

UV LO22 - Méthodes de Test et de Vérification du Logiciel



Mohamed Sallak – Walter Schön
Université de Technologie de Compiègne

Cours 1: Introduction et concepts de base des processus de test et de vérification du logiciel

Plan du cours 1

I. Introduction

II. Faute, Erreur et Défaillance

III. V&V

IV. Evaluation des capacités de tests et des compétences de votre entreprise



Introduction

Introduction

Question 1 : Qu'est-ce que ça veut dire que "tester un logiciel" ?

- Vérifier qu'il est conforme, par rapport à quoi ?
Suivant quels critères ?
- Tentative de définition « Processus d'exécution d'un programme avec l'intention de détecter des anomalies dans le but de le valider. »

Introduction

- Bien que les tests peuvent évaluer **l'exactitude** des logiciels sous certaines hypothèses spécifiques, les tests ne peuvent pas identifier tous les défauts dans le logiciel.
- Les tests fournissent une critique ou une comparaison qui compare l'état et le comportement du logiciel par rapport à des principes ou des mécanismes établis.

Introduction

- Définition adoptée par la communauté scientifique:

Software testing is an investigation conducted to provide stakeholders with information about the quality of the software under test.

Introduction

Question 2 : Y a-t-il plusieurs types de tests ?

- **Tests fonctionnels ou structurels.**
- **Tests statiques ou dynamiques.**

Introduction

Question 3 : Y a-t-il une alternative aux tests ?

- Méthodes formelles ?????

Question 4 : Quel est le coût du test d'un logiciel ?

- 30% à 40% ? des coûts de développement d'un logiciel suivant son niveau de criticité.

Introduction

- **Les logiciels sont le produit d'un processus mis en œuvre par l'humain d'où l'existence d'anomalies.**
- **Des bogues avec des conséquences désastreuses ...: Ariane 5, Velsatis (Voir document joint), ...**

Pour aller plus loin ...

- L'**ISTQB** (en anglais « *International Software Testing Qualifications Board* ») est le *Comité international de qualification du test logiciel*.
- L'ISTQB propose de valider le titre de testeur certifié ISTQB, une qualification standardisée pour le [test logiciel](#).
- Le comité de l'ISTQB est composé de 47 membres représentant plus de 71 pays.

Pour aller plus loin ...

L'ISTQB propose trois niveaux de certification :

➤ Le niveau fondamental :

- "Testeur Certifié" : Notions fondamentales - Tester tout au long du cycle de vie logiciel - Techniques statiques - Techniques de conception des tests - Gestion des tests - Outils de test
- "Extension AGILE" : Approche spécifique du test Agile en accord avec le Manifeste - Valeurs et principes des projets Agiles - positionnement du testeur dans l'équipe projet
- "Extension Model Based Testing" : En cours de Finalisation

Pour aller plus loin ...

- Le niveau avancé contient trois modules avec chacun un examen et une certification distincte
 - "Manager de tests" : Processus de test - Gestion des tests - Revues - Gestion des défauts - Amélioration du processus de test - Outils de test et automatisation - Gestion des compétences de test
 - "Analyste de tests" : Processus de test - Gestion des tests - Techniques de test - Test des caractéristiques de la qualité logicielle - Revues - Gestion des défauts – Outils
 - "Analyste technique de tests" : Test orienté risque - Test basé sur la structure - Techniques analytiques - Caractéristiques qualité du test technique - Revues - Outils et automatisation

Pour aller plus loin ...

- Le niveau expert : contient quatre modules avec chacun un examen et une certification distincte: « l'amélioration du processus de test », « le management du test », « automatisisation de test » et le « test de sécurité ».

Exemple d'un test ISTQB

The difference between re-testing and regression testing is

- re-testing is running a test again; regression testing looks for unexpected side effects
- re-testing looks for unexpected side effects; regression testing is repeating those tests
- re-testing is done after faults are fixed; regression testing is done earlier
- re-testing uses different environments, regression testing uses the same environment
- re-testing is done by developers, regression testing is done by independent testers

Faute, erreur et défaillance

Faute, erreur et défaillance

- **Faute : cause adjugée ou supposée d'une erreur**
- **Erreur : état susceptible d'entraîner une défaillance**
 - latente/détectée
 - propagation d'erreur, produit d'autres erreurs
- **Défaillance : manifestation d'une erreur qui par propagation traverse la frontière du système avec son environnement.**
- **...défaillance -> faute -> erreur -> défaillance -> faute ->**

