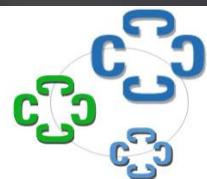




Cognizant  
Passion for building stronger businesses

Cognizant

# Présentation de la société

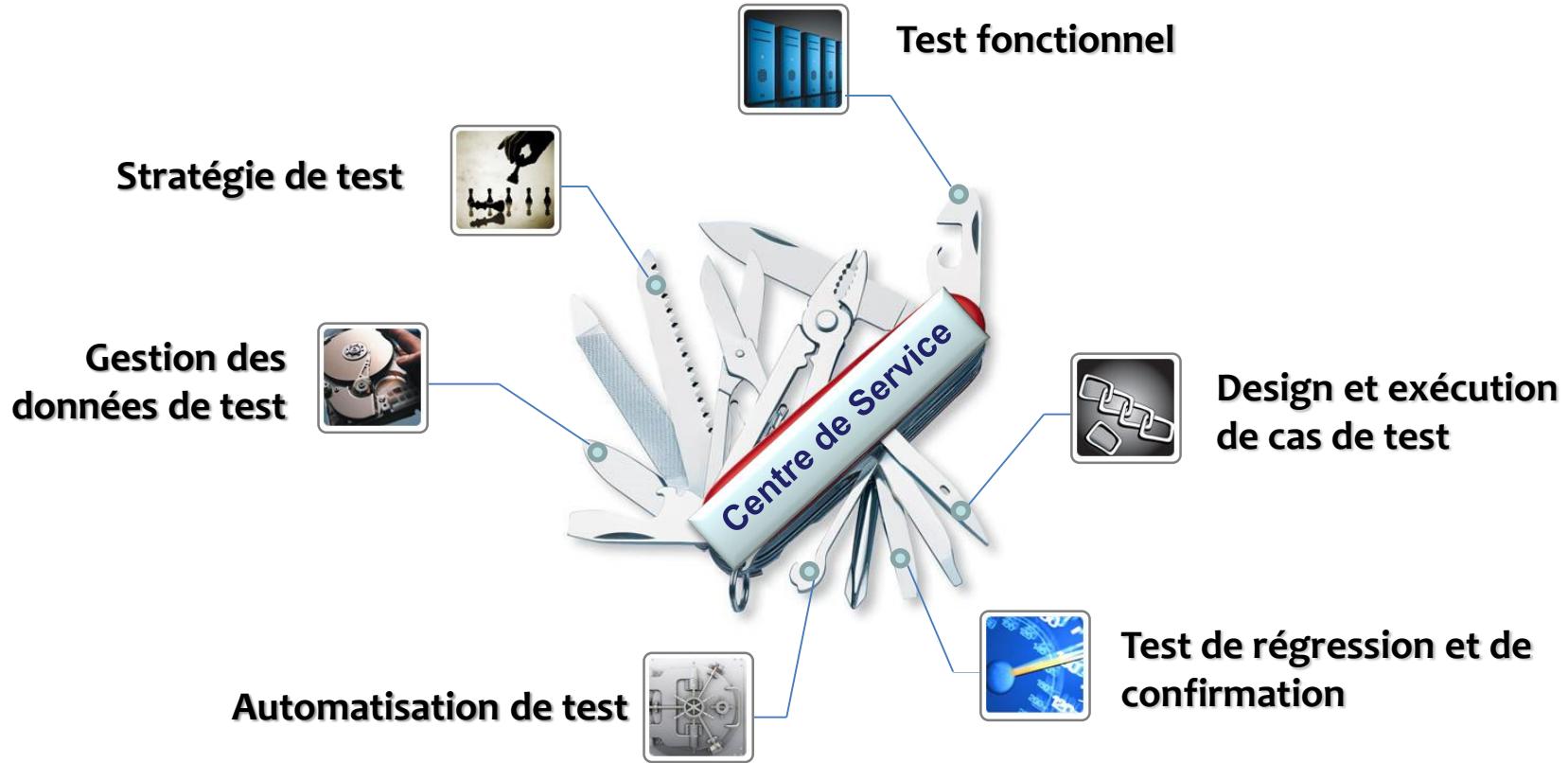


A  
**FORTUNE  
1000  
COMPANY**

# *Présentation générale de Cognizant*

- Cognizant a été fondée en 1994.
- Cognizant est une entreprise multinationale américaine fournissant des services informatiques et de conseil. Le siège social est à Teaneck dans le New Jersey aux Etats-Unis.
- Elle a des pratiques significatives dans les opérations bancaires et les services financiers, l'assurance, la santé, les sciences de la vie, vente au détail, fabrication, logistique, voyage ...
- Cognizant compte 220 000 employés dans le monde avec une forte présence en Inde et aux Etats Unis.
- Leader dans le domaine du Test logiciel
- <http://www.cognizant.com>

# *Le centre de services de Grenoble*

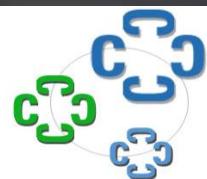




Cognizant  
Passion for building stronger businesses

# Cognizant

## Le Test Logiciel



A  
**FORTUNE**  
1000  
COMPANY

# Sommaire

- **Le métier du test**
  - ✓ Histoire du test
  - ✓ Les bugs célèbres
  - ✓ Le métier du test aujourd’hui
- Définition du test
- Test et risques
- Classification des tests
- La stratégie de test

# *Histoire du test*

- La notion de test logiciel prend ses racines dans les **années 60**. Les coûts de production d'un logiciel dépassaient de loin ceux du matériel du fait des anomalies.
- Pendant les **années 80** le test logiciel se développe en harmonie avec l'ingénierie logicielle. On différentie en particulier le test logiciel du debugging.
- Depuis le début des **années 2000** cette notion s'est développée autour de méthodes et d'outils pour devenir aujourd'hui une profession à part entière. Des « bug » célèbres ont permis de se rendre compte de l'importance de tester.

# Les bugs célèbres

- **Fusée Ariane 5** : explosion en vol après 36,7s suite à un dysfonctionnement informatique - 1996
- **le satellite Mars Climate Orbiter** : Objectif : modéliser le climat de Mars - perte de la sonde qui s'est crashée sur Mars au lieu de se mettre en orbite - 1999
- **Le bug de l'an 2000** : Aucun problème critique mais des centaines de milliards dépensés pour prévenir tout incident - 2000



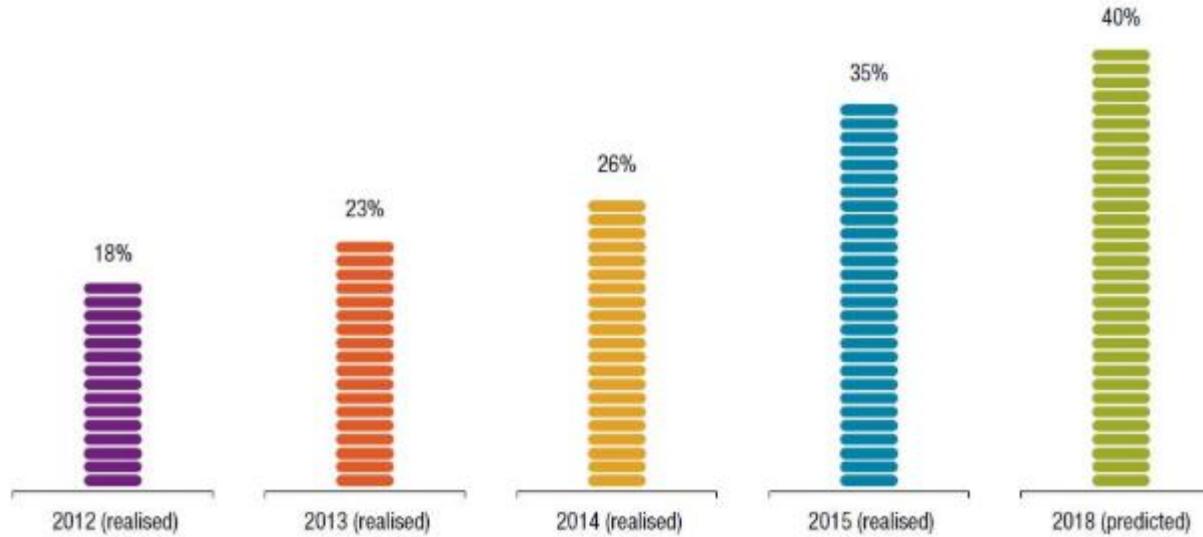
# *Les bugs célèbres*

- **Toyota Lexus ES350:** Un bug de l'ordinateur de bord entraînait l'accélération jusqu'à 150 km/h en désactivant la pédale de frein (200 à 400 morts) - 2009
- **Administration :** *Un certain nombre de projets nationaux ont été abandonnés entraînant des pertes de milliards d'euro*
  - » Agence Nationale de Santé UK (12 milliards – 2011)
  - » Service des impôts américains (8 milliards – 2002)
  - » Ministère de la défense français (470 millions – 2011)

<http://www.topito.com/top-10-des-erreurs-de-programmations-les-plus-couteuses-de-lhistoire>

# *Le métier du test aujourd'hui*

- Part du budget IT consacré aux tests et à la qualité logicielle



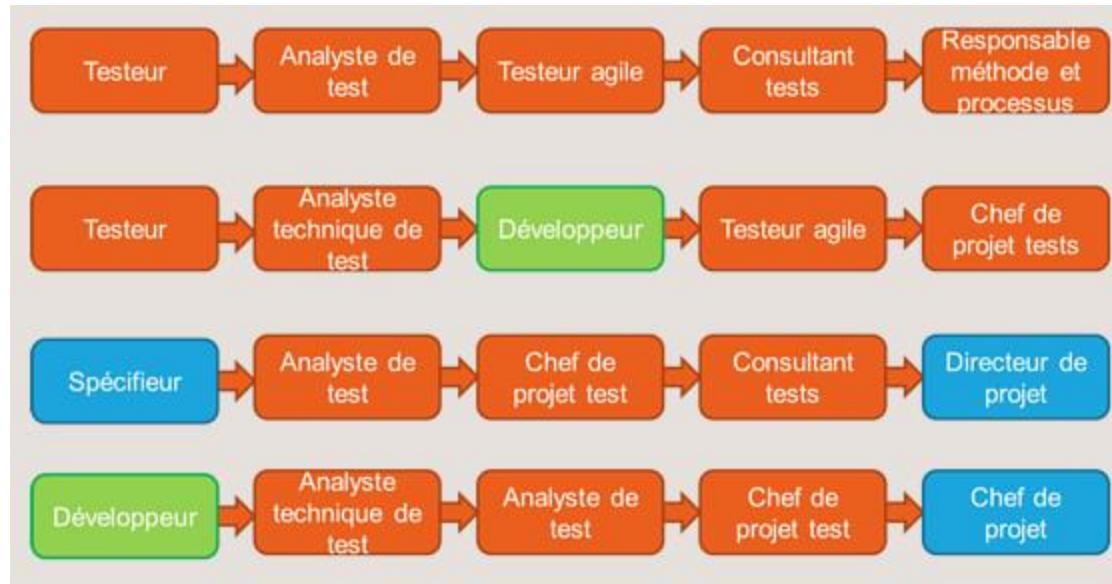
Budget divisé en parts égales entre maintenance d'applications existantes et nouveaux développements

# *Le métier du test aujourd'hui*

- Le métier du test est aujourd'hui une profession structurée
    - » *Normes : ISO, IEEE, CMMI*
    - » *La norme mondiale ISTQB*
    - » *Le comité CFTL en France*
  - *Les principaux postes*
    - » *Testeur*
    - » *Analyste de test*
    - » *Analyste technique de test*
    - » *Gestionnaire d'environnement de test*
    - » *Consultant test*
    - » *Chef de projet test*
    - » *Responsable méthode et processus de test*
- <http://www.cftl.fr/cf-tl/ressources/> Profils métiers

# *Le métier du test aujourd'hui*

- Exemple de parcours possibles



# Sommaire

- Le métier du test
- **Définition du test**
  - ✓ Définition
  - ✓ Origine des défauts logiciels
  - ✓ Pourquoi les tests sont-ils nécessaires ?
  - ✓ Objectifs des tests
  - ✓ Principes généraux des tests
  - ✓ La psychologie des test
- Test et risques
- Classification des test
- La stratégie de test

