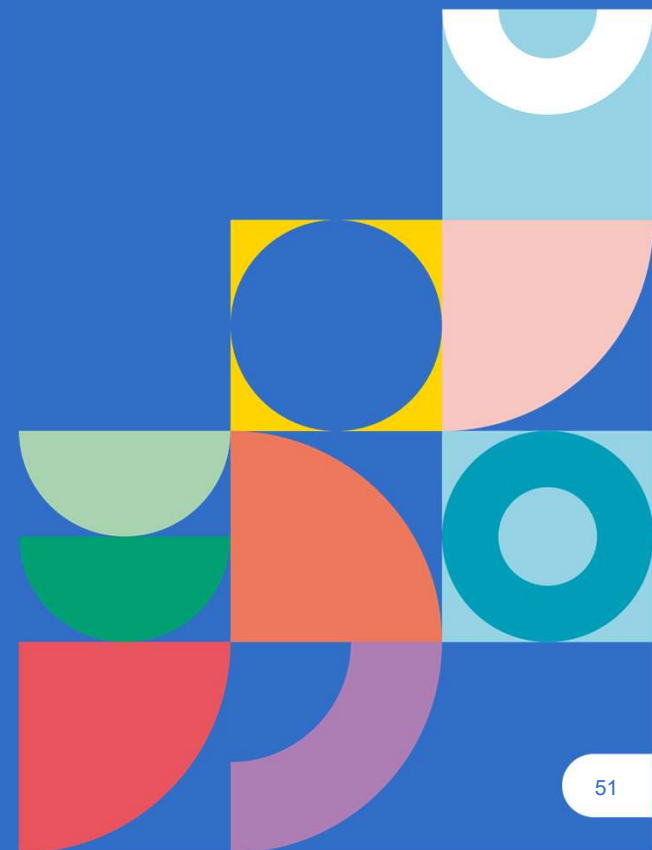
Notions de base

Vue globale



5

Les flots de contrôle



Les flots de contrôle

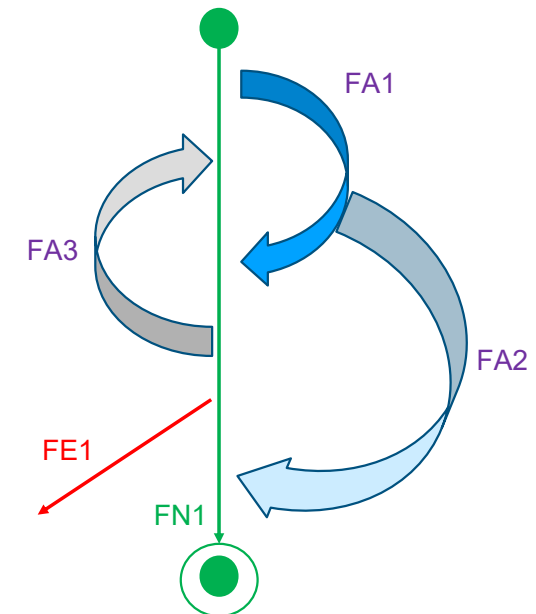
Définition

« Une représentation abstraite de toutes les séquences d'événements (chemins) dans l'exécution d'un composant ou système. »

« Représentation des interactions entre l'utilisateur et le système »

Il existe 3 types de Flots :

- **Flot Nominal**
 - L'objectif est atteint de façon normale et simple
- **Flot Alternatif**
 - L'objectif est atteint avec une variante du flot nominal
- **Flot Exceptionnel**
 - L'objectif n'est pas atteint à cause d'une situation anormale



Les flots de contrôle

Exemple d'un distributeur de billet de banque

- Objectif : Récupérer la carte bancaire et les billets de banque

Flot Nominal

- Introduction de la carte
- Saisie du code
- Saisie du montant
- Récupération de la carte
- Récupération des billets

Résultat :

- Carte bancaire récupérée
- Billets récupérés

Objectif Atteint : **OK**

Flot Alternatif

- Introduction de la carte
- Saisie du code
- Consultation du solde
- Saisie du montant
- Récupération de la carte
- Récupération des billets

Résultat :

- Carte bancaire récupérée
- Billets récupérés

Objectif Atteint : **OK**

Flot Exceptionnel

- Introduction de la carte
- Saisie du code
- Saisie du code
- Saisie du code

Résultat :