Famille de fautes

- **calcul :**
 - Exemple: existence d'une instruction $x := x + 2$ au lieu de $x := y + 2$;
- **logique : mauvaise expression d'un prédicat.**
 - Exemple: le programmeur a écrit « if (a>b) then » au lieu de « if (a<b) then » ;
- **entrée/sortie :**
 - nous incluons dans cette classe tous les défauts exprimant une mauvaise description, une mauvaise conversion ou un formatage inadéquat des entrées/sorties (communication avec l'extérieur) ;
- **traitement des données :**
 - mauvais accès ou manipulation des données
 - ✓ mauvaise utilisation d'un pointeur,
 - ✓ variable non définie,
 - ✓ ...
- **interface :**
 - mauvaise communication entre les composants internes du logiciel
 - ✓ appelle la procédure P1 au lieu d'appeler la procédure P2,
 - ✓ le passage des paramètres n'est pas correct,
 - ✓ ...
- **définition des données :**
 - Exemple: une donnée a été déclarée comme un entier alors qu'elle aurait du être déclarée comme un réel
-

Faute, erreur et défaillance

- **Difficulté de faire le lien entre une faute et une erreur.**

- **Exemple:**

- Soit X suffisamment grand

...

a=X*100; // au lieu de X/100

a=100*a; // overflow

...

Les défauts

➤ **Loi de Paréto (loi sociologique):**

- 80 % de la valeur vient de 20 % des articles et 20 % de la valeur de 80 % des articles.

➤ **Interprétation**

- 20% des fautes provoquent 80% des défaillances, car 80% du temps d'exécution est consacrée à 20% du code.
- => il est possible d'optimiser le coût de la réduction du taux de défaillance.

Remarque

- pas de faute
 - => pas d'erreur
 - => pas de défaillance

Quelque Moyens

➤ La prévention des fautes

- éviter les fautes : **Assurance Qualité**
- détecter les fautes, et corriger leurs effets : **Contrôle Qualité** du processus

➤ L'élimination des fautes

- détecter et corriger les défauts : **Vérification et Validation**
Test indirectement, à partir des pannes

➤ L'inhibition des fautes :

- En cours d'exécution,
détecter et récupérer les défaillances **Programmation tolérante**

Vérification et Validation

Vérification & Validation

➤ **Vérification :**

- Le produit (logiciel) correspond à sa spécification?
- Le construisons nous bien ?

➤ **Validation :**

- Le produit (logiciel) correspond aux attentes?
- Construisons nous le bon produit ?

Vérification

➤ **Vérification**

- confirmation par examen et apport de preuves que les propriétés et caractéristiques attendues sont satisfaites

➤ **Répond à la question :**

- est-ce un système bien fait, conformément aux règles de l'art ?

➤ **Point de vue interne, s'occuper de :**

- la structure du système, comment il est fait, en référence à des normes, des propriétés à satisfaire (vérifier le produit)
- des moyens mis en œuvre, le processus de production ; en référence à des règles sur la méthode de travail, comment on doit procéder (vérifier le processus),

Deux modes de vérification

➤ Les vérifications statiques :

- sans exécution du code;

➤ Les vérifications dynamiques :

- avec exécutions du code:
 - ✓ Exécution symbolique
 - ✓ Test unitaire, d'intégration ou fonctionnel.

Validation

➤ Validation

- confirmation par examen et apport de preuves que *les exigences requises pour l'usage prévu* sont satisfaites

➤ Répond à la question :

- est-ce le bon système, répondant aux besoins effectifs ?

