

Préparation Revue 2

Introduction à l'ingénierie des tests



Introduction à l'ingénierie des tests

- **Introduction**
- **Types de tests**
- **Les 7 principes des tests**
- **Les tests unitaires**
- **Exemple de mise en oeuvre**
- **Conclusion**

Introduction

- **Les tests informatiques vérifient le bon fonctionnement d'un système.**
- **La manière dont les tests sont organisés peut varier en fonction des équipes et des entreprises.**
 - Certains tests peuvent être plus formels ou automatisés, impliquant l'exécution de programmes dédiés.
- **Les tests sont essentiels pour évaluer la qualité d'un produit.**
- **Leur objectif principal est de repérer les écarts entre le comportement attendu et celui effectivement observé.**

Introduction

- **L'ingénierie des tests implique donc des activités et des spécialités différentes :**
 - Le testeur en charge de réaliser les tests
 - La personne en charge de définir les procédures à réaliser (les scénarii de test).
 - Il ne s'agit pas nécessairement du testeur.
 - Le développeur en charge de réaliser le ou les programmes d'automatisation des tests.

Types de tests

- **Les tests unitaires :**
 - testent une partie (une unité) d'un système afin de s'assurer qu'il s'exécute sans erreur.

Types de tests

- **Les tests unitaires**
- **Les tests d'acceptation :**
 - Testent le système afin de s'assurer qu'il est conforme aux attentes de l'utilisateur.

Types de tests

- **Les tests unitaires**
- **Les tests d'acceptation**
- **Les tests d'intégration :**
 - Testent le système sur une plate-forme proche de la plate-forme cible afin de s'assurer que les différents composants s'intègrent correctement les uns avec les autres.

Types de tests

- **Les tests unitaires**
- **Les tests d'acceptation**
- **Les tests d'intégration**
- **Les tests de performance :**
 - Testent les performances du système afin de s'assurer qu'il répond à certains critères de réactivité.

Types de tests

- **Les tests unitaires**
- **Les tests d'acceptation**
- **Les tests d'intégration**
- **Les tests de performance**
- **Les tests de sécurité :**
 - Testent que l'application ne contient pas de failles de sécurité connues (injection de code, attaque XSS, ...)

Types de tests

- Les tests unitaires
- Les tests d'acceptation
- Les tests d'intégration
- Les tests de performance
- Les tests de sécurité
- Les tests de robustesse (ou stress tests) :
 - Testent le comportement de l'application aux limites des ressources disponibles (mémoire, CPU, ...).

Types de tests

- **Les tests de non-régression**
 - Les tests de non-régression ne sont pas un type particulier de tests.
 - Suite à une modification apportée à un système, un test de non-régression permet de s'assurer que le système continue de fonctionner comme précédemment pour les parties qui ne devraient pas être impactées par cette modification.