# Définition

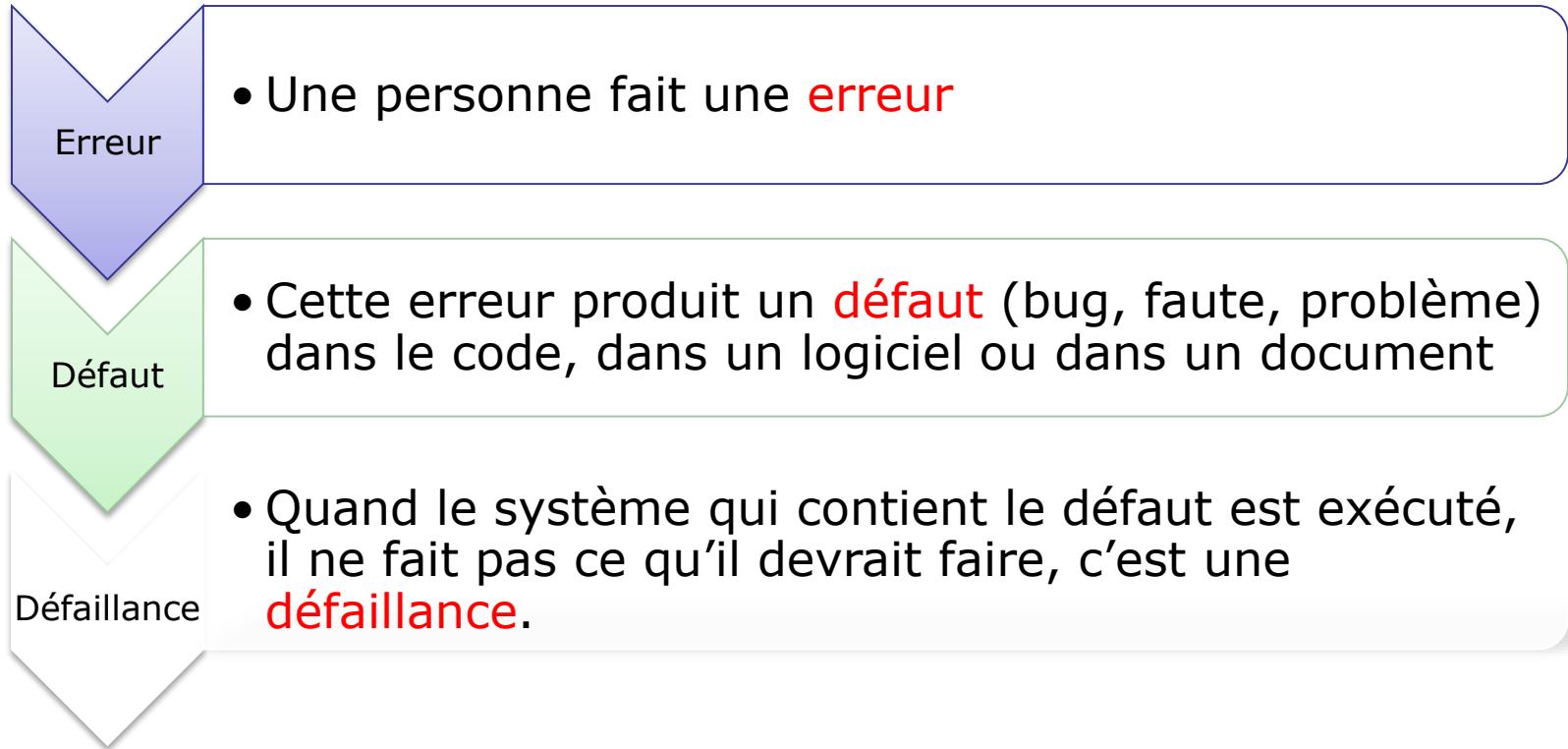
- « Le test est l'exécution ou l'évaluation d'un système ou d'un composant par des moyens automatiques ou manuels, pour vérifier qu'il répond à ses spécifications ou identifier les différences entre les résultats attendus et les résultats obtenus »

*IEEE (Standard Glossary of Software Engineering Terminology)*

- « Tester, c'est exécuter le programme dans l'intention d'y trouver des anomalies ou des défauts »

*G. Myers (The Art of Software Testing)*

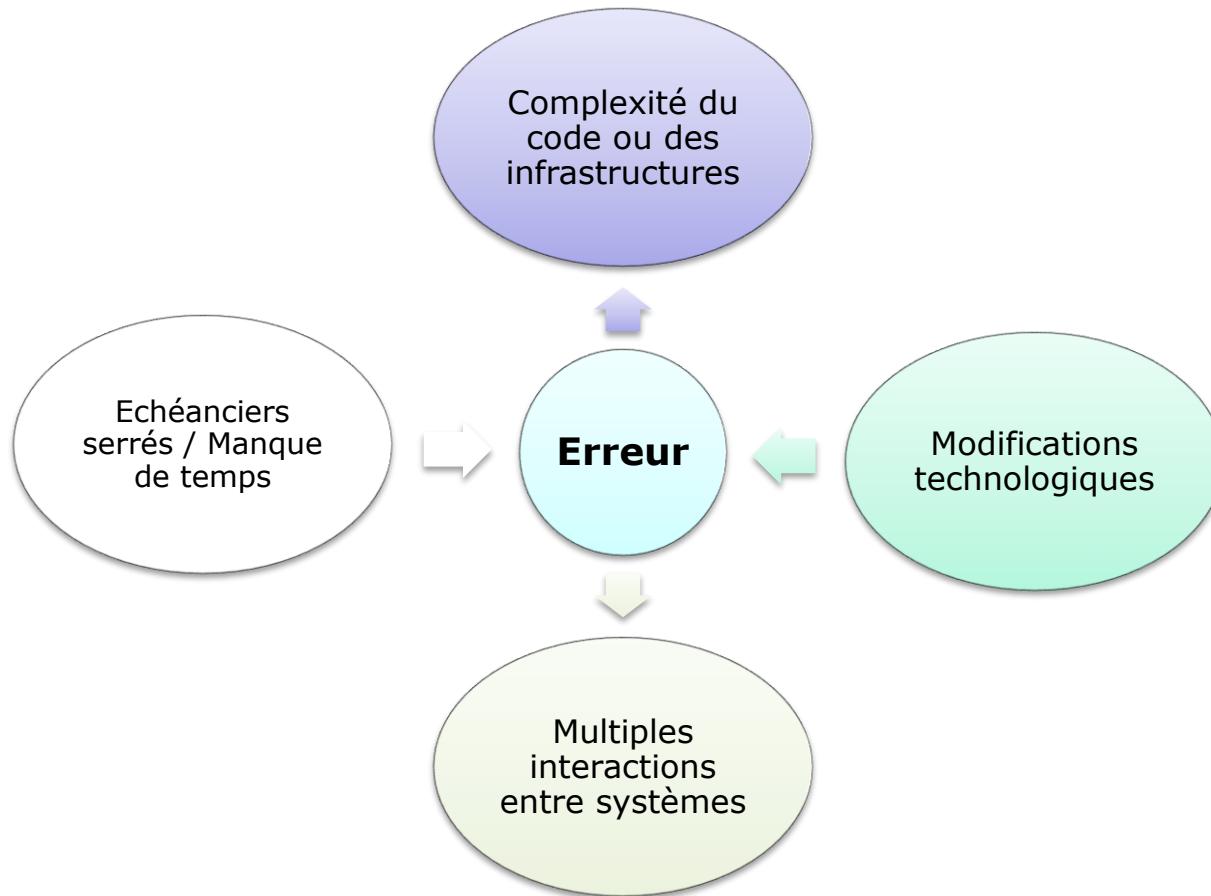
# Origine des défauts logiciels



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

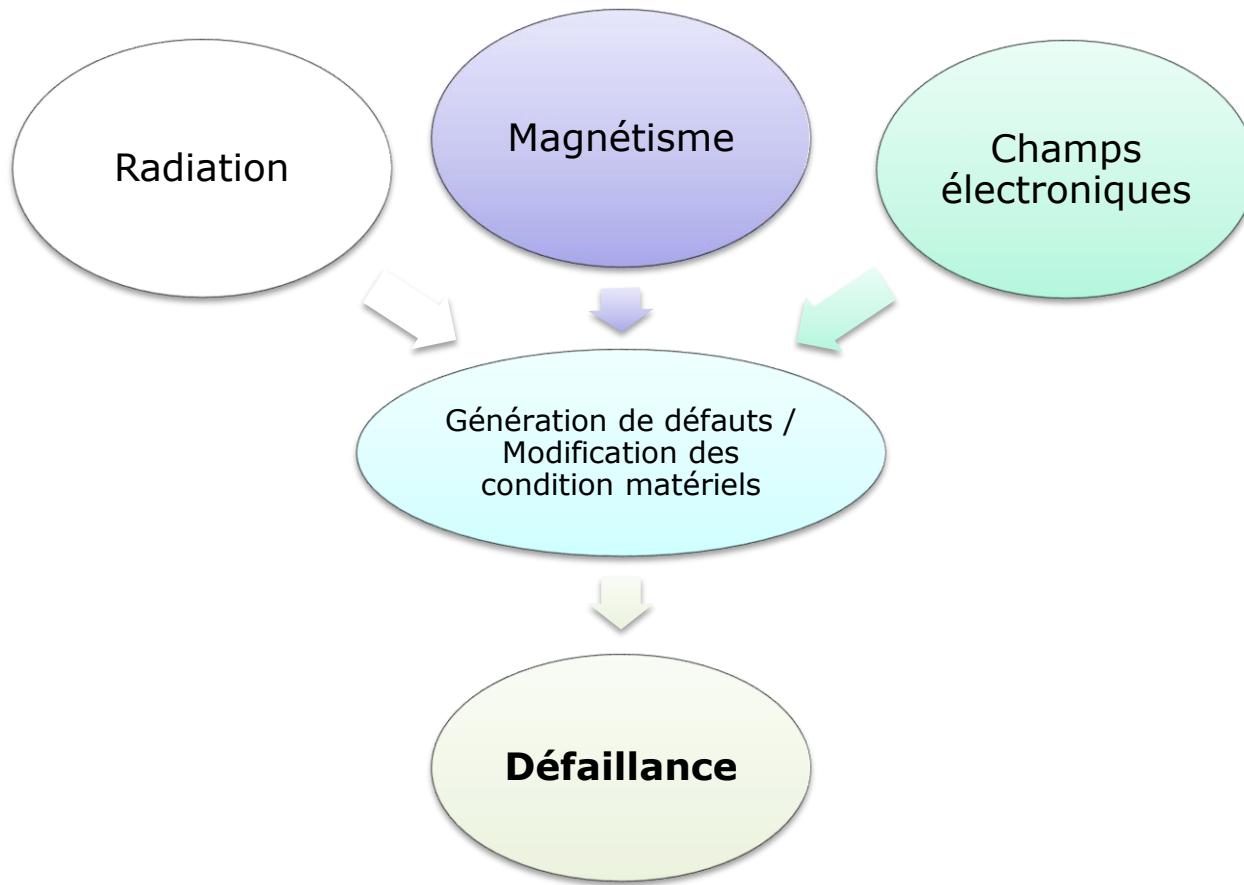
# Origine des défauts logiciels

- Causes d'une erreur :



# Origine des défauts logiciels

- Autres causes d'une défaillance (ex.) :



# *Pourquoi les tests sont-ils nécessaires ?*

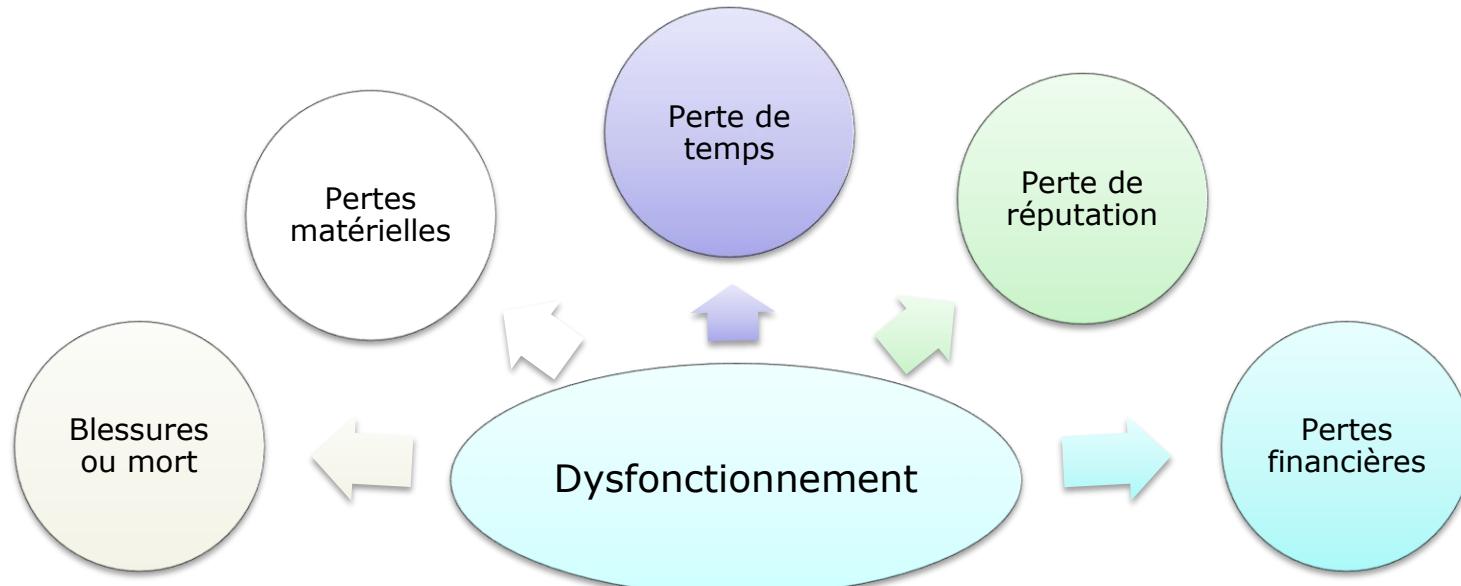
- Contexte des systèmes logiciels:
  - Les systèmes logiciels deviennent une part intégrante de notre existence

<b>Contexte</b>	<b>Problèmes</b>
Usage commercial (paiements, ventes...)	Erreurs de commandes ou de produits livrés Erreurs de paiement
Secteur de la santé (appareils d'analyses : scanners...)	Patients surexposés à des éléments dangereux Diagnostiques erronés
Grande consommation (voitures, machines à laver, jouets...)	Appareils mal-fonctionnant : peuvent générer des dommages aux objets, blesser des utilisateurs, causer des dégâts (surchauffe, court-circuit – feu...)

- La plupart d'entre nous a eu l'expérience d'un logiciel qui n'a pas fonctionné comme attendu

# *Pourquoi les tests sont-ils nécessaires ?*

- Conséquences possibles d'un dysfonctionnement :



- Exemples :
  - Lexus ES350, Ariane 5, SNCF, Système de santé anglais, ...

