esprit
Ecole Supérieure Privée
d'Ingénierie et de Technologies

www.esprit.ens.tn

Test & validation

Conformément à la certification ISTQB
(International Software Testing Qualification Board)

Se former autrement

pour une nouvelle génération d'ingénieurs

esprit
Ecole Supérieure Privée
d'Ingénierie et de Technologies

Tests et validation

- Introduction
- Chapitre 1 : Fondamentaux de Test
- Chapitre 2 : Test pendant le cycle de vie
- Chapitre 3 : Techniques statiques
- Chapitre 4 : Technique de conception des tests
- Chapitre 5 : Gestion des tests
- Chapitre 6 : Outils de support aux tests
- Laboratoire de tests

Plan

- Introduction
- Modèles de développement logiciel
- Niveaux de tests
- Types de tests
- Conclusion
- Exercices

Introduction

- Tous les projets de développement ont des activités communes :

- Les tests n'existent pas de façon isolée, ils sont liés aux activités de développement logiciel.
- Les différents modèles de cycle de développement nécessitent des approches de tests adaptées.
- Différents modèles par exemple : Modèle en cascade, Modèle en V, Modèle itératif ….

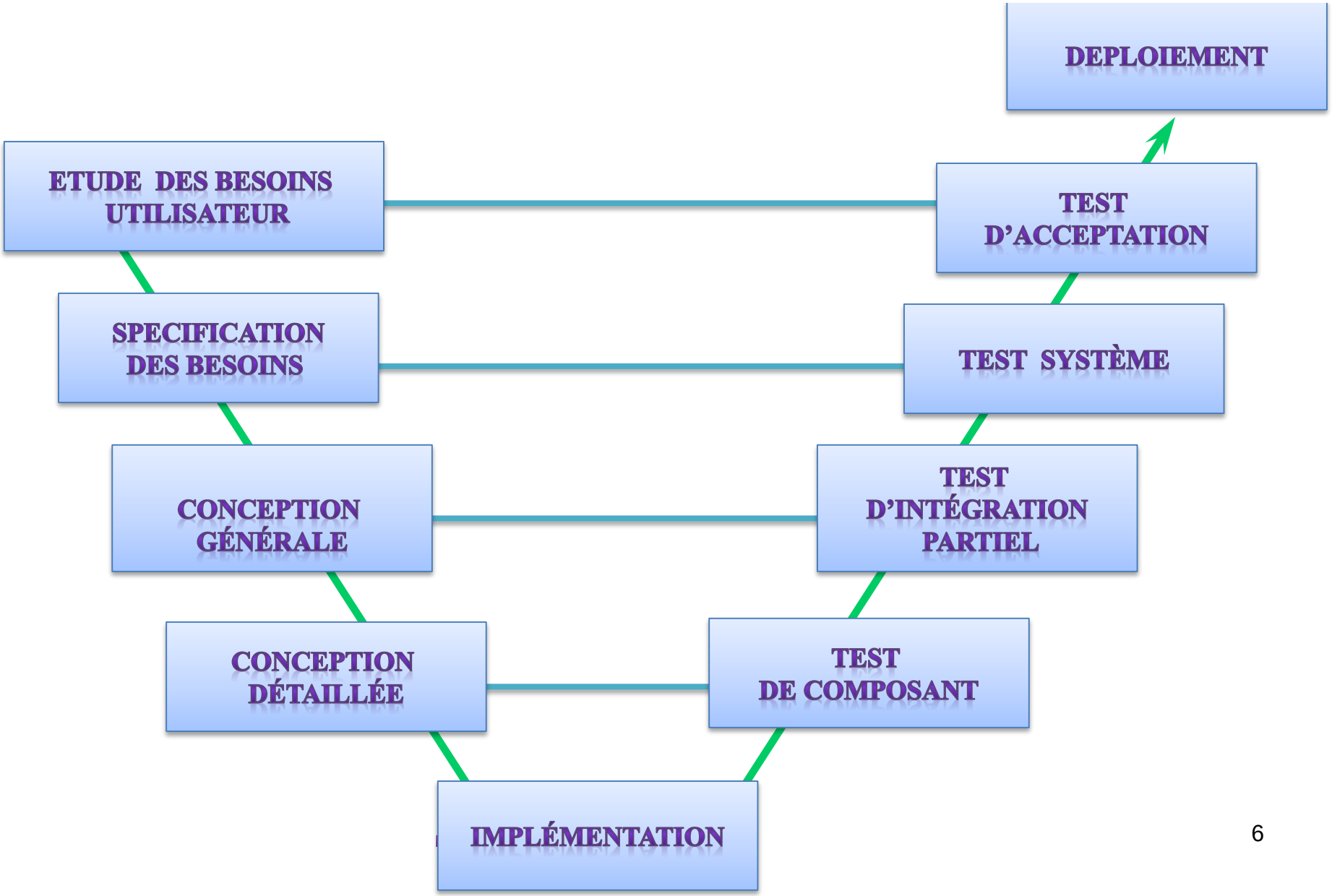
Bonnes Pratiques

- Quel que soit le modèle de cycle de vie de développement logiciel, il y a plusieurs caractéristiques de bonnes pratiques:

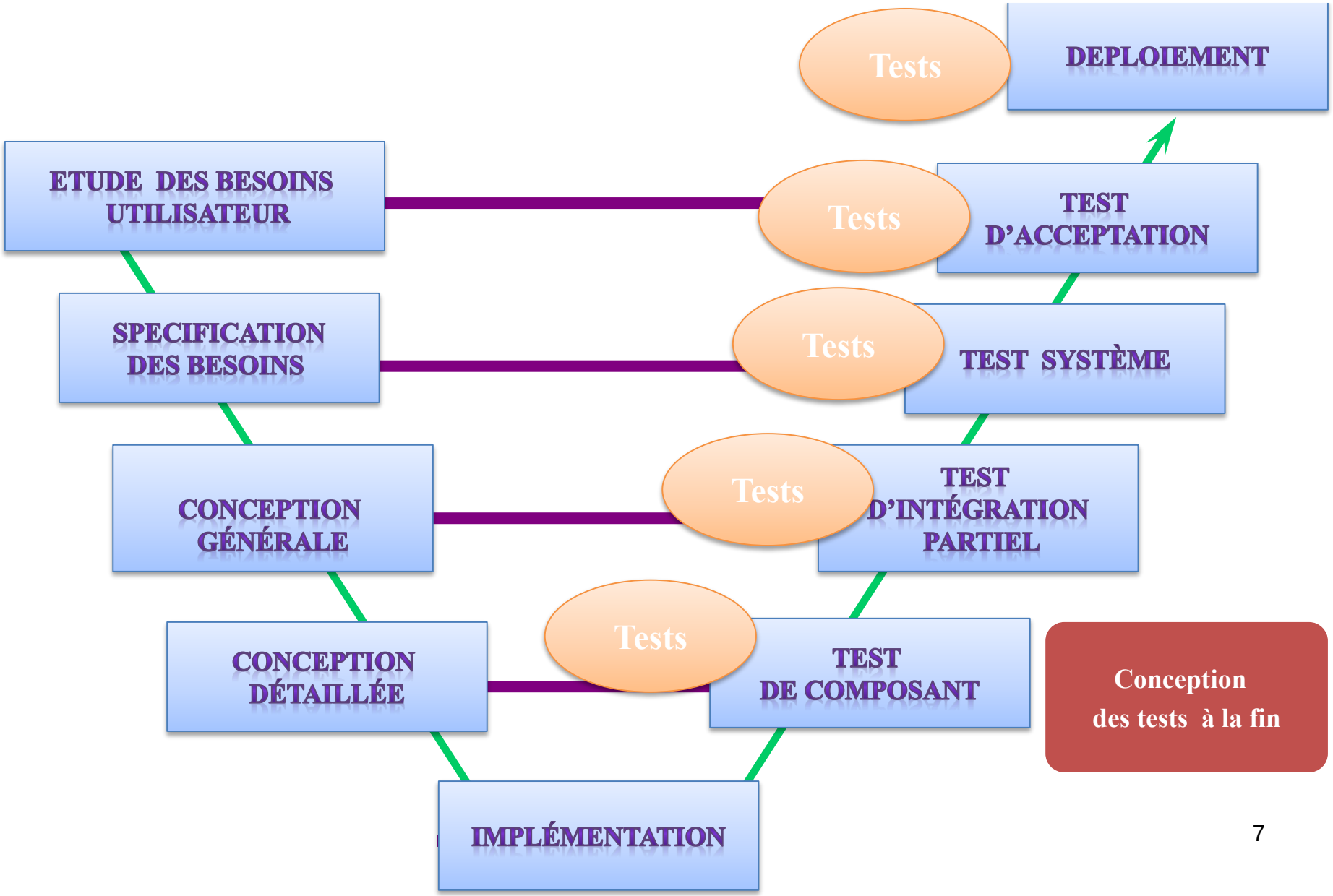
  - Pour chaque activité de développement, il y a une activité de test correspondante.
  - Chaque niveau de test a des objectifs de test spécifiques à ce niveau.
  - L'analyse et la conception des tests pour un niveau de test donné commencent au cours de l'activité de développement correspondante.
  - Les testeurs participent aux discussions pour définir et affiner les exigences et la conception, et sont impliqués dans la revue des produits d'activités (p. ex. les exigences, la conception, les User Stories, etc.) dès que des versions préliminaires sont disponibles.