- Carte bancaire avalée
- Billets non récupérés

Objectif Atteint : **NOK**

6

Scénarios, Cas de test, Etapes de test



Scénarios, Cas de test, Etapes de test

Définition

- Scénario
 - Ensemble de transactions à effectuer permettant de valider un objectif spécifique
- Cas de test
 - Un ensemble de valeurs d'entrée, de préconditions d'exécution, de résultats attendus et de post-conditions d'exécution, développées pour un objectif ou une condition de tests particulier, tel qu'exécuter un chemin particulier d'un programme ou vérifier le respect d'une exigence spécifique
- Etape de test
 - Une étape de test comprend :
 - Une référence d'ordonnancement
 - Une description des actions à effectuer
 - Le résultat ou les résultats attendus suite aux actions effectuées

Scénarios, Cas de test, Etapes de test

Exemple

Scénario : Le cotisant émet une demande de délai de paiement

User Story ou
Cas d'usage

Liste des Cas de test associés

Prérequis pour démarrer le scénario : le cotisant doit être connecté à son espace DCL, dans la rubrique de contact

- Le cotisant sélectionne le motif « Demande de délai » puis il remplit et valide le formulaire (DCL)
- Le système envoie un accusé de réception de la demande (ODEA)
- L'agent URSSAF responsable du dossier reçoit la demande du cotisant (WATT)

Ensemble des cas de test couvrant le scénario

Détails des étapes d'un cas de test

Etapes de test du 1er cas de test

Nom de l'étape : Etape 1

Action : Dans la section « Echanger avec mon URSSAF », cliquer sur la rubrique « Un paiement »

Résultat attendu : La rubrique se déplie et affiche les 11 demandes relatives au paiement.

Nom de l'étape : Etape 2

Action : Cliquer sur le motif « Demander un délai de paiement »

Résultat attendu : DCL ferme la page et affiche le formulaire. Le formulaire doit contenir la progression du formulaire, un cadre sur les coordonnées et l'objet du message.

Nom de l'étape : Etape 3

Action : Remplir le formulaire avec le montant, le nombre d'échéances, le moyen de paiement, la 1ere date d'échéance et le motif de la demande. Cliquer sur « Valider »

Résultat attendu : La page s'actualise avec le fil d'ariane à « Envoi ». Les coordonnées du cotisant s'affichent ainsi que le message « Votre demande a été prise en compte. La réponse vous sera communiquée après examen. ».



Exigence et couverture de test



Exigence et couverture de test

Définition

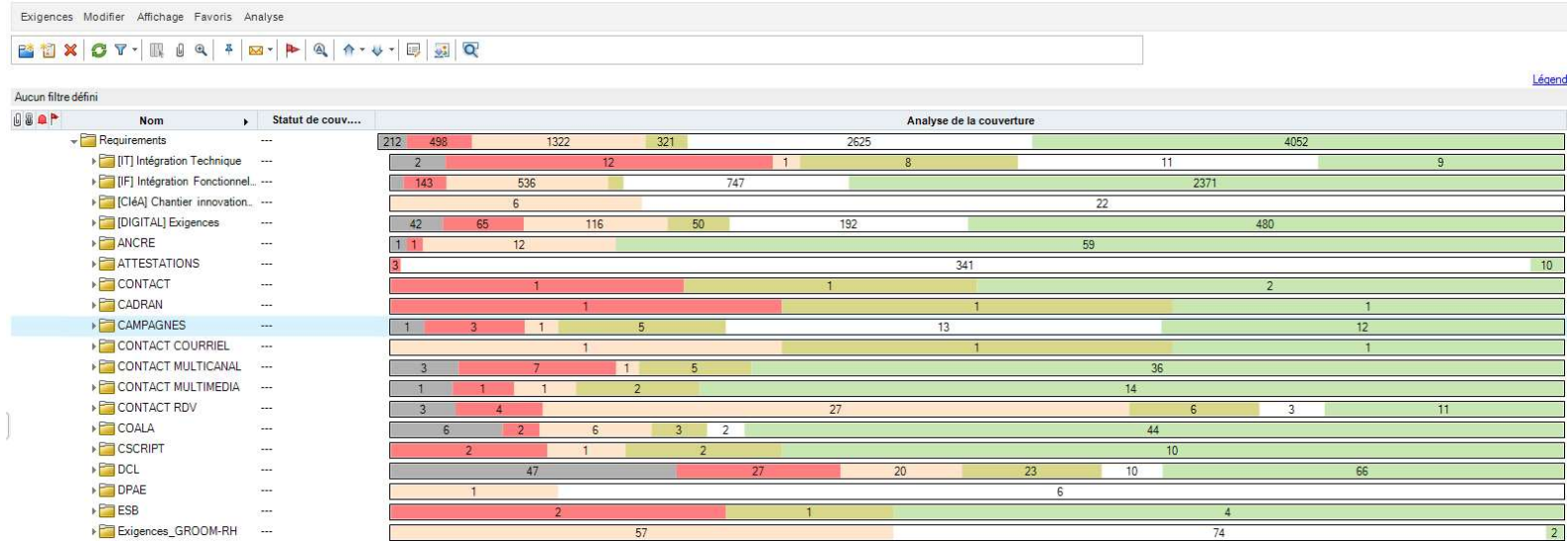
- Exigence
 - Une condition ou capacité requise par un utilisateur pour résoudre un problème ou atteindre un objectif qui doit être tenu ou possédé par un système ou composant pour satisfaire à un contrat, standard, spécification ou autre document imposé formellement.
 - Elle peut être fonctionnelle ou technique.
- Description d'une exigence
 - « Ajouter un document sur proposition d'un utilisateur »
- Description d'une règle de gestion
 - « Deux documents ne peuvent avoir le même titre »