Les 7 principes des tests

- **7 principes de tests ont été formalisés par le comité ISTQB.**

Les 7 principes des tests

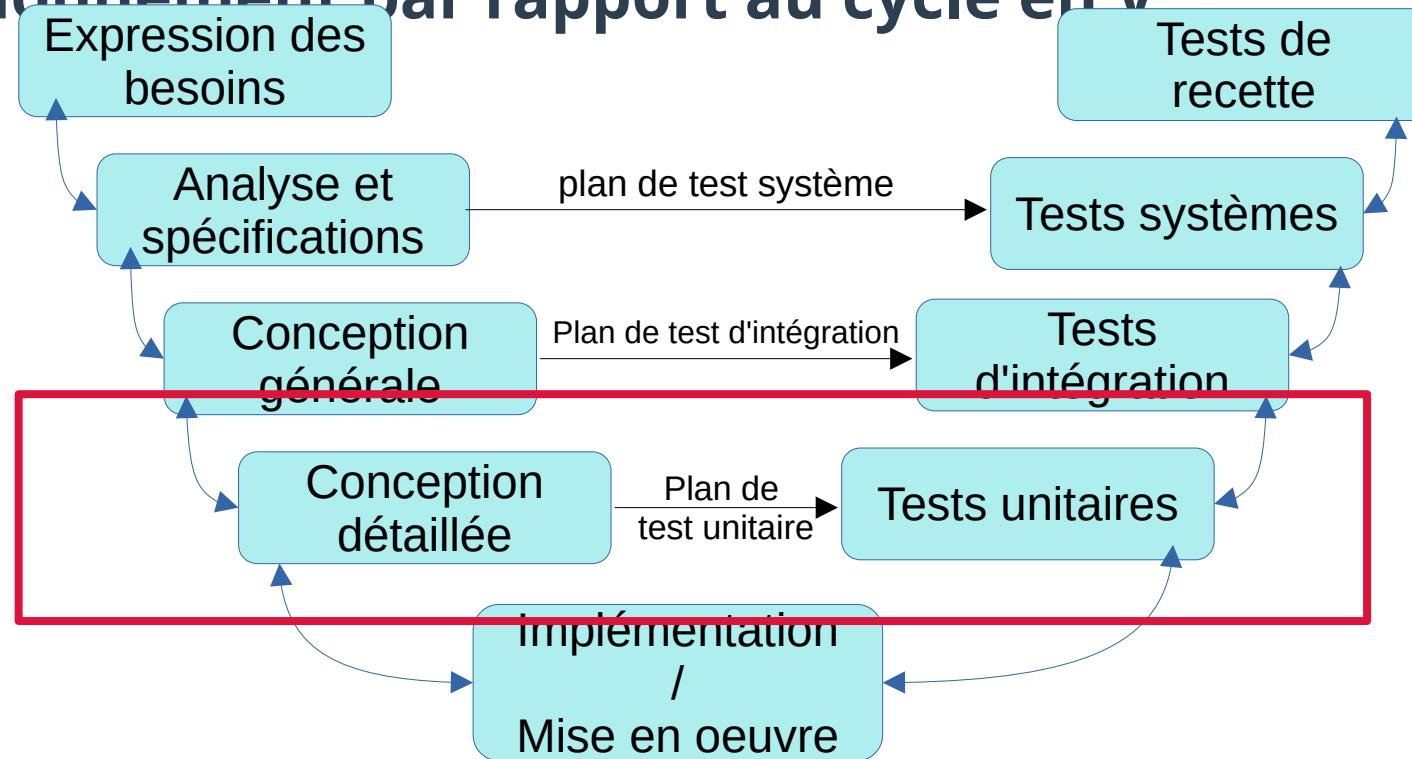
- **7 principes de tests ont été formalisés par le comité ISTQB.**
 - Les tests montrent la présence d'un défaut
 - Les tests exhaustifs sont impossibles ([Théorème de Rice](#))
 - Tester au plus tôt
 - Regroupement des défauts

Les 7 principes des tests

- **7 principes de tests ont été formalisés par le comité ISTQB.**
 - Les tests montrent la présence d'un défaut
 - Les tests exhaustifs sont impossibles ([Théorème de Rice](#))
 - Tester au plus tôt
 - Regroupement des défauts
 - Le paradoxe du pesticide
 - La stratégie de test dépend du contexte
 - L'illusion d'absence d'erreur

Les tests unitaires

- Positionnement par rapport au cycle en V



Les tests unitaires

- **Un test unitaire est une procédure permettant de vérifier le bon fonctionnement d'une partie précise d'un programme ou d'un système.**
 - En programmation procédurale, une unité est une **fonction**.
 - En programmation orientée objet, une unité est une **classe**.
 - En architecture système et réseau, une unité peut être un **service** (DHCP, DNS, etc)

Les tests unitaires

- **Objectifs :**
 - Test séparé de chacun des composants du logiciel en vue de leur intégration
- **Activités :**
 - Réalisation des tests prévus pour chaque module
 - Les tests sont à faire par un membre de l'équipe n'ayant pas participé à la fabrication du module
- **Résultats :**
 - Résultats des tests avec les jeux d'essais par module selon le plan de test.

Les tests unitaires

- **2 types de tests**
 - Lors d'un test **fonctionnel**, on se réfère uniquement à l'observation de la sortie pour certaines valeurs d'entrée (**test boite noire**).
 - Il n'y a aucune tentative d'analyser le code, qui produit la sortie.
 - Lors d'un test **structurel**, on exploite la connaissance de la structure et la mise en œuvre du logiciel (**test boite blanche**).
 - Passage en mode debug, mode pas à pas, ...