esprit
Ecole Supérieure Privée
d'Ingénierie et de Technologies

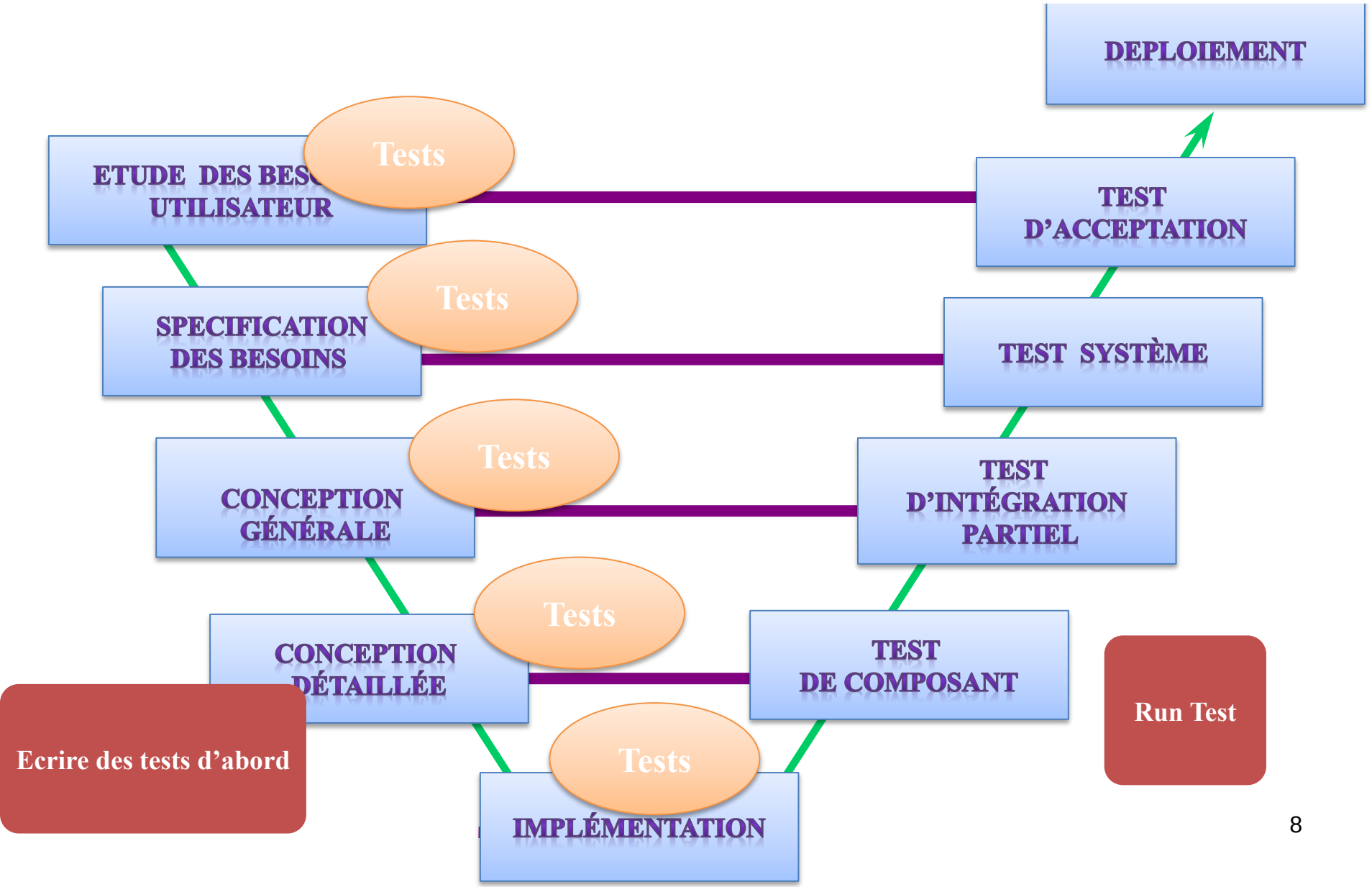
Modèle de développement de logiciel : Modèle en V