Exigence et couverture de test

Définition

- Couverture de test
 - Le degré, exprimé en pourcentage, selon lequel un élément de couverture spécifié a été exécuté lors d'une suite de test



EXEMPLE

8

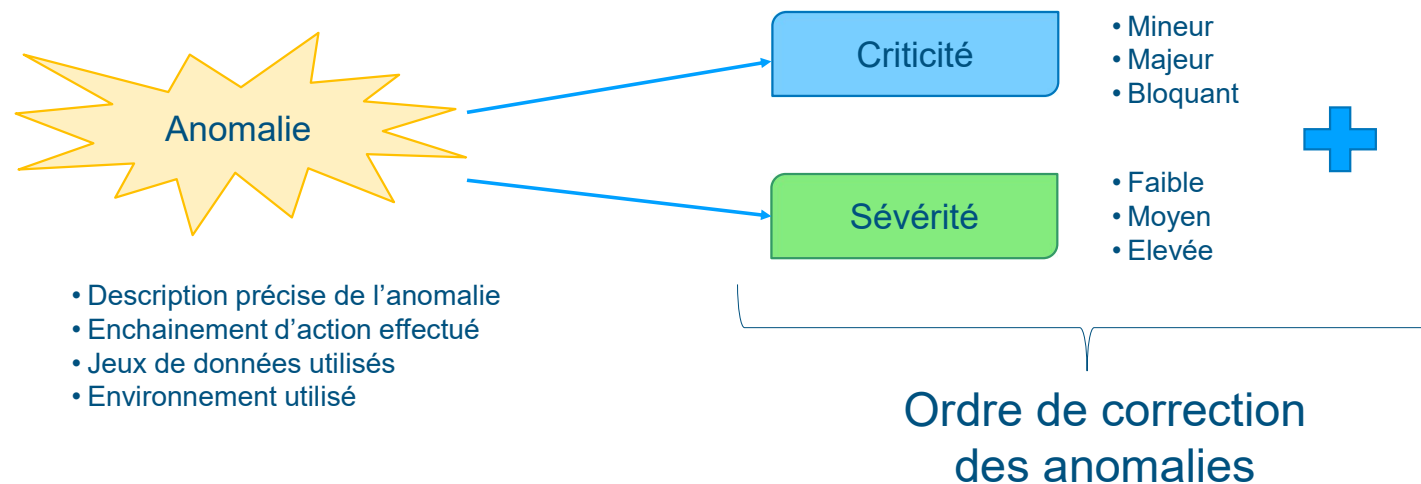
Gestion des anomalies



Gestion des anomalies

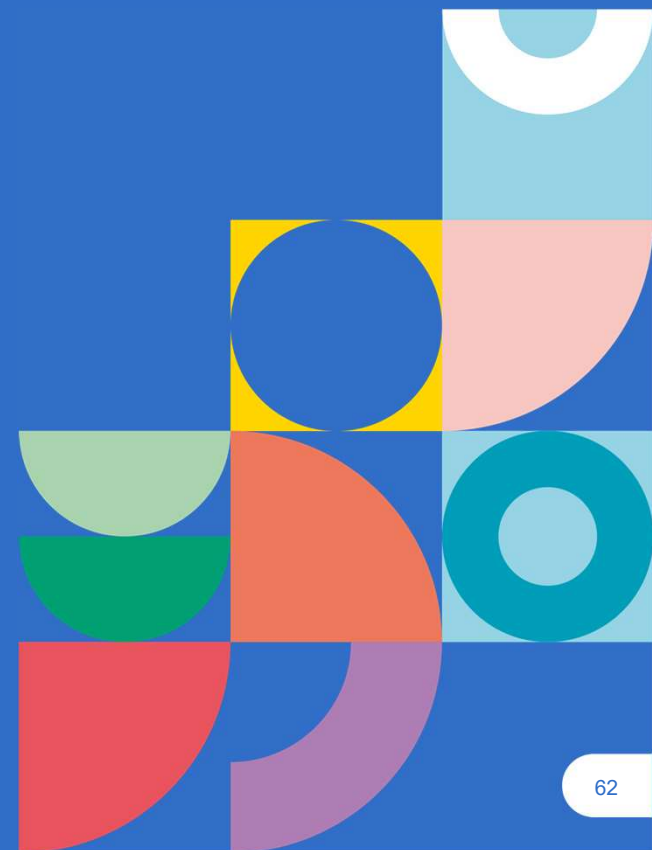
Définition

- Anomalie
 - Toute condition qui dévie des attentes basées sur les exigences de spécifications, documents de conception, documents utilisateurs, standards etc, ou des perceptions ou expériences de quelqu'un. Les anomalies peuvent être trouvées pendant, mais pas uniquement, les revues, tests, analyses, compilations ou utilisation des produits logiciels ou de la documentation applicable





Indicateurs de suivi

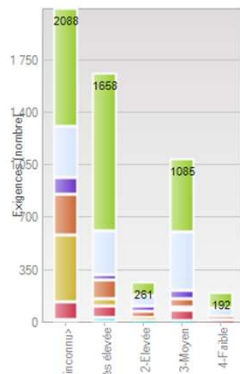


Indicateurs de suivi

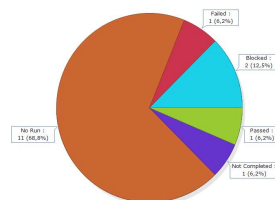
Disponibles

- Dépend
 - du contexte projet
 - des besoins de visibilité des CP