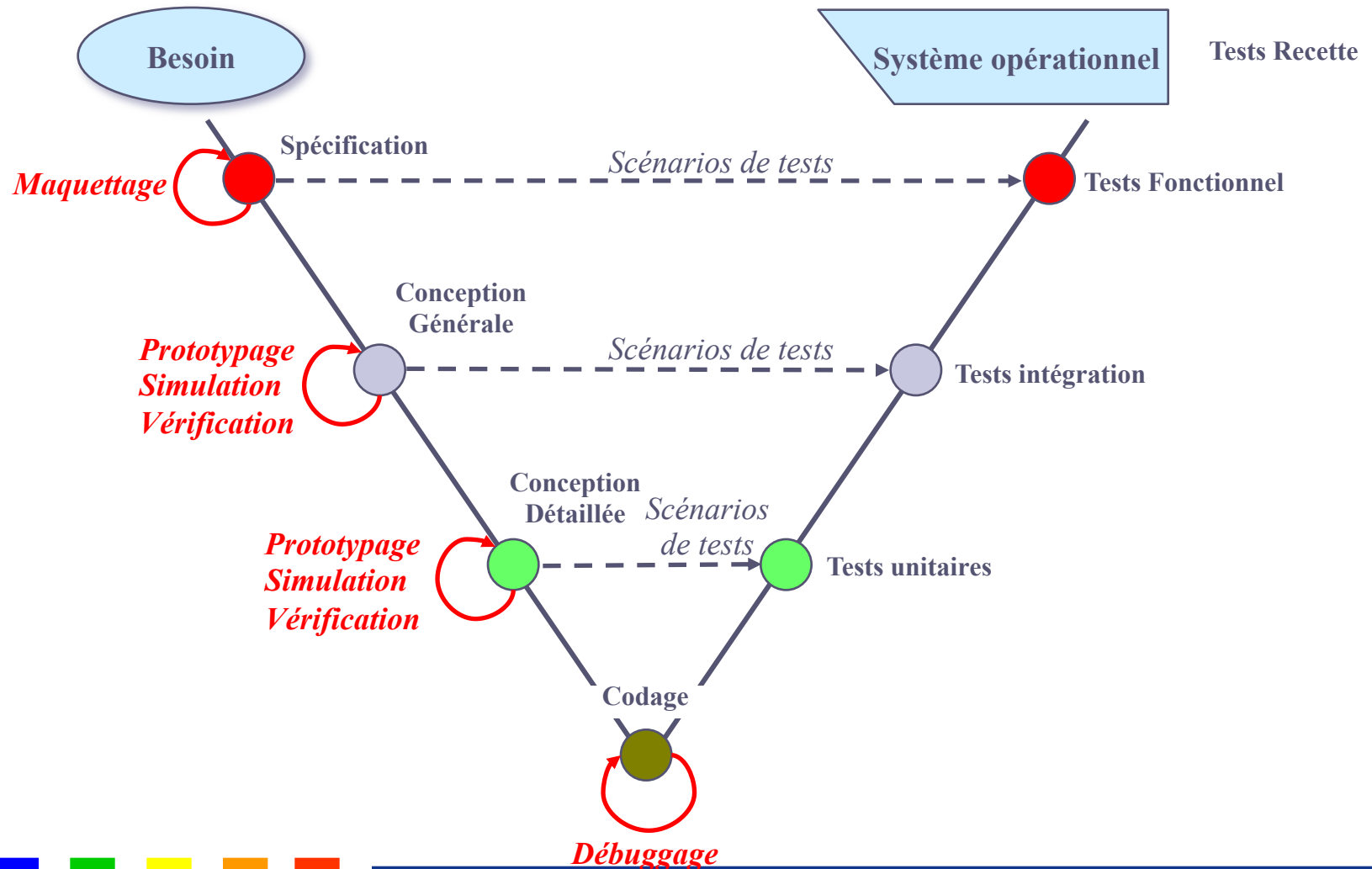
➤ Point de vue externe, s'occuper de :

- l'aptitude du système à accomplir ses missions, atteindre les objectifs assignés
- ce que l'on peut faire avec (le quoi)