Modèle de développement de logiciel : Les tests



Modèle de développement de logiciel : Les tests



Modèle de développement de logiciel : Modèle itératif

Principe:

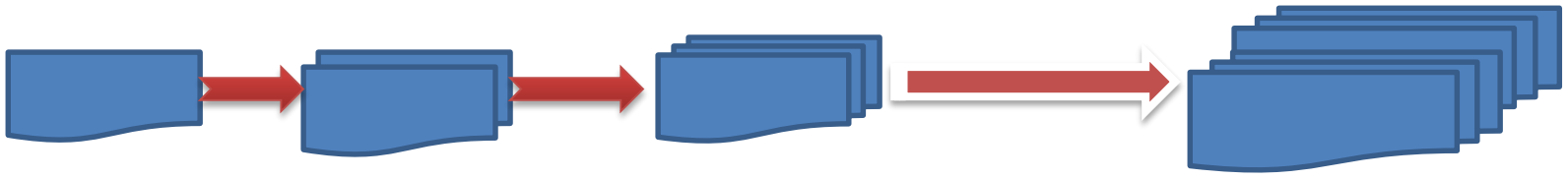
Développement logiciel : une série de petits fragments à développer.

Un fragment : exigences, conception, construction et tests.

Un incrément : est le système logiciel résultant à chaque itération qui ajoute un fragment développé. Il forme un système partiel en croissance.

Plusieurs niveaux de tests

Les tests de régression sont de plus en plus importants sur toutes les itérations.



Développement logiciel en contexte

- Les modèles de cycle de vie du développement logiciel doivent être **sélectionnés** et **adaptés** au contexte du projet et aux caractéristiques du produit en fonction :
  - de l'objectif du projet,
  - du type de produit développé,
  - des priorités métier (p. ex., délai de mise sur le marché),
  - des risques produit et projet identifiés.

- Selon le contexte du projet, il peut être nécessaire de combiner ou de réorganiser les niveaux de test et/ou les activités de test.

- Les modèles de cycle de vie du développement logiciel eux-mêmes peuvent être combinés (exp: un modèle en V pour le back-end et un modèle Agile pour le front-end (IHM).

esprit
Ecole Supérieure Privée
d'Ingénierie et de Technologies

Niveaux de Test

Test de composant

Test unitaire: revue de code
Tests unitaires structurels
Tests unitaires fonctionnels

Tests d'intégration

Tests d'assemblage
Tests Interface Homme Machine (IHM)
Interaction entre les composants

Tests Système

Tests de comportements

Tests d'acceptation

Tests de régression
Tests d'acceptation utilisateurs
Tests de performances / stress
Tests d'acceptation opérationnelle (pré-exploitation)