# Exercice

- Quelle conséquences pour quelle entreprise ? Donnez un exemple de bug possible.
  - Google
  - Facebook
  - Tesla
  - Apple
  - Amazon
  - Samsung

# Tests et Qualité

La norme ISO 9126 (caractéristiques de qualité des logiciels)

## Capacité fonctionnelle

- Conformité au besoin fonctionnel exprimé

## Fiabilité

- Disponibilité du système, tolérance aux pannes

## Facilité d'utilisation

- Facilité de compréhension, d'apprentissage

## Efficacité

- Performance du système et efficacité de l'utilisation des ressources matérielles

## Maintenabilité

- Facilité de l'évolution du logiciel

## Portabilité

- Facilité du portage d'une plateforme ou d'un environnement à un autre



Cognizant

Passion for building stronger businesses

# Tests et Qualité

Mesurer la qualité

- En terme de défauts trouvés pour chaque caractéristique ou exigence

Améliorer la qualité

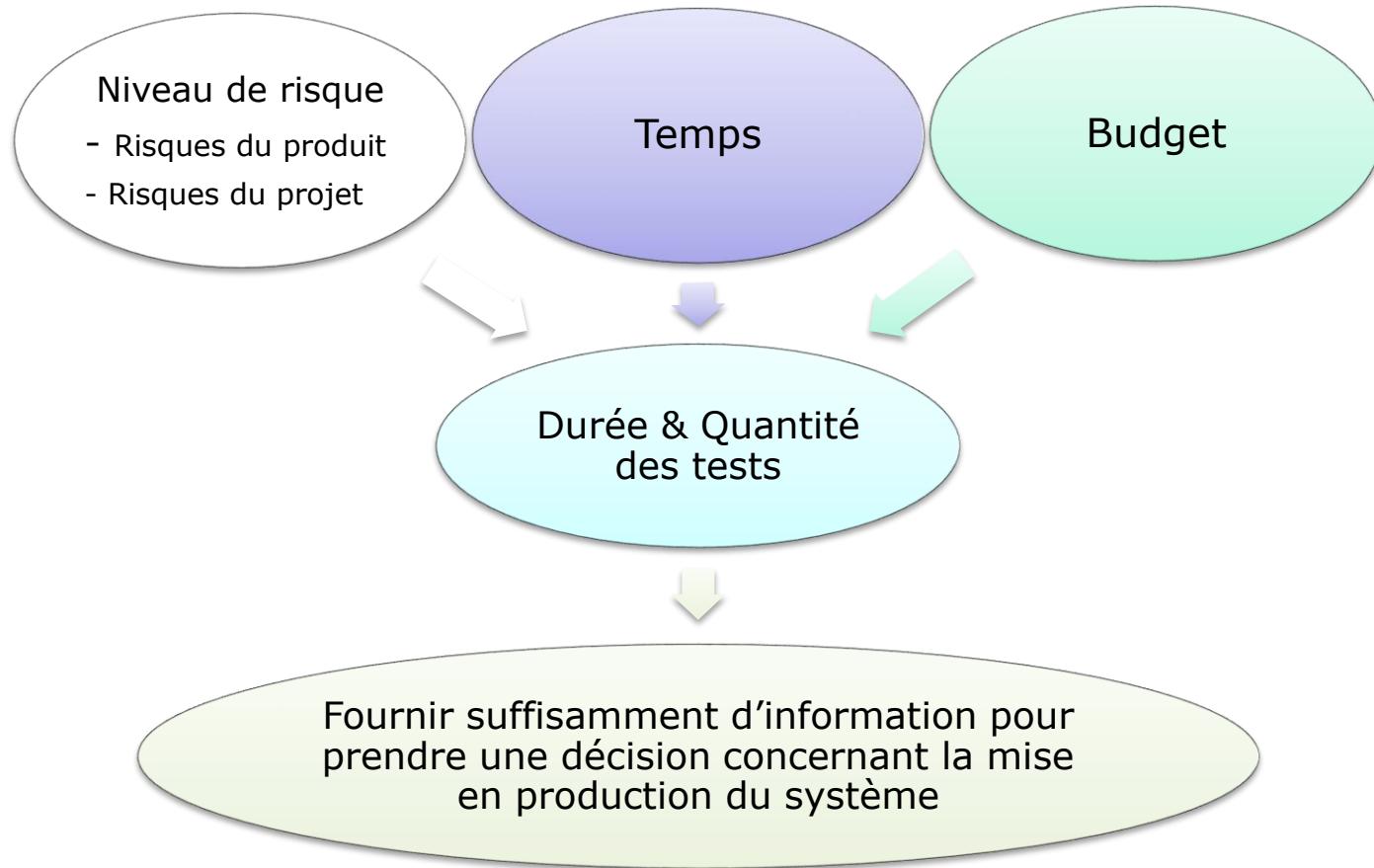
- Un défaut trouvé et corrigé augmente la qualité du système

Améliorer le niveau  
de confiance

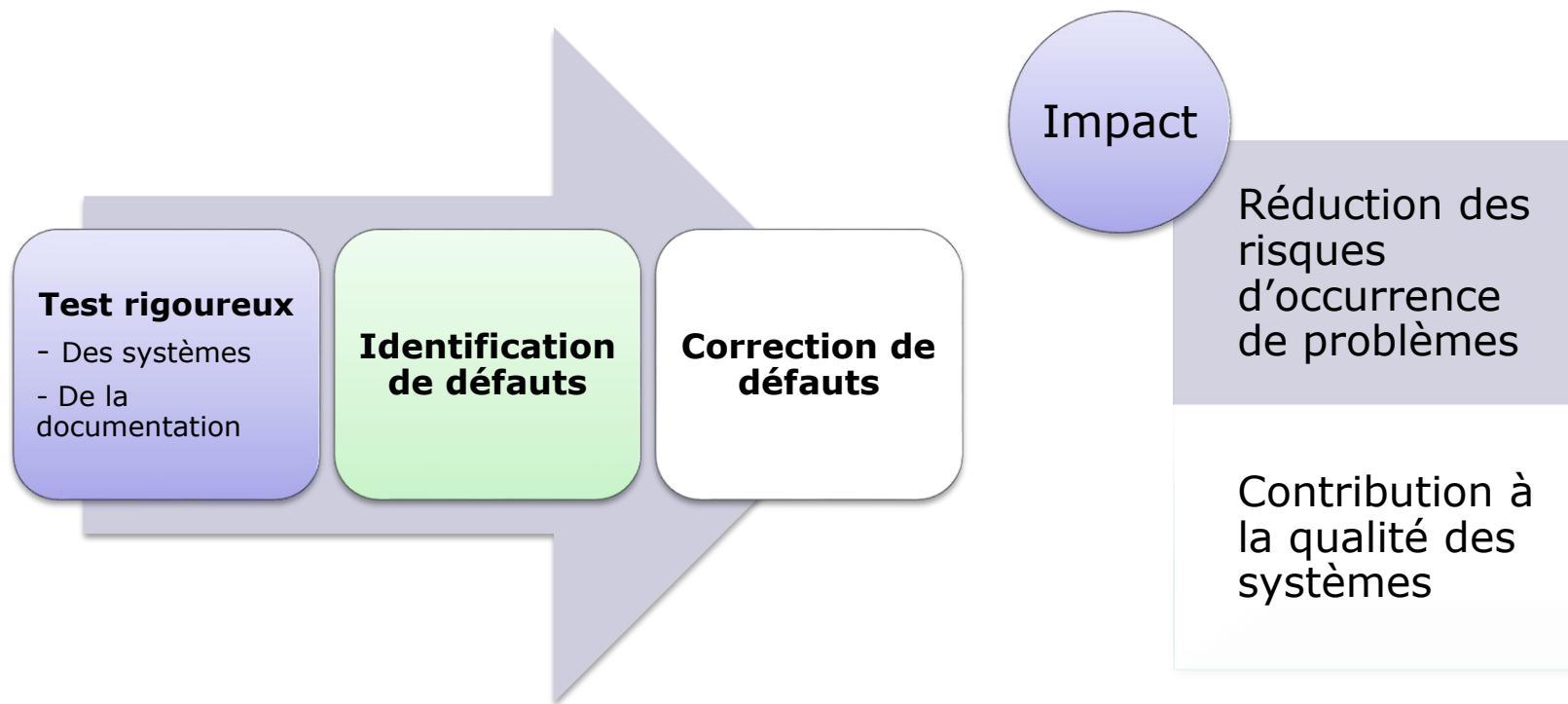
- Un test conçu correctement exécuté sans erreur réduit le niveau de risque général du système

# Tests et Qualité

Combien de tests sont suffisants ?



# *Rôle des tests dans le développement, la maintenance et l'opération des logiciels*

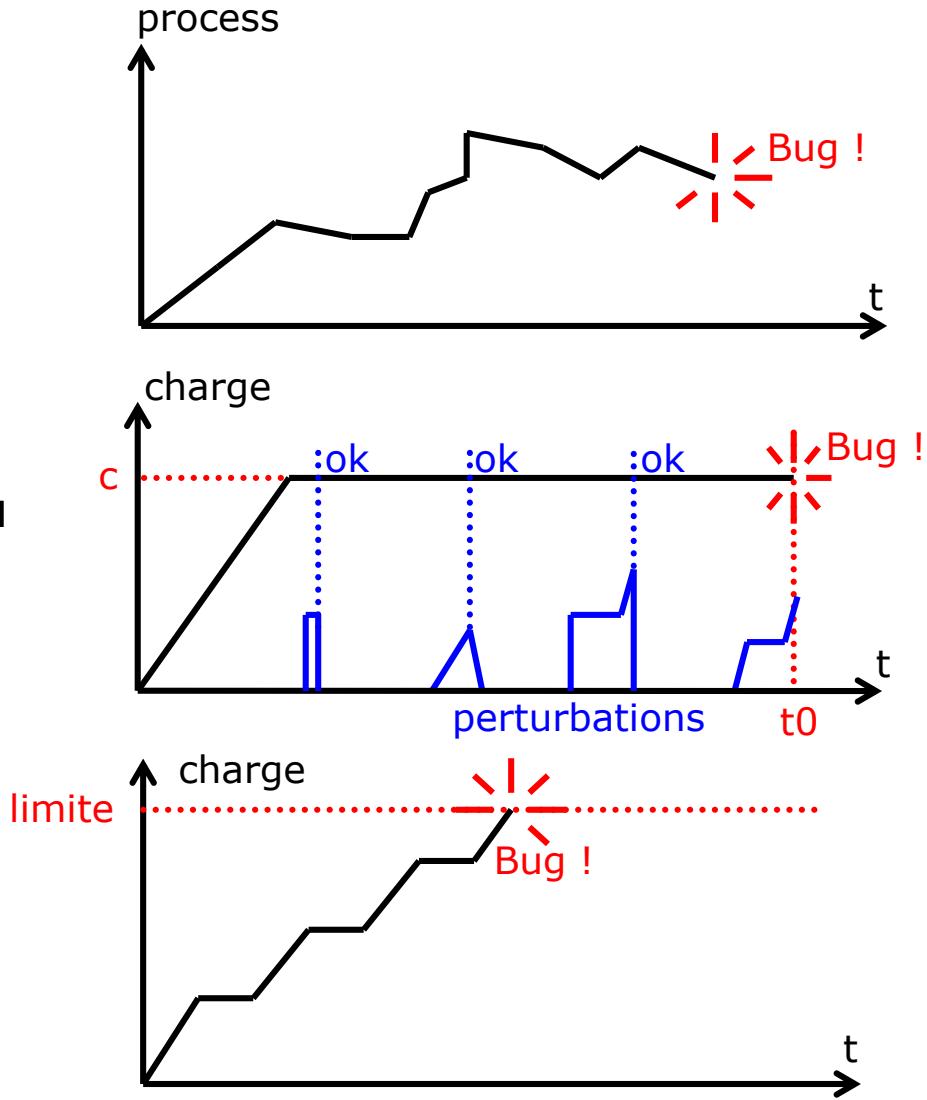


Autres rôles : Satisfaire

- Des exigences légales
- Des exigences réglementaires
- Des normes industrielles spécifiques

# Objectifs des tests

- Trouver des défauts :
- Contrôler la robustesse du système:
- Identifier les limites de fonctionnement :



# *Principes généraux des tests*

## Principe 1 : Les tests montrent la présence de défauts

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

# *Principes généraux des tests*

## Principe 2 : Les tests exhaustifs sont impossible

- Sauf dans les systèmes simples, il est impossible de « tout » tester.
- Les tests sont sélectionnés par rapport aux risques et aux priorités.

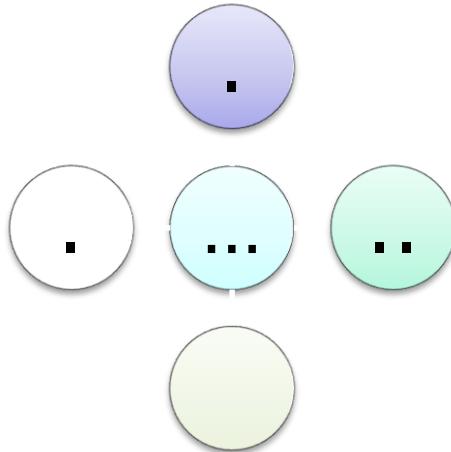
# *Principes généraux des tests*

## Principe 3 : Tester tôt

- Les activités de test doivent commencer tôt dans le cycle de vie de l'application.
  - Anticiper les activités de test (ressources, environnement...)
  - Valider la qualité des entrants (notamment spécifications)
  - Déetecter les défauts au plus tôt (permet de réduire le coût de correction des défauts)

# *Principes généraux des tests*

## Principe 4 : Regroupement des défauts



- Les défauts trouvés dans la phase de test ou en production sont souvent regroupés dans un nombre limité de modules.
  - Rappel : les causes d'erreur comme le manque de temps ou la complexité du système concernent des composants entiers ou s'appliquent souvent pendant la durée du travail sur un certain aspect du système.
- Localisation d'un défaut dans le logiciel

# *Principes généraux des tests*

## Principe 5 : Paradoxe du pesticide

- La répétition des mêmes tests devient inefficace, il est nécessaire de renouveler ses tests.
- Pour trouver de nouveaux défauts, enrichir les tests (par exemple sur la base des défaillances constatées en production), introduire de nouveaux types de test, changer la valorisation des données, l'ordonnancement des tests, etc.