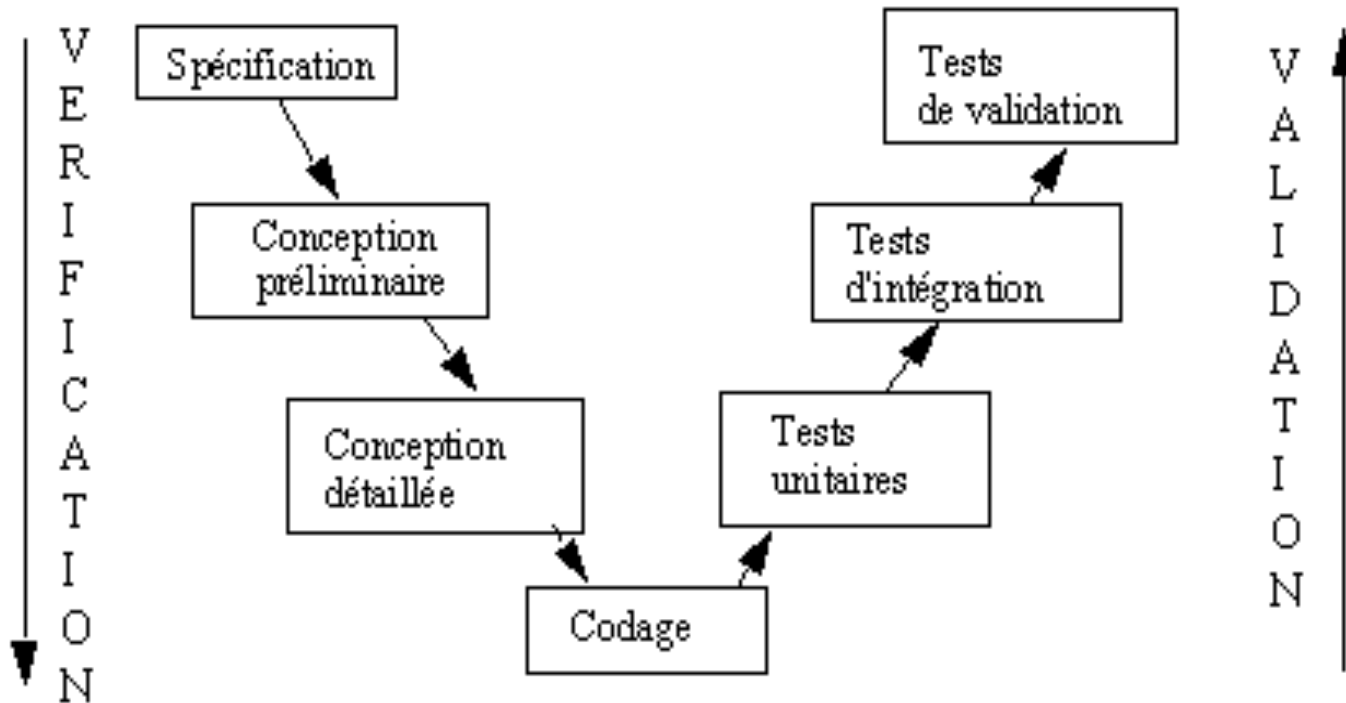
➤ Se fait en référence à ce que l'on attend :

=> **il est impossible de valider quelque chose si ce que l'on en attend n'est pas précisément défini.**

Vérification & Validation



Vérification & Validation (7)