Niveaux de Test

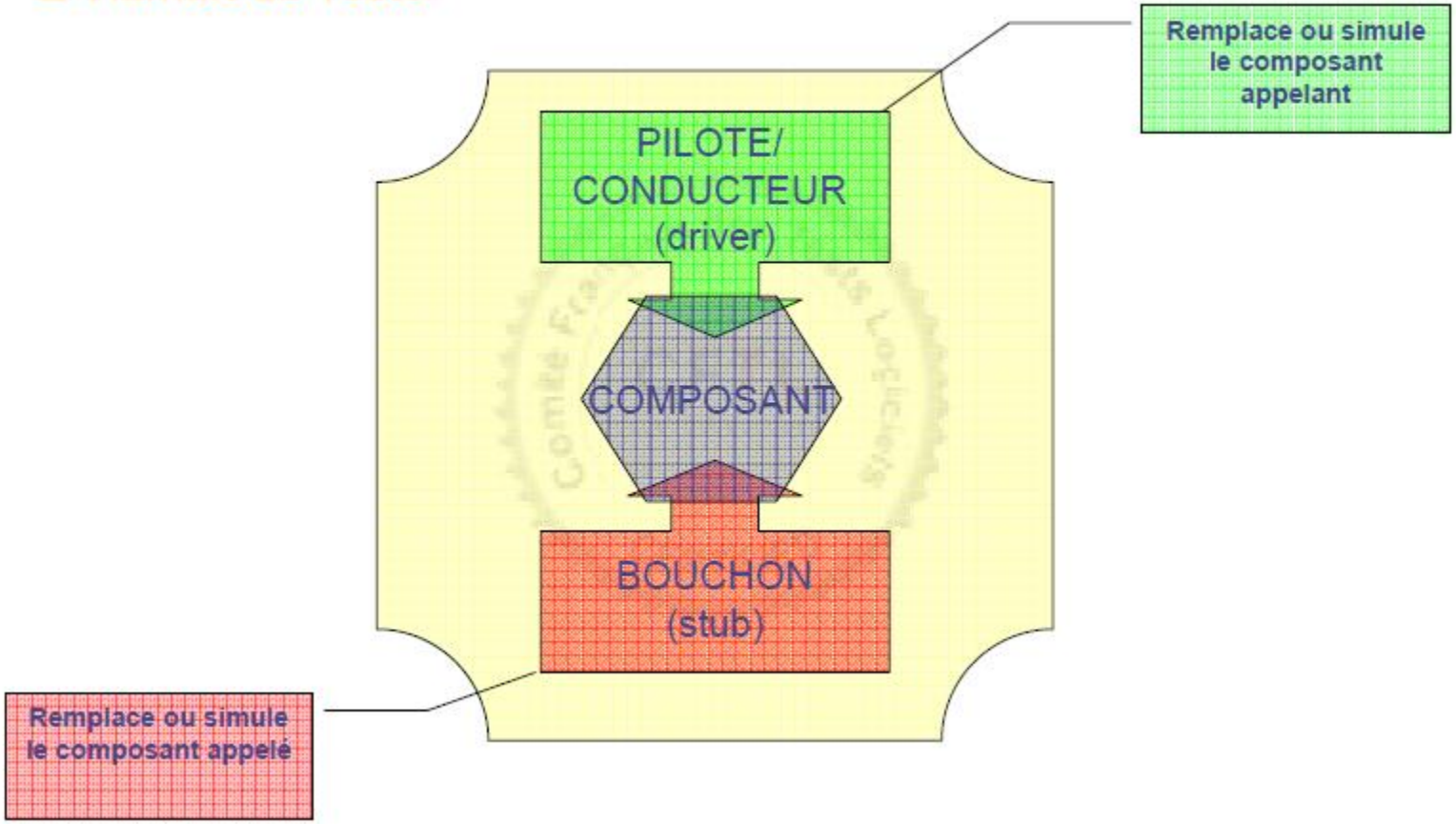
Tests des composants :