# *Principes généraux des tests*

## Principe 6 : Les tests dépendent du contexte

- Les tests doivent être adaptés au contexte, notamment au risque du système.
- Exemple : plus de tests de sécurité pour un système qui traite des données critiques en temps réel, que pour un site web.

# *Principes généraux des tests*

## Principe 7 : L'illusion de l'absence d'erreurs

- Le but final d'un système est de répondre aux attentes des utilisateurs.
- Si leurs besoins ne sont pas pris en compte dans les spécifications, le fait de parfaitement et sans défaut implémenter ce qui est décrit ne résulte pas en un système de qualité.
- Une absence d'erreurs peut aussi indiquer un problème avec les tests : peu pertinents, utilisation d'un jeu de données non pertinent...

# *La psychologie des tests : Indépendance du test*

## Rôle du développeur

- Trouver des solutions pour les besoins des utilisateurs
- Implémenter correctement une solution
- Corriger des défauts éventuels

## Rôle du testeur

- Selon les objectifs du test, vérifier la conformité du produit aux besoins utilisateurs
- Signaler les écarts

Avantages de faire intervenir un testeur en plus des tests réalisés par le développeur

- Formé au métier du test
- Une autre perspective sur le produit :
  - Eviter le parti-pris
  - L'indépendance du testeur permet de trouver d'avantages d'erreurs

# *La psychologie des tests : Mentalité*

Développeur et testeur ont une mentalité et un état d'esprit différent

## **Le développeur**

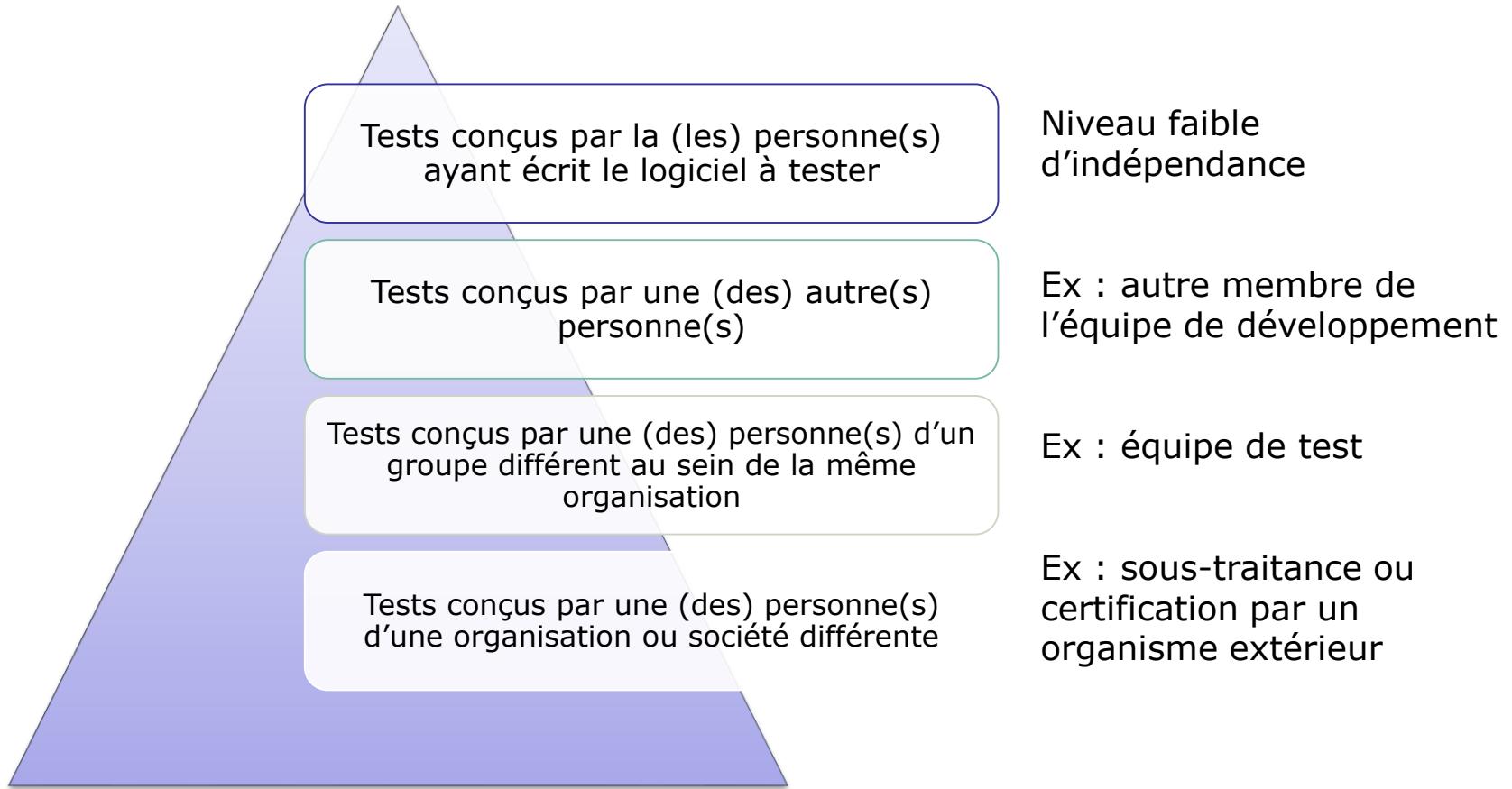
- Crée sur la base de son expérience, de sa compréhension des besoins utilisateurs
- Doit montrer que son développement fonctionne
  - *N'aime pas les erreurs (qui prouvent sa faillibilité)*

## **Le testeur**

- Doit anticiper les problèmes possibles sur la solution proposée
- Doit montrer que le système ne fonctionne pas comme attendu
  - *Aime les erreurs (capacité à générer un test efficace)*

# *La psychologie des tests : Indépendance du test*

## Niveaux d'indépendance



# *La psychologie des tests : Mentalité*

Communication et relationnel : interaction avec le développement

Réactions possibles d'une personne responsable de l'objet nécessitant une correction

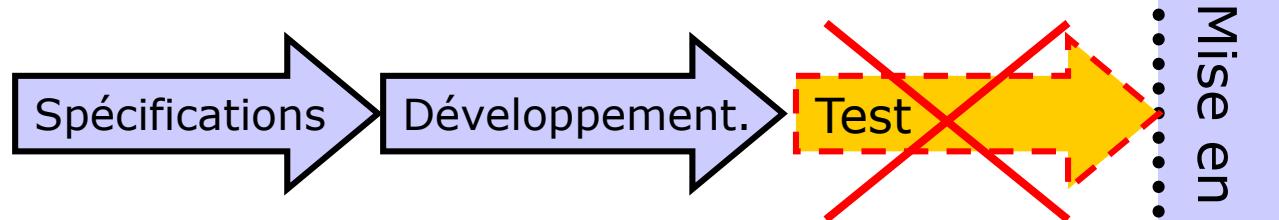
- Se sent vexée et critiquée personnellement
- Défend ou essaye de justifier son travail

Principaux moyens pour une communication constructive

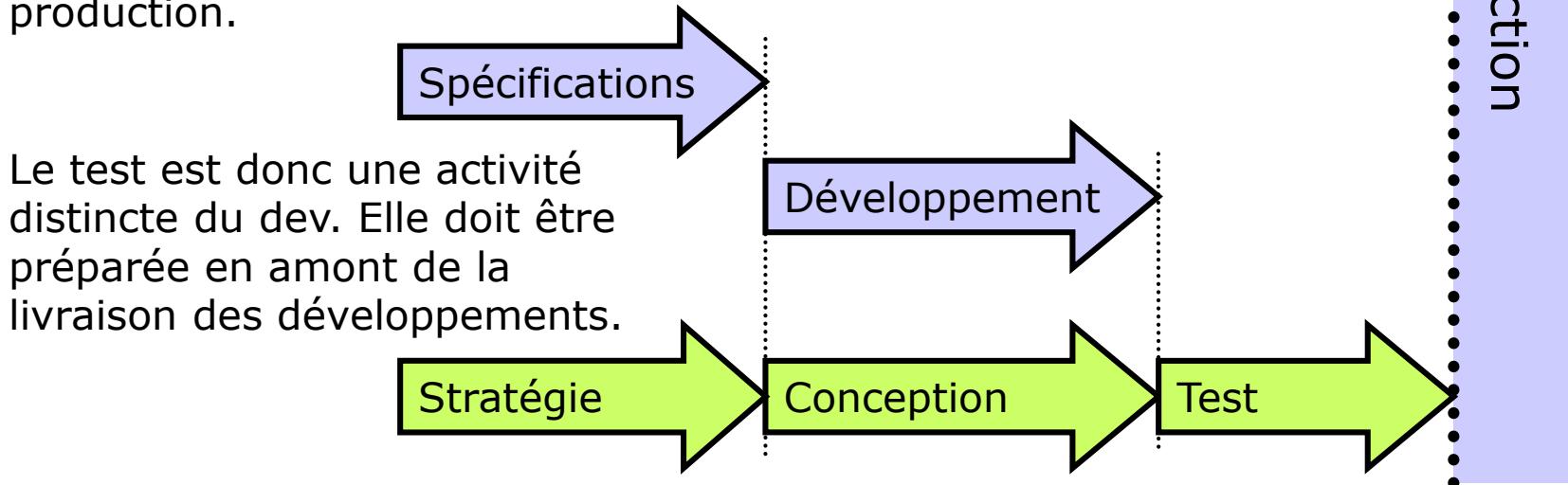
- Les équipes de test et de développement ne sont pas « adversaires », mais elles collaborent afin d'atteindre un meilleur niveau de qualité
- Expression factuelle des défauts dans le rapport d'incidents et autres communications

# *La psychologie des tests*

Le test logiciel n'est pas une phase terminale de contrôle du développement.



L'activité de test vise à vérifier l'ensemble de la qualité d'un projet avant sa mise en production.



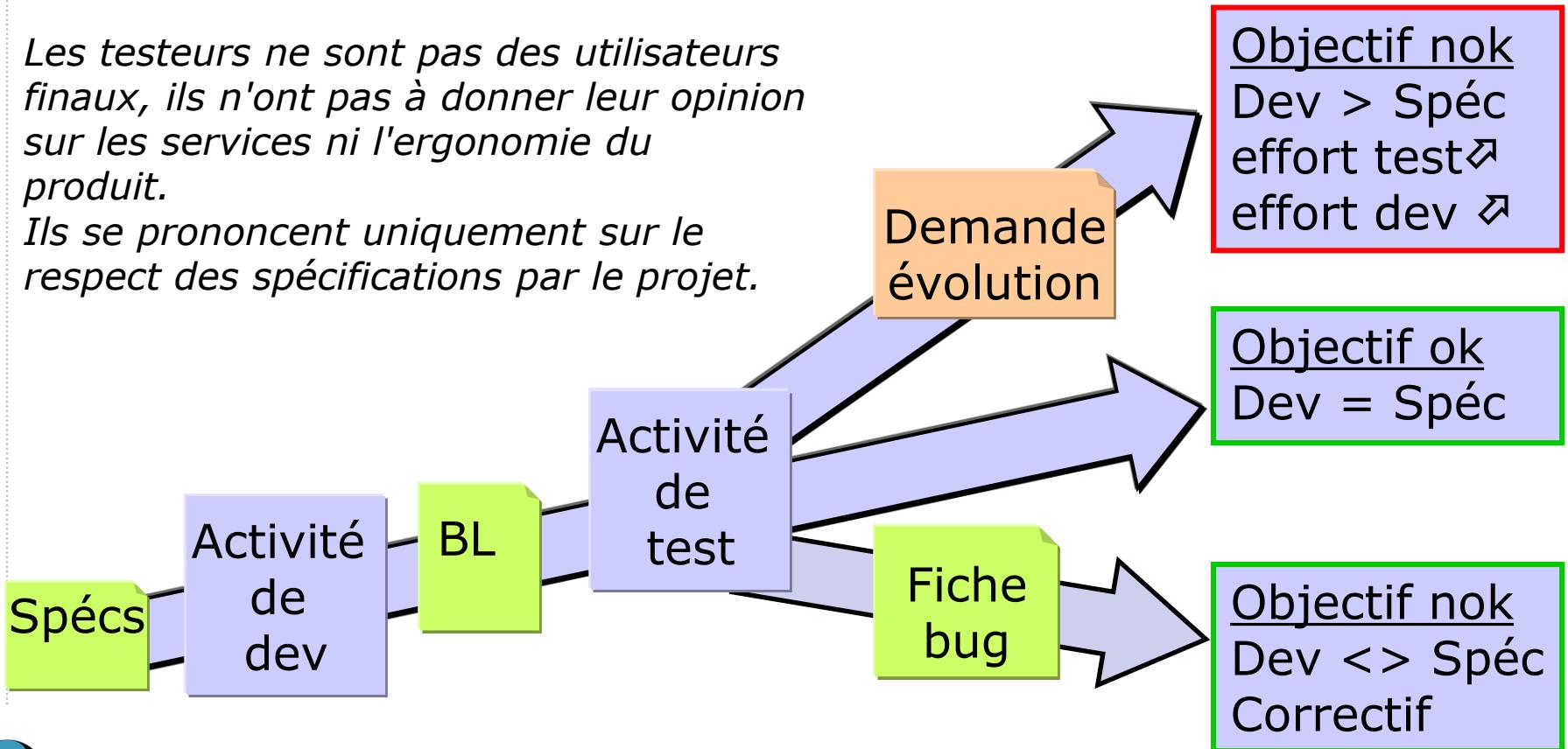
Le test est donc une activité distincte du dev. Elle doit être préparée en amont de la livraison des développements.