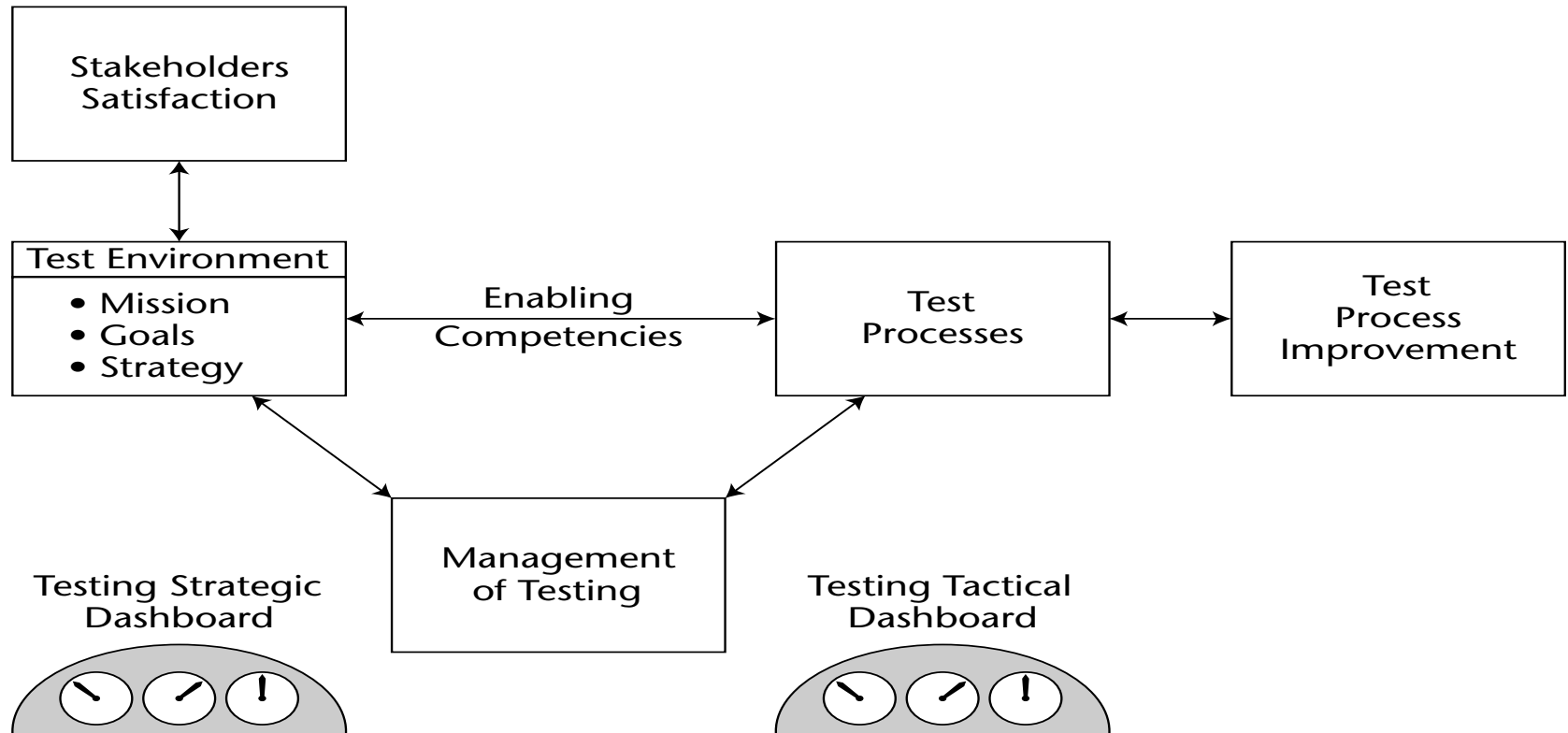


Evaluation des capacités de tests et des compétences de votre entreprise

Feuille de route en trois étapes:

1. Définir ou adopter un modèle de test de logiciels.
2. Déterminer le niveau actuel de l'organisation de votre entreprise par rapport aux fonctions de test de logiciel, les compétences et la satisfaction des utilisateurs.
1. Élaborer et mettre en œuvre un plan de mise à niveau de vos capacités et compétences.

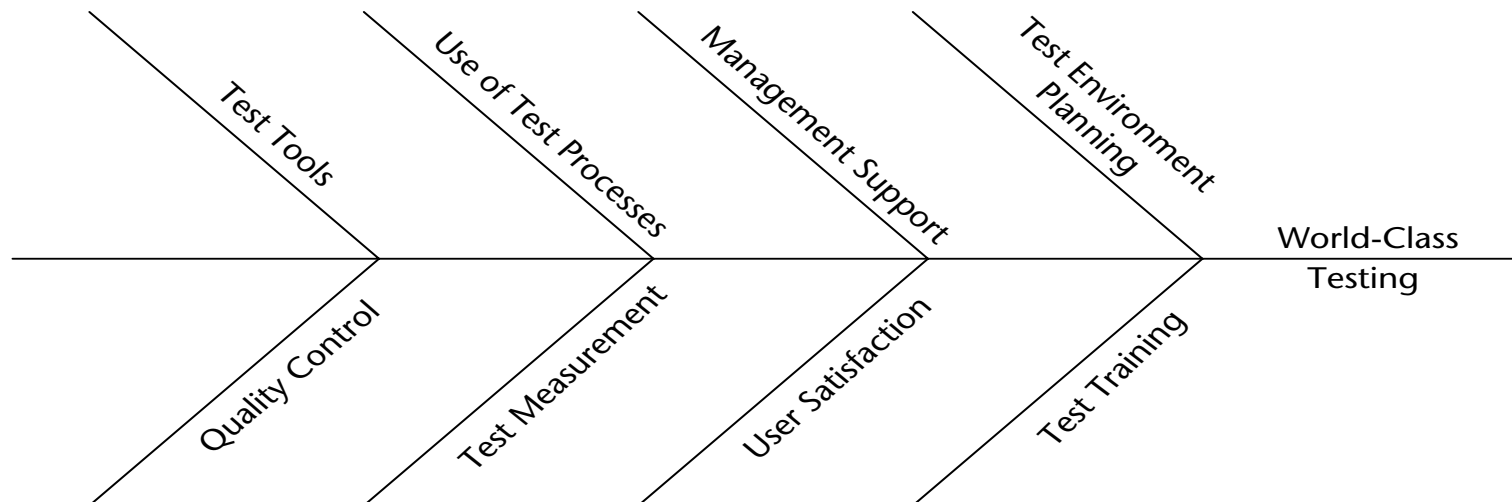
Model of a world-class software testing organization



Overview of the testing environment

DRIVERS OF WORLD-CLASS TESTING

DESIRED RESULTS



Example of a software testing organization