- Recherche des défauts et vérification du bon fonctionnement des modules, objets, classes qui sont testables séparément
- Peuvent inclure des tests fonctionnels, non fonctionnels et boite blanche
- Les tests de composants se font généralement avec l'accès au code du logiciel testé et sur un environnement de développement comme un framework de tests unitaire
- Vérification de la couverture du code
- Utilisation des Bouchons, Pilote et des simulateurs …

Niveaux de Test

Tests des composants :

- Harnais de Tests:



Niveaux de Test

Tests des composants :

- Comment tester ?
  - **Avec des pilotes (drivers) et des bouchons (stubs) :** Nécessite une collaboration des développeurs …
- Quoi tester ?
  - **Aspects fonctionnels :** Fonctionnement du composant selon ses spécifications
  - **Aspects non fonctionnels :** Fuites mémoire, robustesse, temps de réponse, …
  - **Aspects structurels :** Couverture des branches, des décisions, etc.
- Particularités
  - **Tests effectués avec le développeur**
  - **Multiples itérations possibles**

Niveaux de Test

Tests des composants :

- Boite noire (basées sur les spécifications)

Plusieurs catégories

- Partitions d'équivalence
- Analyse des valeurs limites
- Test de transition d'états
- Tests par tables de décisions
- Tests de cas d'utilisation

- Boite blanche(basées sur les structures)
  - Test de déclaration
  - Test de Decision
  - Test de flux de données
  - Test de branche de conditionnement
  - Branch condition combination testing
  - Modified condition decision testing.
  - LCSAJ testing

Niveaux de Test

Tests d'intégration :

- Tests d'intégration des composants testant les interactions entre les composants logiciels
- Tests d'intégration système testant l'intégration entre les différents systèmes ou entre logiciel et matériel
- Peuvent inclure des tests fonctionnels, non fonctionnels et boite blanche
- Plus le nombre de composants est élevé, plus il devient difficile d'isoler les défauts

Niveaux de Test

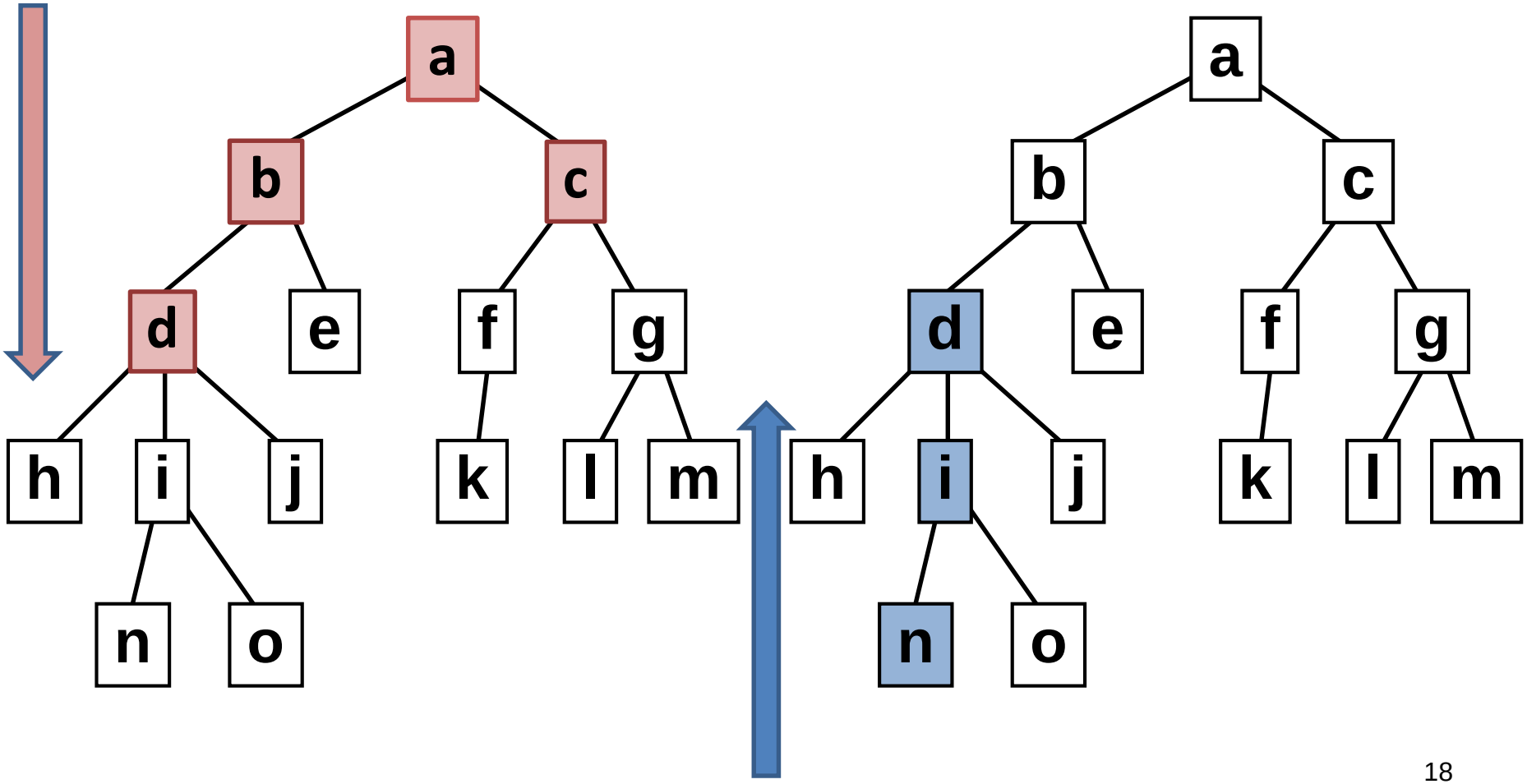
Tests d'intégration :