Plan de test

- **Préciser le(s) cas d'utilisation(s) concerné(s)**
- **L'objectif du test**
- **L'unité testée**
 - type de test
- **Les conditions préalables et l'environnement de test**
- **Les données du test**
- **Les scénarii**
 - Procédure détaillée
 - Critères de réussite

Exemple de mise en oeuvre

- **Système Agence de voyage**

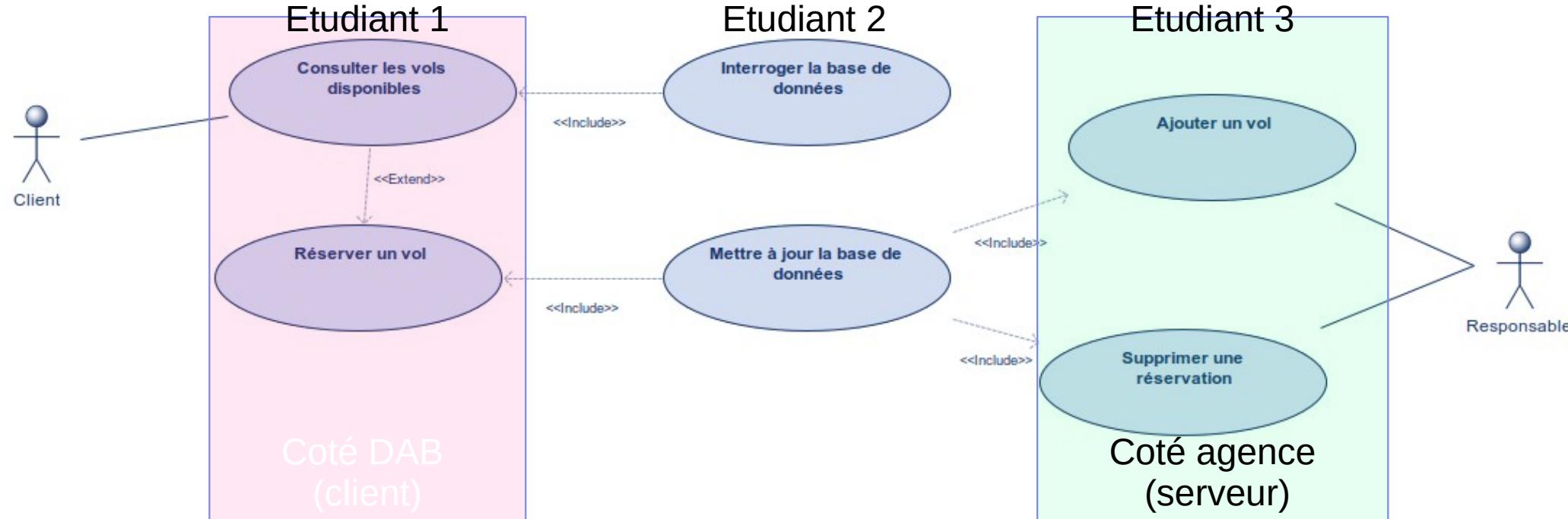
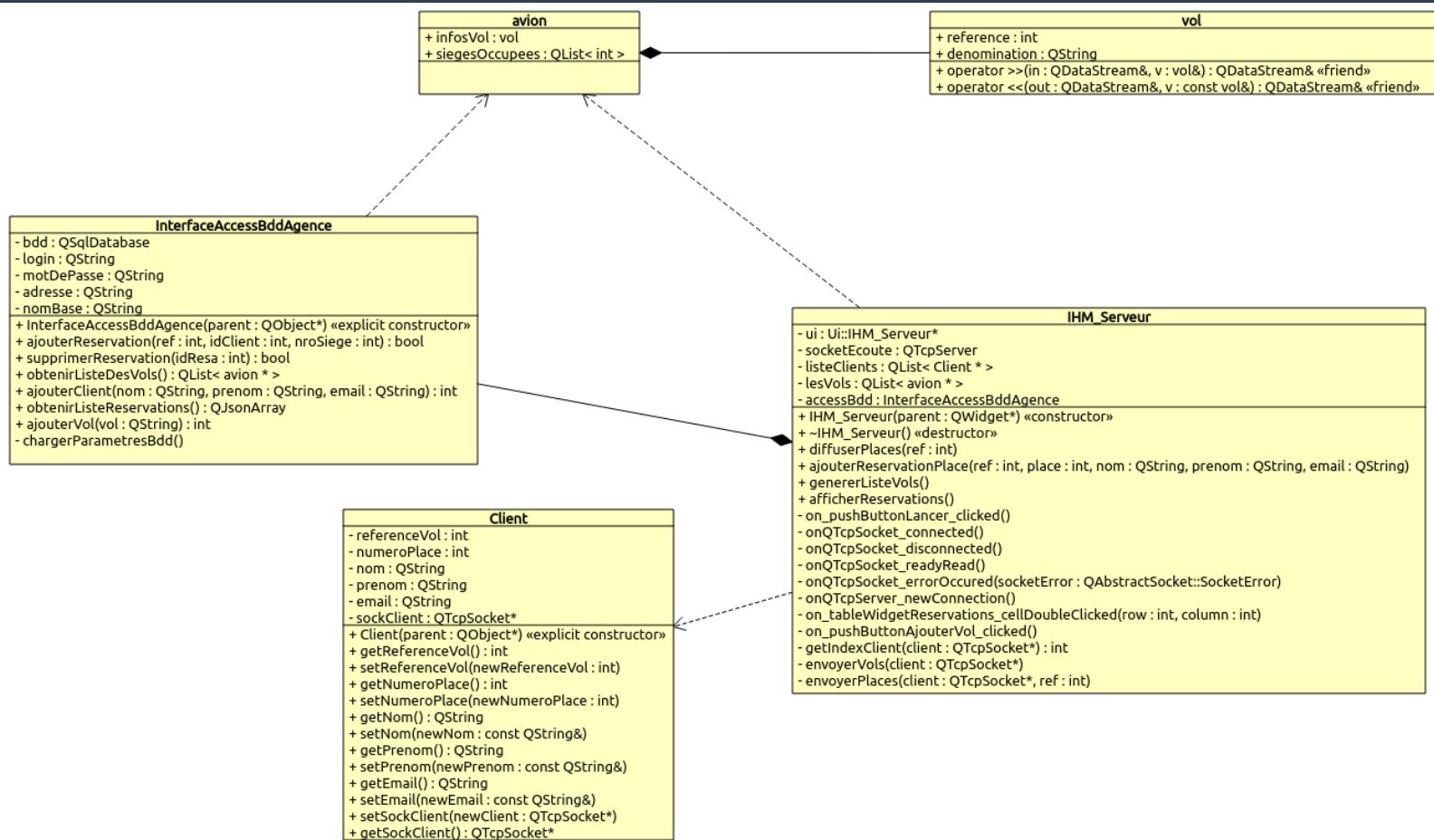
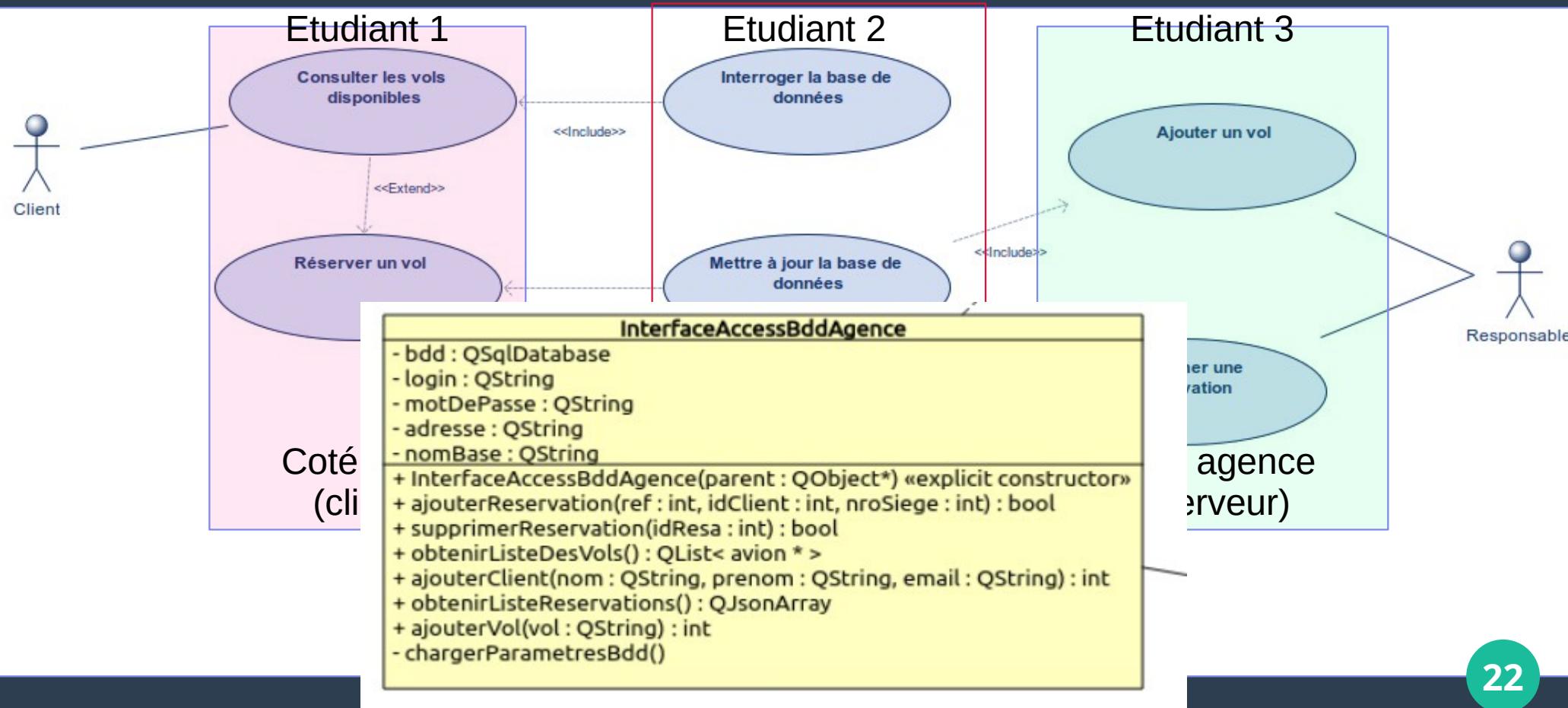