# La psychologie des tests

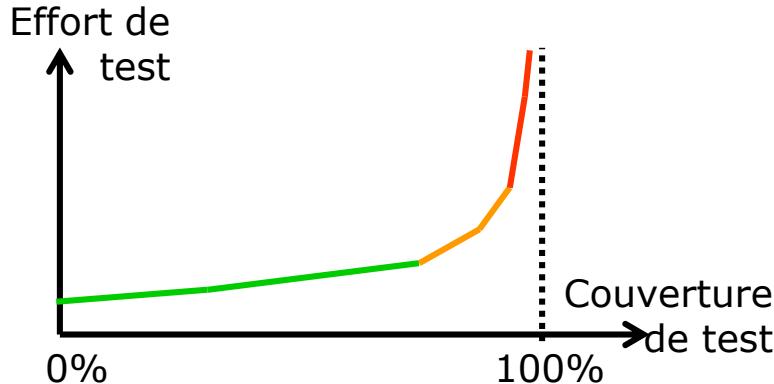
Ne pas oublier l'**objectif des tests** :  
⇒ Vérifier que les spécifications sont bien respectées.

*Les testeurs ne sont pas des utilisateurs finaux, ils n'ont pas à donner leur opinion sur les services ni l'ergonomie du produit.*

*Ils se prononcent uniquement sur le respect des spécifications par le projet.*



# La psychologie des tests



On ne peut jamais tout tester car alors l'effort de vérification complet d'une application est trop important :

- en coût (ressources, automatisation, environnement, complexité),
- en temps (mise en œuvre, exécution complète, time to market).

Et pourtant une anomalie non détectée sera d'autant plus chère à corriger qu'elle sera détectée tardivement.

Il faut alors aménager l'activité de test en suivant ces règles :

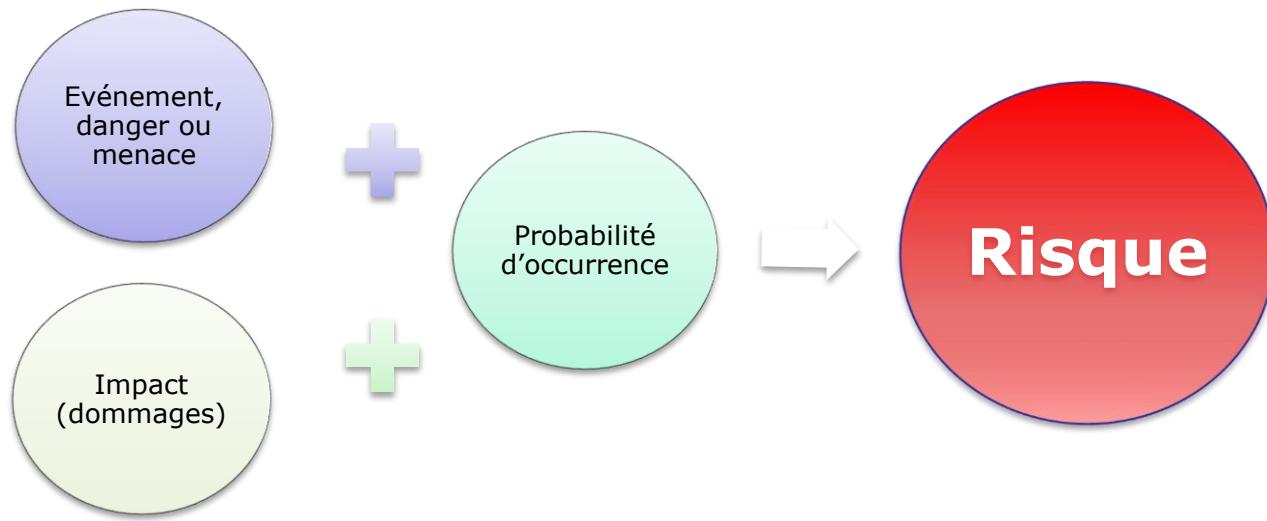
- ⇒ La couverture des tests à vérifier doit être fixée en début de projet.
- ⇒ Le niveau des exigences de test doivent être relativisées entre elles.
- ⇒ L'effort de conception de test sera fonction des exigences à vérifier.

# Sommaire

- Le métier du test
- Définition du test
- **Tests et risques**
  - » Introduction
  - » Tests et risques
  - » Gestion des tests basée sur le risque
  - » Evaluation des risques
  - » Réduction des risques
- Classification des tests
- La stratégie de test

# Tests et risques - Introduction

- Un risque est un danger potentiel pour le bon déroulement d'un projet.
- Un risque se définit selon 3 axes :



- Les tests sont une façon de contrôler le risque.
- Ils fournissent un retour sur le risque restant par la mesure de l'efficacité de l'élimination des défauts critiques.

# *Tests et risques – Risques de projet*

- Les risques de projet sont liés à l'exécution du projet.

## Problèmes d'acquisition

- Défaillance d'une tierce partie
- Problèmes contractuels

## Facteurs organisationnels

- Manque de savoir-faire et de personnel
- Problème de personnel et de formation

## Problèmes techniques

- Problème de définition des exigences
- Environnement de test indisponible
- Qualité

# Tests et risques – Risques de produit

- Les risques de produit sont liés aux futures défaillances potentielles du produit.

**Logiciel défectueux**

- Implémentation incorrecte

**Logiciel entraînant des dommages**

- Dommages humains
- Dommages matériels

**Qualité réduite ou insuffisante**

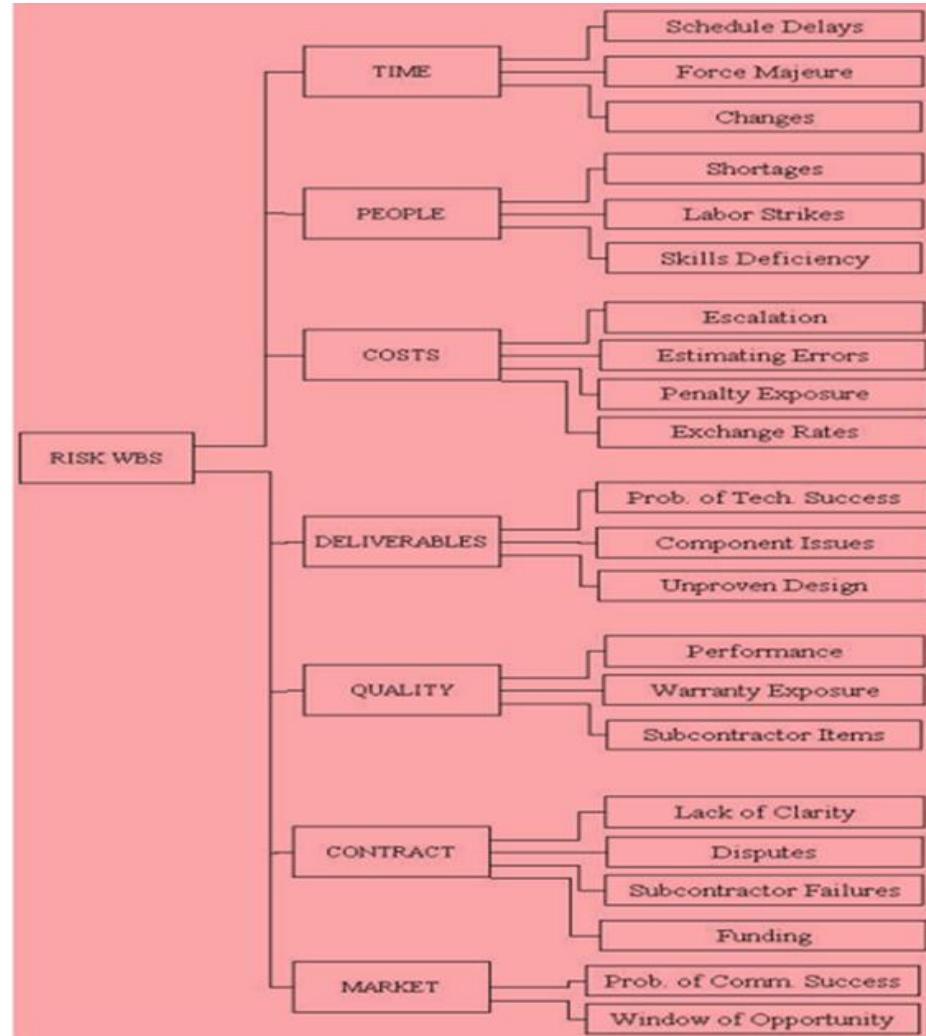
- Une fonction qui ne s'exécute pas toujours correctement

**Fonctionnalités non pertinentes**

- Une fonctionnalité qui ne correspond pas aux processus métier

# *Exemple de structure de risque*

Identification de risques  
avec leur source



# Gestion des tests basée sur le risque

## Analyse des risques

- L'analyse des risques permet une identification des risques dès les premières étapes – et tout au long – du projet.

## Approche de test

- La connaissance des risques est employée pour
  - Déterminer les techniques de test
  - Déterminer le volume du test à effectuer
  - Définir les priorités des tests afin de trouver les défauts critiques le plus tôt possible

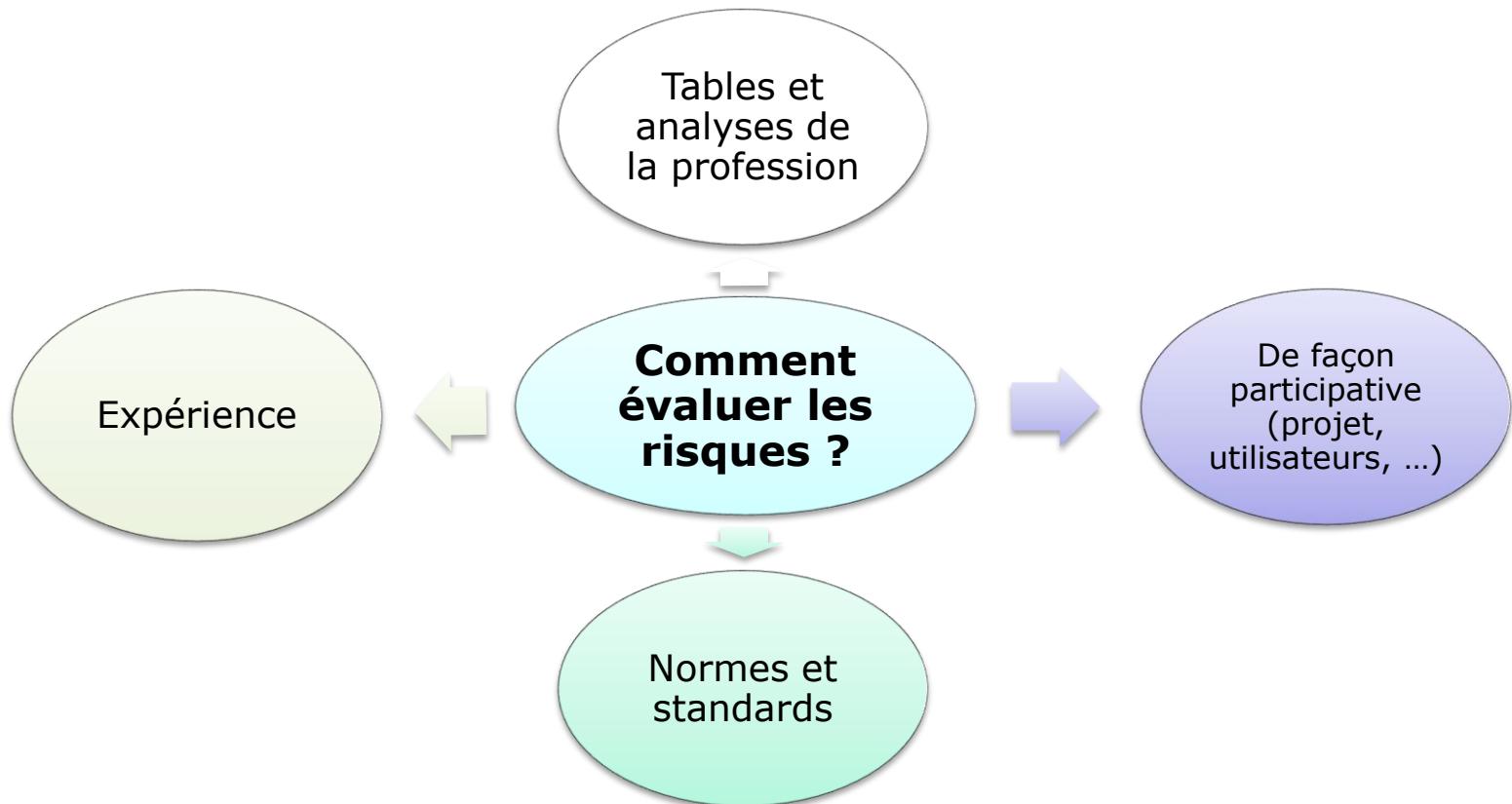
## Gestion des risques

- Estimer (et re-estimer régulièrement) ce qui peut faillir (les risques)
- Déterminer quels sont les risques importants à couvrir
- Entreprendre des actions pour couvrir ces risques

## Approche d'Assurance Qualité

- Une approche d'assurance qualité basée sur les risques identifie aussi des activités ne faisant pas partie du test qui pourraient aider à réduire les risques

# *Evaluation des risques produit*



# Réduction des risques produit

- Les risques peuvent être limités par :

## Annulation du risque

- Suppression des défauts (corrections)

## Réduction du taux d'occurrence

- Information de l'utilisateur (manuels, documentation)

## Réduction de l'impact

- Redondance de fonctionnalités ou de composants

# *Couverture de test basée sur les risques*

- Dans le cas d'une stratégie de test basée sur les risques, la couverture de test est faite à partir de la criticité des risques

Risque Critique

- Nombre de test élevé, pourcentage de réussite obligatoire

Risque Moyen

- Nombre de test moyen. Echantillonnage possible selon coût et délais

Risque Faible

- Nombre de test faible, optionnel.