- Des stratégies d'intégration peuvent être basées sur l'architecture des systèmes :
  - Big Bang,
  - Top-down,
  - Et Bottum-up.

} Incrémentales

Niveaux de Test

Tests d'intégration : Top-down vs Bottum-up.



Niveaux de Test

Tests Systèmes :

- Les tests systèmes traitent le comportement d'un système/produit fini
- Inclusion des tests basés sur des risques
- L'environnement de test devrait correspondre le maximum possible à la cible finale
- Les tests système devraient examiner à la fois les exigences fonctionnelles et non-fonctionnelles du système
- Une équipe indépendante réalise généralement les tests systèmes

Niveaux de Test

Tests d'acceptation:

- Les objectifs des tests d'acceptation sont de prendre confiance dans le système ou dans des caractéristiques non-fonctionnelles du système.
- Evaluer si le système est prêt à être déployé .
- Souvent de la responsabilité des clients ou utilisateurs d'un système
- Les formes des tests d'acceptation incluent:
  - Tests d'acceptation utilisateurs

    Vérifier l'aptitude et l'utilisabilité du système par des utilisateurs
  - Tests (d'acceptation) opérationnelle

    Tests des backups et restaurations, reprise après sinistre, Gestion des utilisateurs, Tâches de maintenance, Chargement de données et tâches de migration
  - Tests d'acceptation contractuelle et réglementaire
  - Tests alpha

    Tests opérationnel réel ou simulé par des utilisateurs/clients potentiels ou par une équipe de test indépendante sur le site de développement, mais en dehors de l'organisation de développement.
  - Tests beta

    Tests opérationnels par des utilisateurs/clients potentiels et/ou réels sur un site externe non associé aux développeurs, pour déterminer si un composant ou système satisfait ou non les besoins des utilisateurs/clients et s'adaptent aux processus d'entreprise.

Types de Test

Tests fonctionnels : Que fait le système?

- Tests basés sur une analyse des spécifications d'une fonctionnalité d'un composant ou système.

Tests non fonctionnels : Comment le système fonctionne ?

- Les tests non-fonctionnels incluent, mais pas uniquement, les tests de performances, tests de charge, tests de stress, tests d'utilisabilité, tests de maintenabilité, tests de fiabilité et les tests de portabilité.

Types de Test

Tests boîte-blanche: Quel est le contenu du système?