Diagramme Classe coté serveur



Unité à tester



Tables et données de la base

agence

clients		
id	nom	email
1	robert	rbidochon@ici.fr
2	galopin	rgalopin@labas.com
3	toto	hdsfsh@dfs.gt

idResa	referenceVol	idClient	numeroSiege
9	3456	1	1
11	3456	1	8
13	3456	1	15

reservations			
idResa	referenceVol	idClient	numeroSiege
9	3456	1	1
11	3456	1	8
13	3456	1	15

vols	
reference	denomination
1234	Paris - Londre
2345	Paris - New York
3456	Paris - Sydney
3457	Paris - Berlin
3458	Le Mans - Angers
3459	Le Mans - Nantes

reference	denomination
1234	Paris - Londre
2345	Paris - New York
3456	Paris - Sydney
3457	Paris - Berlin
3458	Le Mans - Angers
3459	Le Mans - Nantes

Plan de test

Compte rendu de test

Compte rendu de test

<u>Référence du module testé :</u>		F2.1
N°Opérations	Résultats attendus	Résultats obtenus
		Conformes
		non Conformes

Conclusion

- **Les Tests : une discipline à part entière**
- **Très importants dans le cycle de développement**
- **Les tests unitaires :**
 - Peuvent être en partie automatisé (QTest, PHPUnit, CPPUNIT, etc ..)
 - Malheureusement ne couvrent pas tous les cas,
 - Restent néanmoins considérés comme un gage de qualité

Le test de programmes peut être une façon très efficace de montrer la présence de bugs mais est désespérément inadéquat pour prouver leur absence. The humble programmer - 1972

Edsger Dijkstra
(Mathématicien/informaticien)