# Risk Based Testing (RBT)

- Le risk based testing est une manière de planifier et d'organiser les test en fonction des risques



- Identification des risques à l'aide du métier, des documents et des experts.  
Préparation d'une liste de risques.



- Lors d'un workshop, classification des risques. Calcul des taux de probabilité et conséquences de chaque risque.



- Choix des objectif de test et des techniques de test. Documentation des tests.  
Calcul de priorité des tests par rapport aux risques



- Validation de la couverture et des budgets



- Mise en place des tests

# Exercice

- Enoncé du projet Symfony

Le site devra offrir un « front-office » permettant à des utilisateurs, connectés via un navigateur Web, de :

- Consulter le catalogue d'un commerçant : catégories d'articles, liste des articles d'une catégorie donnée, caractéristique d'un article (photo, prix, quantité en stock, etc...).
- Constituer un « panier » et passer commande en s'inscrivant sur le site.
- Revenir ensuite consulter l'état d'avancement de leurs commandes.

Le site devra également offrir un « back-office » permettant à un administrateur de gérer la base de données :

- Ajouter/supprimer des catégories et des articles
- Mettre à jour les stocks
- Valider les commandes en attente

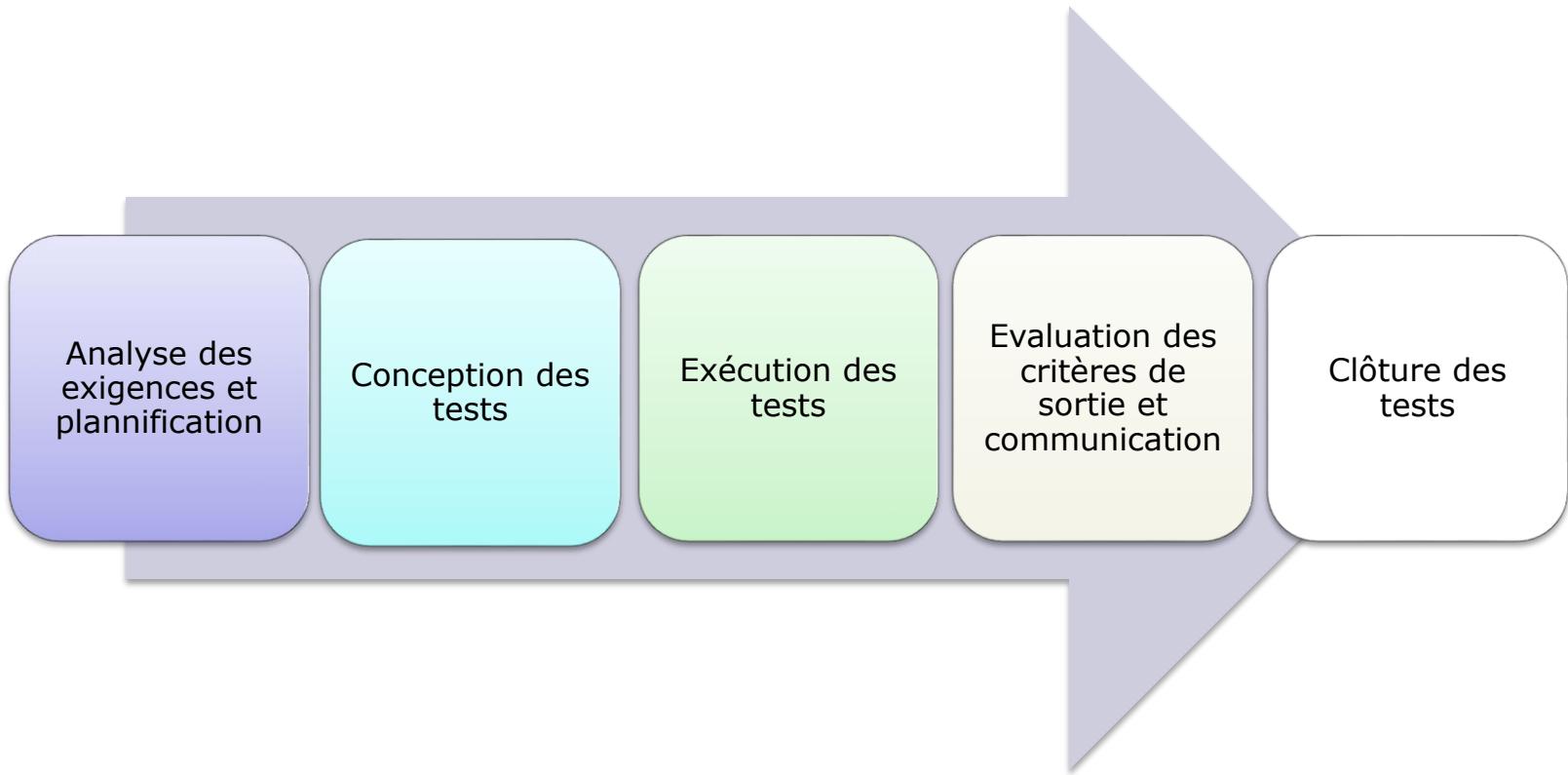
- Définissez plusieurs risques (au minimum 3), déterminez leur probabilité d'occurrence et priorisez les.

# Sommaire

- Le métier du test
- Définition du test
- Test et risques
- **Classification des tests**
  - » Le processus de test
  - » Les tests dans les projets
  - » Les méthodes de test
  - » Les niveaux de test
  - » Les différents types de test
  - » Types et niveaux de test
- La stratégie de test

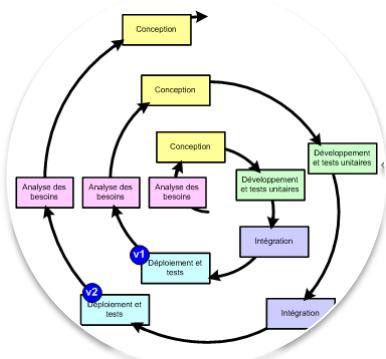
# *Le processus de test*

- Le test est un métier
- Le test est une activité fondamentale des projets
- Le test regroupe plusieurs activités :

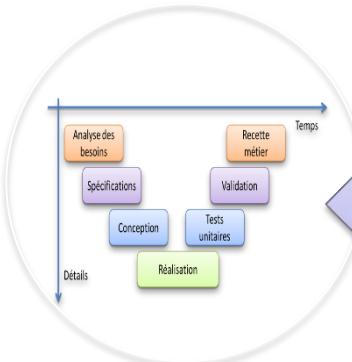


# Les tests dans les projets

Des modèles de développement logiciels

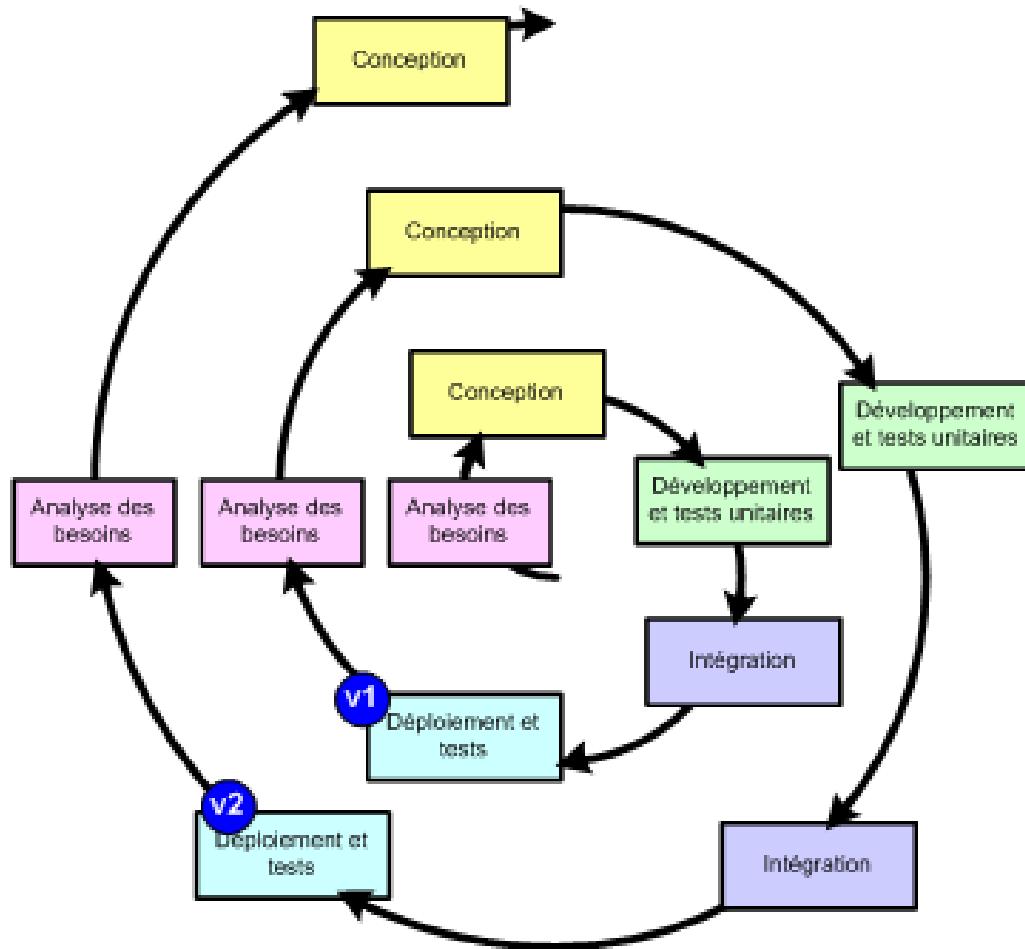


Modèle de développement itératif  
*Développement rapide d'applications,  
modèles de développement agiles*



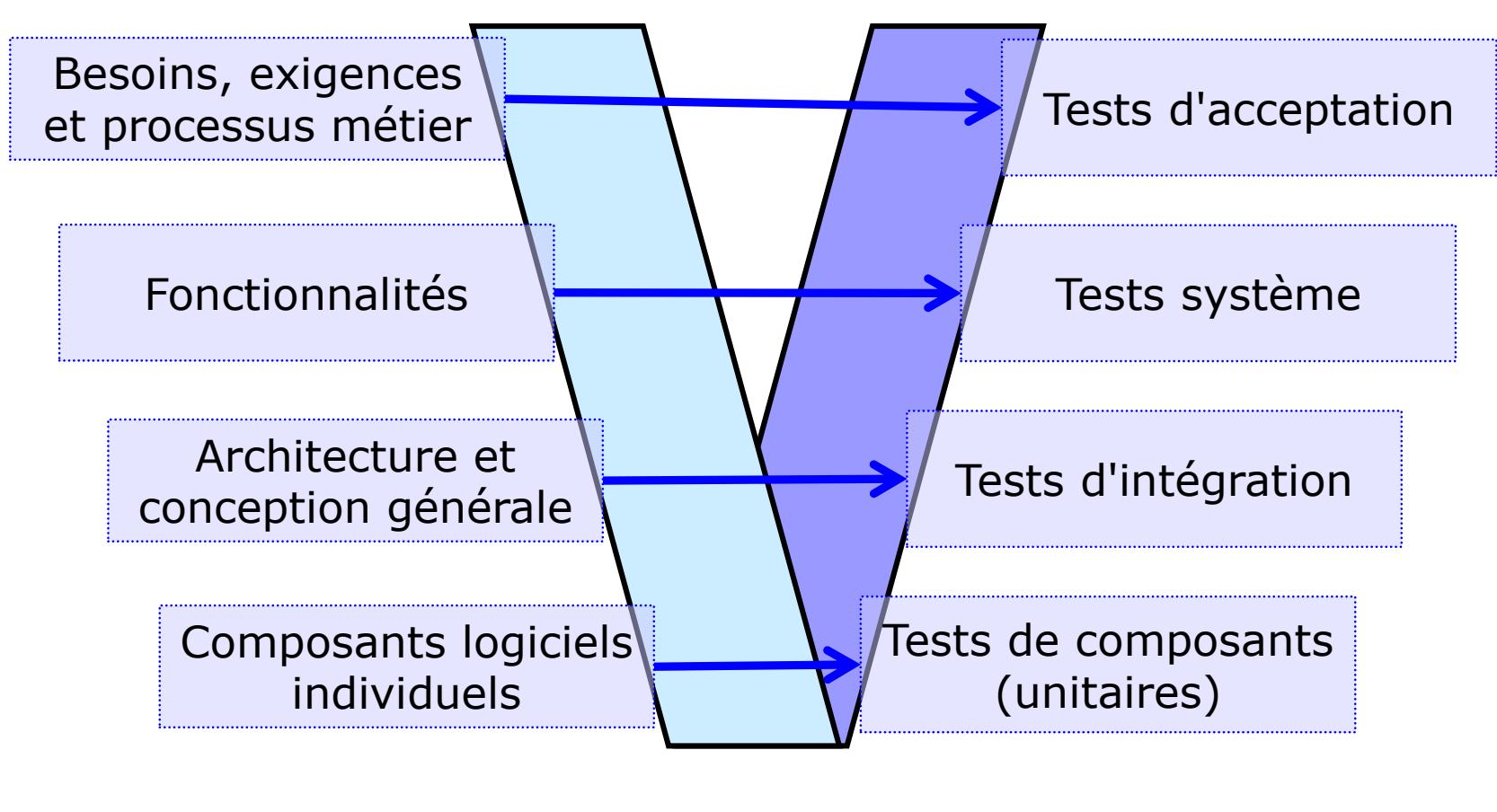
Modèle en V  
*Modèle de développement classique avec phases*

# Les tests dans les projets – Développement itératif



- Le développement se fait en plusieurs itérations
- Avec chaque itération, un incrément est produit qui s'ajoute au système existant
- Les tests de régression deviennent donc de plus en plus important à chaque nouvelle itération

# *Les tests dans les projets - Cycle en V*



# *Les niveaux de test*

## Tests de composants

- Identifier les défauts de fonctionnement du composant du système de façon isolée (avec bouchons ou simulateurs) ou intégrée

## Tests d'intégration

- Identifier des défauts dans les interfaces entre composants, modules, systèmes, ...