 - Des besoins de visibilité des autres équipes
- Au sein de l'URSSAF CN, les cinq indicateurs suivants sont disponibles

Etat des exigences par criticité

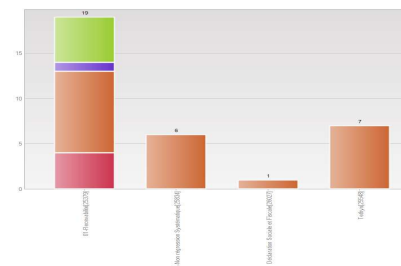


Suivi des anomalies par criticité



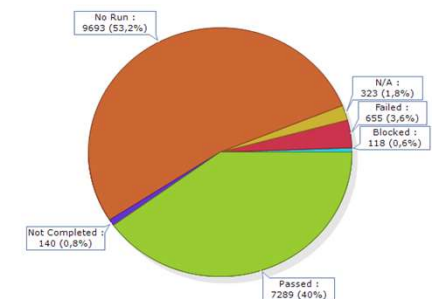
	<Inconnu>	Blockant	Blockant Tests	Critique	Majeur	Mineur	<Total>
A Qualifier Valid.					3	3	
A analyser BE					9	9	18
Demande de cré.	5	1	4	1	5	3	19
Fermée		1			1	7	9
Fermée Demand.			5	1	12	3	21
NO Qualification.				1		1	2
Nouveau		1			23	15	39
Prise en charge.					1		1
Qualifiée HDE					3	1	4
Réouverte					1	1	2
Réponse Apport.					2	1	3
<Total>	5	2	10	3	57	44	121

Couverture des exigences



Etat d'avancement des campagnes de tests

Etat des dernières exécutions du référentiel de test



10

Livrables et Traçabilité



Livrables et Traçabilité

Traçabilité

Capacité à identifier les éléments liés d'une documentation et d'un logiciel, tels que les exigences et les tests associés [Définition ISTQB].