- Tests basés sur une analyse de la structure interne du composant ou système.
- Sert à mesurer l'ampleur des tests via l'évaluation de la couverture d'un type de structure.
- Tests structurels (v. 2011) .

→ Détection des erreurs de programmation

Types de Test

Tests liés au changement

- Tests de confirmation

  Tests qui exécutent des cas de test qui ont été en échec la dernière fois qu'ils furent exécutés, de façon à vérifier le succès des actions de correction.

- Tests de régression

  Tests d'un programme préalablement testé, après une modification, pour s'assurer que des défauts n'ont pas été introduits ou découverts dans des parties non modifiées du logiciel.

esprit
Ecole Supérieure Privée
d'Ingénierie et de Technologies

Tests de maintenance

- Effectués sur un système opérationnel existant et sont déclenchés par des modifications, migrations ou déclassement (suppression) de logiciels ou de systèmes.

- Ils incluent:
  - Les tests des modifications
  - Les tests de régression

- Une analyse d'impact permet de:
  - évaluer les changements afin d'identifier les conséquences prévues ainsi que des effets secondaires attendus et possibles d'un changement,
  - identifier les zones du système qui seront affectées par le changement.
  - identifier l'impact d'un changement sur les tests existants.

esprit
Ecole Supérieure Privée
d'Ingénierie et de Technologies

Conclusion

- Les tests sont assurés tout au long du projet depuis la mise en place des exigences et la définition du cahier de charge jusqu'à sa mise en vente et sa commercialisation.
- Différents intervenants dans les tests : les développeurs, les clients, les utilisateurs …
- Une exécution des différents tests dans les différents étapes de projet assurent :
  - Une meilleure qualité du logiciel (conformité, robustesse, performance, sécurité …)
  - Coût minimal

esprit
Ecole Supérieure Privée
d'Ingénierie et de Technologies

Exercice

1. Les tests système doivent examiner :

A. Les exigences non fonctionnelles
B. Les exigences fonctionnelles
C. Les exigences non fonctionnelles et fonctionnelles
D. Aucune réponse

2. Laquelle des ces affirmations définissent le test de régression?

1. Ces tests sont effectués quand le logiciel ou son environnement est modifié.
2. Ils sont équivalents aux tests de confirmation (re-test)
3. Ils sont un moyen de s'assurer que les défauts n'ont pas été introduits ou découverts dans des parties de code non modifiés du logiciel.
4. Ils ne sont efficaces que s'ils sont automatisés

A. 1 et 2
B. 1 et 3
C. 2 et 3
D. 2 et 4

esprit
Ecole Supérieure Privée
d'Ingénierie et de Technologies

Exercice

3. Quelle affirmation sur le test fonctionnel est exacte?

A. Le test basé sur la structure est le plus important que le test fonctionnel car il touche le code.
B. Le test fonctionnel est utile tout au long le cycle de vie et peut être appliqué par des analystes métiers, des testeurs, des développeurs et des utilisateurs.
C. Le test fonctionnel est le plus puissant que le test statique puisque le système est réellement sollicité
D. La revue de code est une forme de test fonctionnel

4. Laquelle de ces informations est la plus souvent correcte ?

A. Les inspections du code source sont souvent utilisées dans le test composant.
B. Le test de composant cherche les défauts dans les programmes qui sont testables séparément.
C. Le test de composant est une étape importante dans les tests d'acceptation utilisateurs.
D. Le test de composant a pour objectif d'exposer les problèmes des interactions entre les composants software et hardware

Exercice

5. Les tests de confirmation sont :

A. Les tests des modifications apportés à un programme.
B. Les tests basés sur la structure interne du système.
C. Les tests qui exécutent les cas de test qui ont été en échec la dernière fois qu'ils furent exécutés.
D. Les tests fonctionnels et non fonctionnels.

6. Les tests réalisés pour une intégration de haut en bas (Top-Down)

A. Est préférable à une intégration de bas en haut (Bottom-Up)
B. Est plus efficace qu'une intégration big-bang
C. Nécessite des pilotes
D. Nécessite des bouchons