## Tests systèmes

- Vérifier les comportements du système afin de s'assurer qu'il réponde à des exigences spécifiques

## Tests d'acceptation

- Vérification de l'aptitude du système à accomplir des tâches requises
  - utilisateurs
  - opérationnelle
  - réglementaire

**Exemple de stratégie par niveau**

# *Les méthodes de test*

## Tests dynamiques

- Déroulement des tests dans l'application

## Revue des documents

- Révision d'un document avec le but d'identifier les défauts

## Analyse statique

- Analyse (outillée) du code ou des modèles de logiciel



# *Les différentes catégories de test*

Un type de tests est focalisé sur un objectif de tests particulier, pouvant être réalisé à tout niveau de test

## Tests fonctionnels

- Vérification des fonctions devant être effectuées par le logiciel

## Tests non fonctionnels

- Vérification des caractéristiques comme :
  - fiabilité
  - utilisabilité
  - robustesse
  - maintenance
  - portabilité
  - performance

## Tests structurels

- Vérification de la structure ou de l'architecture

## Tests liés aux changements

- Vérification par rapport aux corrections des défauts
- Vérification de l'absence de défaut dans les parties non modifiées du logiciel



# Type de test : Test passant / non passant

## Test passant

- Vérifier le comportement fonctionnel régulier
  - login avec succès : on passe sur le menu principal

## Test non passant

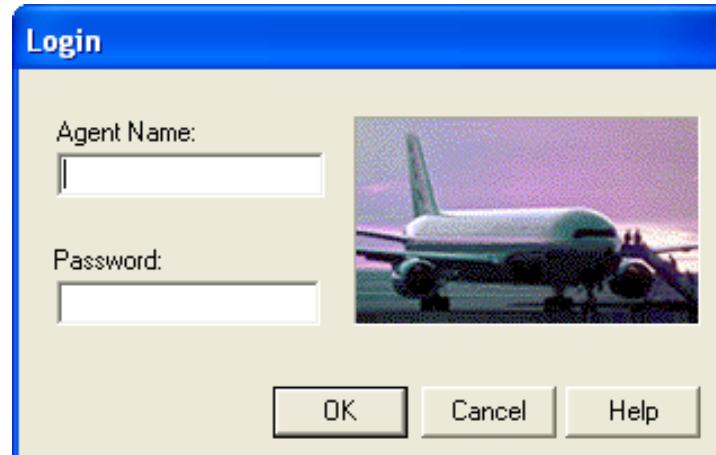
- Vérifier le comportement fonctionnel irrégulier
  - login avec échec : des messages d'erreur guident l'utilisateur pour reprendre le fonctionnement régulier



# *Test passant / non passant - Exercice*

## » Rédiger :

- Les cas de tests passants et non passants nécessaires pour vérifier la gestion du login à Flight :
  - Il faut un login et un mot de passe pour réaliser le login,
  - L'identifiant doit contenir quatre caractères,
  - Le mot de passe est "mercury".
- L'état initial est : application fermée.



# Type de test - Test de non régression

Entrées stables :

- manip
- jeu de données

Système qualifié en V1

Système qualifié en V2

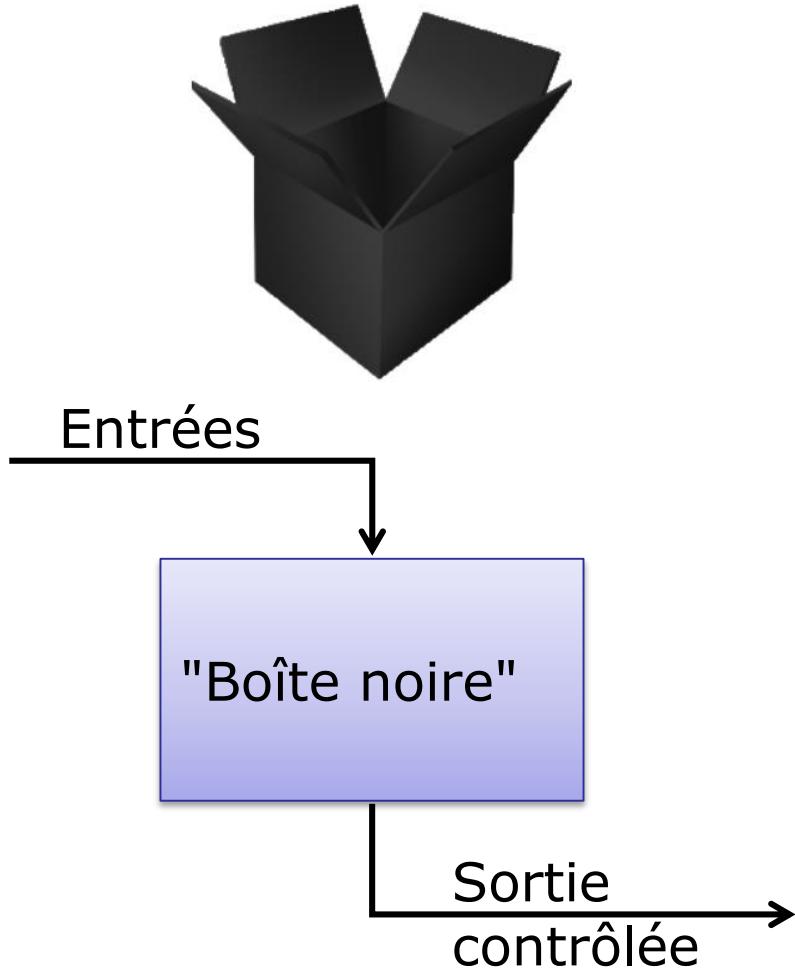
Sollicité par les mêmes entrées, un système dans deux versions différentes doit donner les mêmes résultats.

En cas d'écart, une régression est trouvée.

Sortie contrôlée (V1)

? =  
Sortie comparée (V2)

# Type de test - Test « boite noire »



Ce type de contrôle est recommandé en test fonctionnel afin de vérifier que les services attendus correspondent bien aux sollicitations en entrée du système.

La boîte noire est utilisé pour des systèmes simples (un écran de recherche) ou complexes (intégration de plusieurs modules applicatifs ensemble).

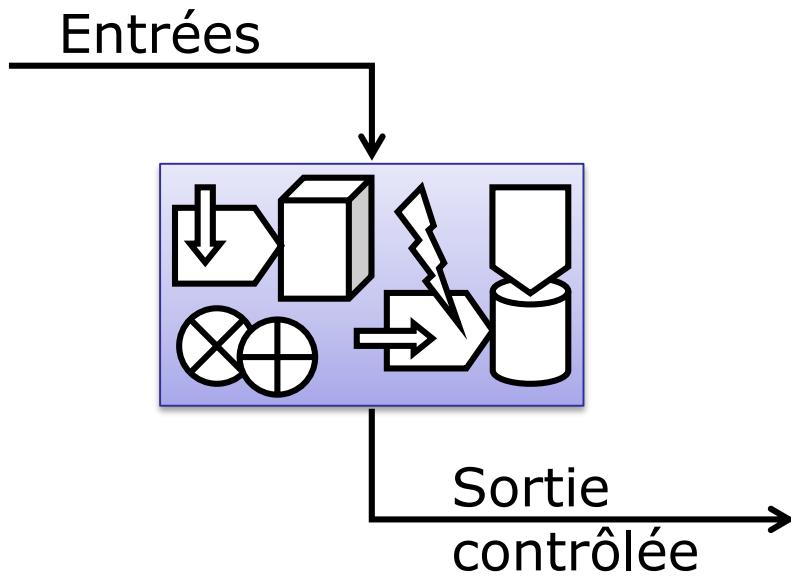
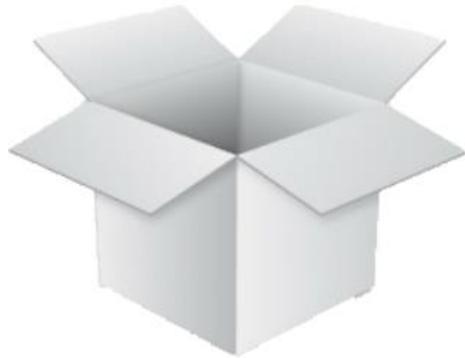
Avantages :

- Simplicité
- Rapidité
- Impartialité

Inconvénients :

- Superficialité
- Redondance

# Type de test - Test « boite blanche »



Ce type de contrôle est recommandé en test unitaire, d'intégration technique ou de performance afin d'analyser le comportement du système en fonction des sollicitations en entrée.

La boîte blanche est utilisée pour des systèmes complexes demandant des contrôles d'expertise technique (temps de réponse, gestion de flux réseau, correspondances d'interfaces).

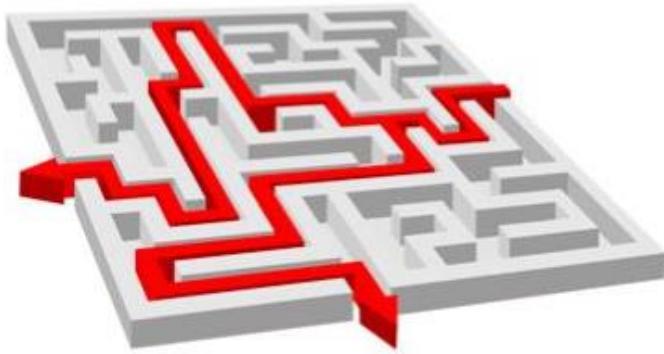
Avantages :

- Anticipation
- Optimisation
- Exhaustivité

Inconvénients :

- Complexité
- Durée
- Industrialisation
- Intrusion

# Type de test - Test de bout en bout

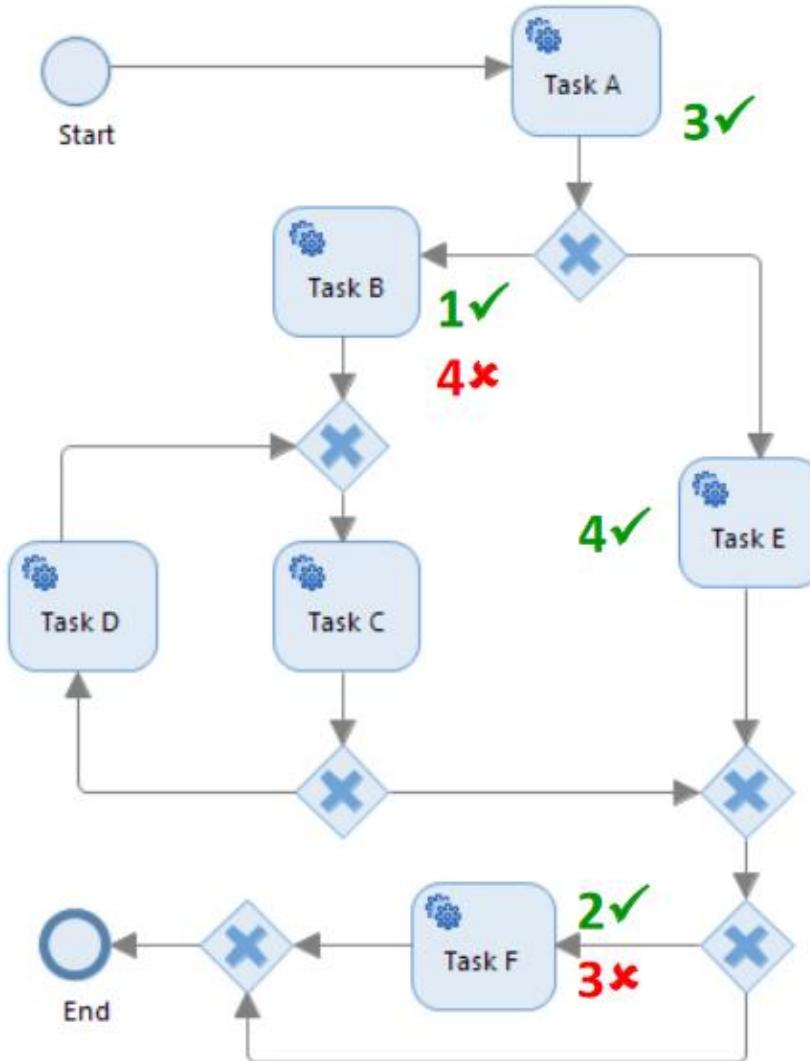


Avantages :

- Complets : grande couverture
- Correspond à des usages réel

Inconvénients:

- Complexes



# Types et niveaux de test

Type Niveau \ Test	Test passant	Test non passant	Test de régression	Boite noire	Boite blanche	Bout en bout
Test unitaire	X	X	X		X	
Test intégration	X	X	X	X	X	X
Test système	X	X	X	X		X
Test de performance	X	X	X		X	X
Test de sécurité	X	X	X		X	X
Test d'acceptation	X	X	X	X		X

# *Autre type de test : Le test exploratoire*

- Le test exploratoire est un type de test qui ne se base pas sur un scenario prédéfini. Il consiste à utiliser l'application de manière aléatoire en fonction des choix du testeur.
- C'est un type de test adapté à des utilisateurs avancés avec une certaine expertise dans le fonctionnel.
- **Avantages** : permet d'éviter le « paradoxe du pesticide »
- **Inconvénient** : nécessite une expertise et difficile à quantifier

# *Types et niveaux de test*

## **Résumé**

- On va retrouver les types et les niveaux de test dans un document de stratégie de test. C'est un moyen de définir un périmètre et établir les rôles et responsabilités de chacun.
- En fonction du type de test ou du niveau de test les techniques de conception, les méthodes et les outils utilisés vont être différent.