Aptitude à retrouver l'historique, l'utilisation ou la localisation d'une entité au moyen d'identifications enregistrées [ISO 8402]



Livrables et Traçabilité

Traçabilité

Objectifs / Avantages :

- Tracer l'ensemble des actions effectuées lors des campagnes de test
 - Une anomalie peut être due à un enchainement d'action
- Construire un référentiel de test réutilisable dans le futur
 - Investissement sur l'avenir pour réduire les coûts d'écriture des campagnes de test
 - Industrialisation de l'écriture des cas de test
- Alimenter automatiquement les indicateurs de suivi
 - Permet de créer des données pertinentes à la prise de décision
- Vérifier que toutes les exigences soient couvertes par des tests
 - Taux de vérification des exigences par l'exécution des tests
- Prioriser les exigences et les cas de tests en fonction de la gestion des risques choisie
 - Permet de doser l'effort de test
- Historique des modifications avec versionning
 - Permet de garder une trace des décisions prises à un instant donné



Livrables et Traçabilité

Livrables

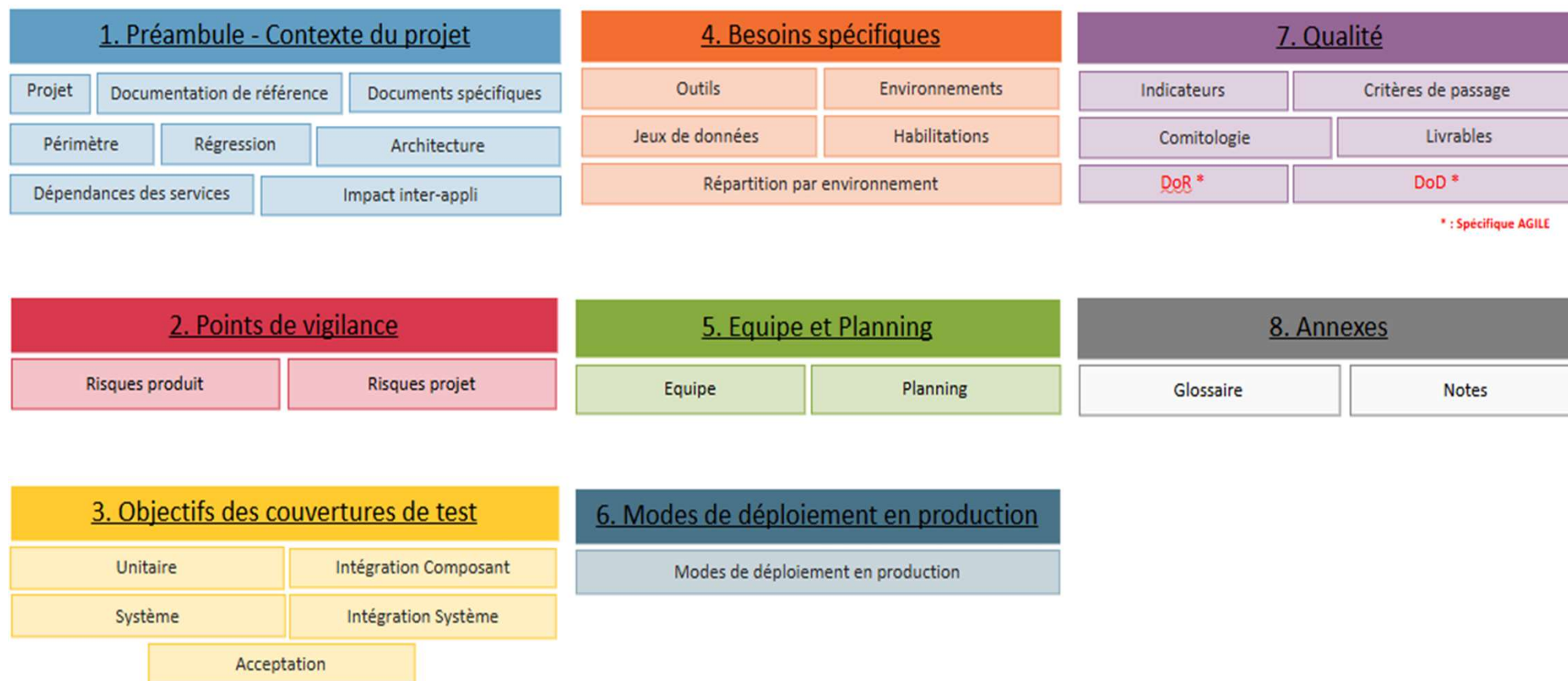
Les livrables en tests peuvent être :

- **Stratégie de test**
- Cahier de test généré à partir de ALM
 - Sans résultat de test
- Rapport de test généré à partir de ALM
 - Avec graphique
 - Sans graphique
- Documentation de lot
- Documentation fonctionnelle



Livrables et Traçabilité

Livrables : la stratégie de test



Des Questions ?





Évaluation de la formation

Enquête de satisfaction formation

Support de formation

Formation Sensibilisation aux tests.pdf