# Exercice

## Description

Tony veux ouvrir une pizzeria dans le centre ville de Grenoble. Il possède une vieille étude de marché et un business plan détaillé. Avant d'ouvrir son restaurant il a l'occasion d'éprouver son concept pendant un mois pour valider sa décision se lancer.

Aidez Tony à planifier ces activités pour le mois afin qu'il prenne la meilleure décision possible.

Tony possède un four à pizza dernier cri et il a investi dans une tablette pour prendre les commandes, directement liée à la caisse. Pour le reste toutes les hypothèses sont possibles.

## Problème

- Identifier les besoins de Tony.
- Identifier les risques associés.
- Identifier les différents éléments à tester pour couvrir les risques et imaginer des façon de tester.
- Identifier les types et niveaux de test à appliquer pour chaque élément/risque.
- Ecrivez au moins un test

# *Exercice : Besoins et Risques*

## **Besoins**

- ✓ Attirer des clients
- ✓ Satisfaire ses clients
- ✓ Etre rentable

## **Risques**

- Client pas intéressé
- Client pas satisfait
- Pas rentable
- Sécurité
- Normes sanitaires

# Strategie

## Les principaux Eléments du restaurant

- ✓ Salle (décoration, disposition, ...)
- ✓ Caisse
- ✓ Tablette
- ✓ Four
- ✓ Carte (Prix, Recettes, horaires, ...)

## Risques

### •Client pas intéressé

- Déco : tester différente déco
- Affluence : tester différent horaires/prix
- Pub : tester retour sur différente pub

### •Client pas satisfait

- Tester Temps pour fournir une pizza en moyenne. Test bout en bout
- Tester places dans le restaurant en fonction de la configuration
- Tester recette de pizza

### •Pas rentable

- Test de prix
- Test sur nombre de salariés

### •Sécurité

- normes sanitaire : tester la propreté toutes les semaines
- normes sécurité : Se mettre en situation, tester une utilisation du four maximum

# Types et niveaux de test

Type Niveau \	Test passant	Test non passant	Test de régression	Boite noire	Boite blanche	Bout en bout
Test unitaire	Caisse Four Tablette	Caisse Tablette		Caisse Tablette Four	Carte Salle	
Test intégration	Caisse + Tablette	Caisse + Tablette		Caisse + Tablette		
Test système						Salle Four Tablette Caisse
Test de performance /sécurité			Salle	Four		
Test d'acceptation			Carte			Salle Four Tablette Caisse

# *Types et niveaux de test*

**Test de disposition de la salle** : verifier que la disposition prevue peut etre mise en place et qu'elle donne satisfaction (nombre de places, encombrement, ...)

## **Test unitaire - boite blanche**

- Step 0 : Mettre en place le paramétrage attendu
- Step 1 : Vérifier que le paramétrage est correct
- Step 2 : Vérifier que le nombre de place maximum est bien celui attend
- Step 3 : Vérifier l'encombrement de la salle

# *Types et niveaux de test*

**Test de temps de service:** Vérifier la chaîne de mise à disposition d'une pizza

## **Test système – bout en bout**

- Step 0 : Mettre en place le paramétrage attendu
- Step 1 : Vérifier que la prise de commande fonctionne avec la tablette. Vérifier le temps de prise de commande
- Step 2 : Vérifier que la commande est bien communiquée au four
- Step 3 : Vérifier le temps pour préparer la pizza
- Step 4 : Vérifier le temps pour servir la pizza
- Step 5 : Vérifier le fonctionnement du paiement

# Sommaire

- Le métier du test
- Définition du test
- Test et risques
- Classification des tests
- **La stratégie de test**
  - » Pourquoi ?
  - » Enjeux
  - » Choix
  - » Approches

# *La stratégie des tests : Pourquoi ?*

## Les enjeux

- Temps limité pour effectuer les tests
- Budget contraint
- Pool de ressource avec les bonnes compétences limité
- Nombre important de cas de test à réaliser
- Améliorer la qualité de l'application

## Objectifs d'une stratégie de test

- Efficacité : Trouver les tests les plus impactant / critiques le + tôt possible (limiter les itérations)
- Coût : Réduire le temps nécessaire au test sans compromettre la qualité de l'application
- Time to market : Assurer le temps de mise sur le marché, voir même le diminuer

# *La stratégie de test : Enjeux*

Voici un cas typique, où un produit stratégique doit être lancé par une compagnie dans un délai restreint. La commercialisation ("time to market") et la qualité du produit sont les conditions impératives qui permettront à cette compagnie de pouvoir se maintenir et rivaliser avec ses concurrents.



Si nous réduisons encore le temps de développement, la qualité de l'application va s'en ressentir. Est-ce que l'équipe QA peut nous aider à réduire le "time to market" ?

Chers collègues, étant donné que nous sommes pris par le temps, comment pouvons-nous nous assurer que nous allons livrer un produit de bonne qualité en si peu de temps

Je souhaite que l'on puisse trouver une solution nous permettant de réduire considérablement l'effort de test sans compromettre la qualité de l'application.

## *Les contraintes du marché*

- ➊ Temps limité pour les activités liées aux tests à cause des besoins de commercialisation
- ➋ Contraintes financières qui amènent à diminuer le budget alloué aux tests
- ➌ Difficultés à trouver des testeurs compétents
- ➍ Des tests efficaces demandent d'avoir de nombreux cas de tests associés à un effort de test important
- ➎ Maintenir le niveau de qualité de l'application

# *La stratégie de test : Choix*

<b>Solutions</b>	<b>Avantages</b>	<b>Inconvénients</b>
Réduire le volume de test	<ul style="list-style-type: none"><li>• Réduction de la période de test et son coût</li></ul>	<ul style="list-style-type: none"><li>• Faible confiance dans le produit</li><li>• Criticité business pas prise en compte</li></ul>
Augmenter le nombre de ressources afin de conserver le volume de test	<ul style="list-style-type: none"><li>• Réduction de la période de test</li><li>• Qualité maîtrisée</li></ul>	<ul style="list-style-type: none"><li>• Disponibilité et compétences des ressources</li><li>• Augmentation du coût des tests</li></ul>
Optimiser le nombre de cas de test (en utilisant des méthodes statistiques de compression)	<ul style="list-style-type: none"><li>• Réduction du coût et de la période de test</li></ul>	<ul style="list-style-type: none"><li>• Processus complexe</li><li>• Criticité business pas prise en compte</li></ul>
Maximiser le taux d'automatisation	<ul style="list-style-type: none"><li>• Réduction de la période de test et du coût des futurs cycles de test</li></ul>	<ul style="list-style-type: none"><li>• Tous les scénarios ne peuvent pas être automatisés</li><li>• Augmentation de coût de la première phase de test</li><li>• ROI atteint après 2 ou 3 cycles de test</li></ul>
Risk Based Testing après étude des priorités business	<ul style="list-style-type: none"><li>• Réduction du coût et de la période de test</li><li>• Alignement des criticités business avec les contraintes projet</li></ul>	<ul style="list-style-type: none"><li>• Succès dépendant d'un effort continu d'analyse de priorités</li><li>• Disponibilité des experts applicatifs et business</li></ul>

# Approches de test

- Les approches de tests sont aussi appelées « stratégies de test »
- Classification des approches :

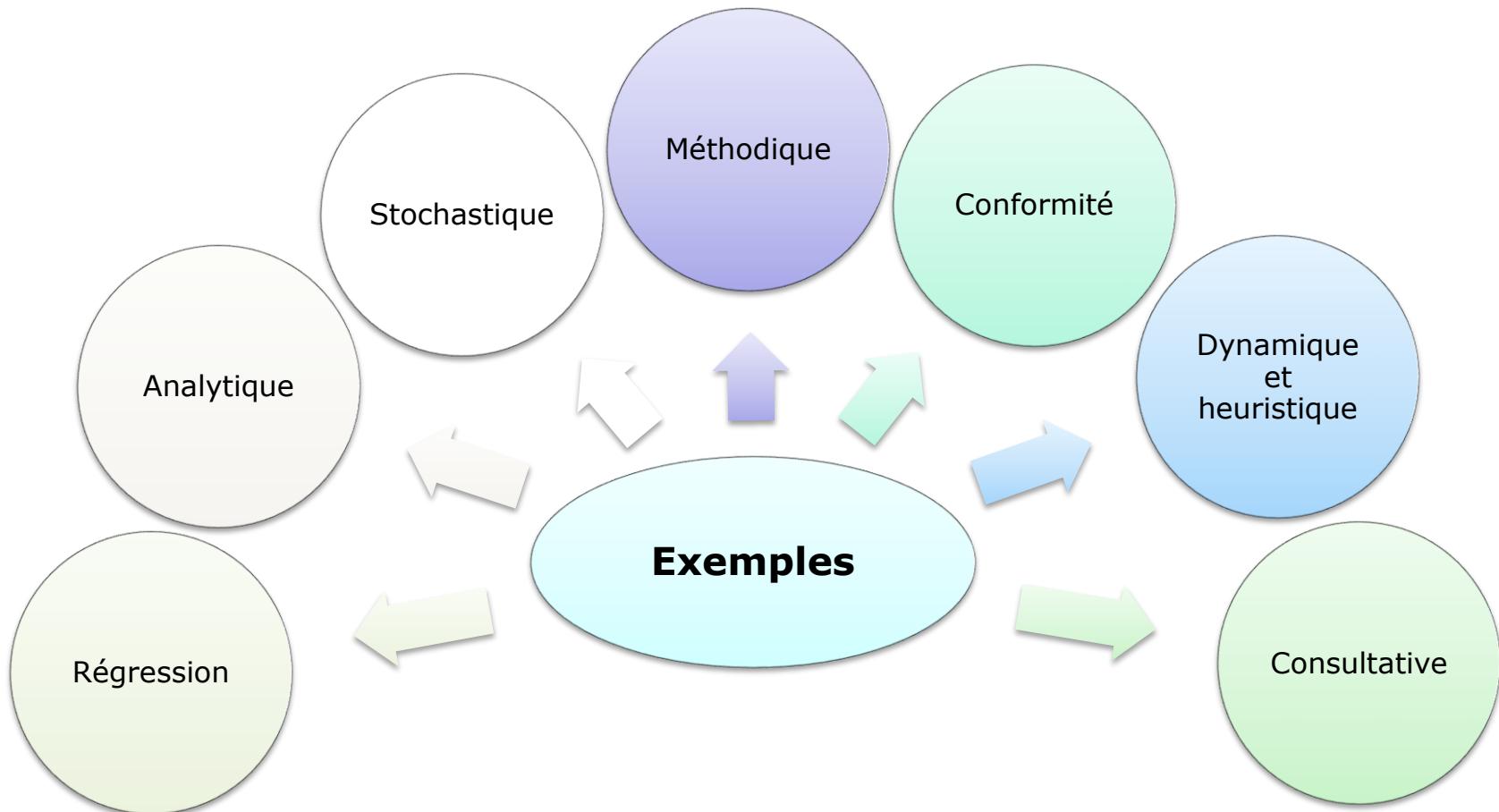
## Stratégie Préventive

- La conception des tests débute le plus tôt possible dans le cycle de vie du développement

## Stratégie Réactive

- La conception des tests commence lorsque le logiciel a été produit

# Approches de test



# *Plan de test*

## Plan de test

- Document décrivant l'étendue, l'approche, les ressources et le planning des activités de test prévues. Il identifie entre autres les éléments et caractéristiques à tester, qui fera chaque tâche, le degré d'indépendance des testeurs, l'environnement de test, les techniques de conception des tests et les techniques de mesure des tests à utiliser, et tout risque nécessitant un plan de contingence.

# Plan de test

Exemple de contenu d'un plan de test (selon IEEE 829)

Objet	Définition
Id	Identifiant du document
Introduction	Résumé des éléments à tester et références aux documents
Périmètre de test	Détail des éléments de test et infos sur les versions, références à la documentation
Fonctionnalités à tester	Fonctions et combinaison de fonction à tester
Fonctionnalités hors périmètre	Eléments hors du périmètre
Approche	Type d'approche utilisée, catégorisation de test, technique de conception de test, ...
Critère d'acceptation ou de rejet	Critère définissant si l'élément à tester a passé le test avec succès ou pas
Critères suspensifs	Condition où les tests peuvent être suspendus

# Plan de test

Exemple de contenu d'un plan de test (selon IEEE 829)

Objet	Définition
Livrables	Plan de test à livrer, procédures de test à produire, PV de test, ...
Liste des environnements	Caractéristiques spécifiques des environnements de test : outils, versions, documentation, interfaçage, bouchonnage, ...
Responsabilités	Equipes responsables
Staffing et besoin de formation	Profils requis pour l'exécution des activités et besoin de formation
Calendrier	Jalons
Risques	Risques et gestion de ces risques (ex. retard de réception des éléments de test)
Approbation	Définir les personnes autorisées à valider les livrables.

# Exercice

## Ecrire un plan de test pour le projet Symfony

- Périmètre
- Fonctionnalité à tester
- Fonctionnalité hors scope
- Type d'approche
- Critères d'acceptation
- Critères suspensifs
- Livrables
- Environnements

# Résumé

- Le test est une profession à part entière qui devient de plus en plus importante
- La principale raison de ce développement sont les dégâts considérables engendrés par certains erreurs.
- Le monde étant de plus en plus dépendant des SI, ces erreurs sont plus fréquentes.
- Les test peuvent être classifiés en fonction des projets et des contextes
- Les choix des test doit être réfléchi dans une stratégie et les risques sont un facteur important à prendre en compte



Cognizant  
Passion for building stronger businesses

## Questions / réponses



[olivier.octobre@cognizant.com](mailto:olivier.octobre@cognizant.com)

