



Chapitre 6

Tests et Validation

UP GL-BD



Plan Chapitre 2

1. Problématique
2. Objectifs
3. Définition
4. Principes
5. Types de défauts logiciels
6. Méthodes de tests
7. Types de tests
8. Le testing: un métier
9. Exemple de fiche de tests



Problématique

- Echec du vol inaugural de la fusée Ariane 5 en 1996 :
 - Dépassement de capacité provoquant le crash informatique de l'appareil.
 - Perte de 457 millions d'euros (le bug informatique le plus coûteux de l'histoire).
- Bug de l'an 2000 ou bug du millénaire :
 - Dysfonctionnement lorsque les dates sont postérieures au 31 Décembre 1999 (calculs se basant uniquement sur les 2 derniers chiffres de l'année).
- Panne de 24H dans le réseau téléphonique de Bouygues Télécom en 2005:
 - Perte de 8 millions d'euros.



Objectifs

- S'assurer que le produit répond aux exigences.


- Conformité au cahier des charges.

-  *Validation.*

- S'assurer que le produit est construit correctement.

-  *Vérification.*

- Améliorer la productivité des équipes.



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).



► Principes (1/2)

- Les tests montrent la présence de défauts:
 - Les tests peuvent prouver la présence de défauts, mais ne peuvent en prouver l'absence.
 - ➔ Les tests réduisent la probabilité que des défauts restent cachés dans le logiciel.
- Les tests exhaustifs sont impossibles:
 - Tester toutes les combinaisons d'entrées et de pré-conditions n'est faisable que pour des cas triviaux.
 - ➔ Utilisation de l'analyse des risques et des priorités pour focaliser les efforts de tests.



Principes (2/2)

- Tester tôt :
 - Les activités de tests devraient commencer aussi tôt que possible dans le cycle de développement du logiciel et devraient être focalisés vers des objectifs définis.
- Les tests dépendent du contexte:
 - Les tests sont effectués différemment dans des contextes différents.



Types de défauts logiciels (1/4)



- **Bug**

- Dysfonctionnement de l'ordinateur causé par un défaut de conception ou de réalisation d'un programme informatique.
- Le mot anglais bug (insecte, bogue) vient du jargon des ingénieurs de matériel et représente les problèmes qui y survenaient.

- **Crash applicatif ou Deny of Service**

- Déclenchement d'un mécanisme à la fois matériel et logiciel qui met hors service le logiciel défaillant lors de la tentative de ce dernier d'effectuer des opérations impossibles à réaliser (exceptions : division par zéro, recherche d'informations inexistantes, etc.)



Types de défauts logiciels (2/4)



- **Fuite de mémoire**

- Dysfonctionnement dans les opérations d'allocation de mémoire. La quantité de mémoire utilisée par le logiciel défaillant va en augmentant continuellement et gêne le déroulement des autres logiciels et les entraîne à des dysfonctionnements

- **Vulnérabilité**

- Faiblesse dans un système informatique permettant à un attaquant de porter atteinte à l'intégrité de ce système, c'est-à-dire à son fonctionnement normal, à la confidentialité et l'intégrité des données qu'il contient. On parle aussi de faille de sécurité informatique.



► Types de défauts logiciels (3/4)

- **Faute de segmentation**

- Dysfonctionnement dans des opérations de manipulations de pointeurs ou d'adresses mémoire.
- Lecture ou écriture des informations dans un emplacement de mémoire (segment) qui n'existe pas ou qui ne lui est pas autorisé.
- Détection des exceptions provoquant la mise hors service du logiciel défaillant.

- **Buffer Overflow**

- Dépassement de tampon ou débordement est un bug par lequel un processus, lors de l'écriture dans un tampon, écrit à l'extérieur de l'espace alloué au tampon, écrasant ainsi des informations nécessaires au processus.
- Le comportement de l'ordinateur devient imprévisible. Il en résulte souvent un blocage du programme, voire de tout le système. C'est une faille de sécurité courante des serveurs informatiques.



Types de défauts logiciels (4/4)



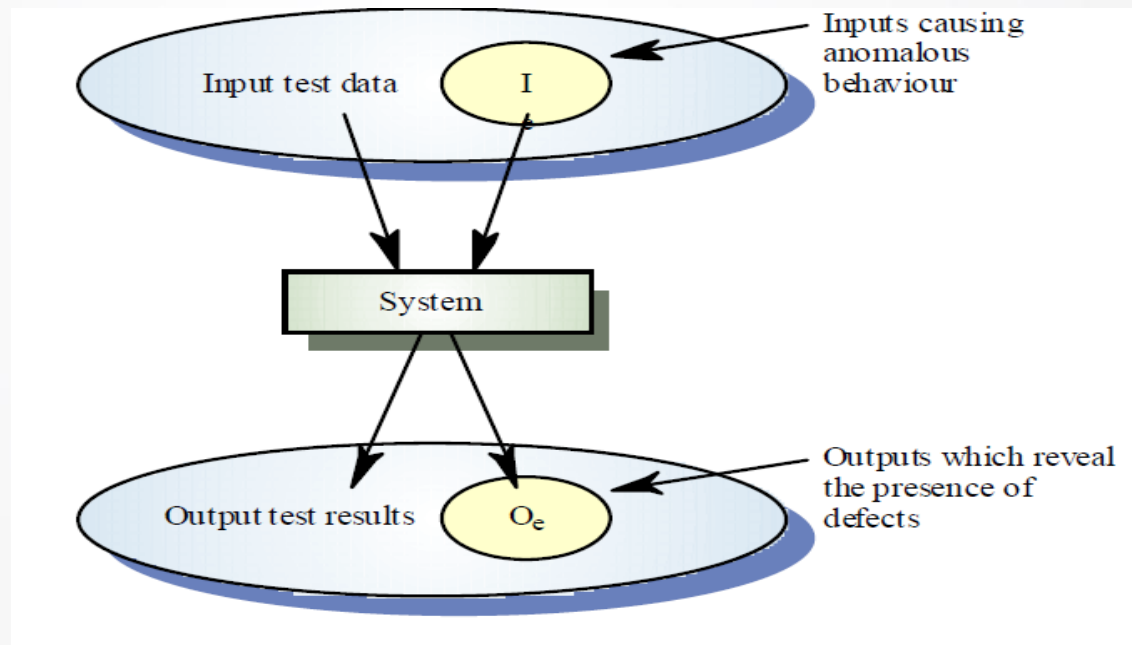
▪ Deadlock ou inter blocage

- Dysfonctionnement durant lequel plusieurs processus s'attendent mutuellement, c'est à dire qu'ils attendent chacun que l'autre libère les ressources qu'il utilise pour poursuivre.
- Ressources verrouillées durant les attentes, pouvant bloquer d'autres processus et par effet domino bloquer l'ensemble du système.
- Un mécanisme de prévention provoque l'annulation de l'opération lorsque la durée d'attente dépasse le délai admissible (anglais *timeout*).

► Méthodes de tests (1/3)

■ Méthode Boite noire

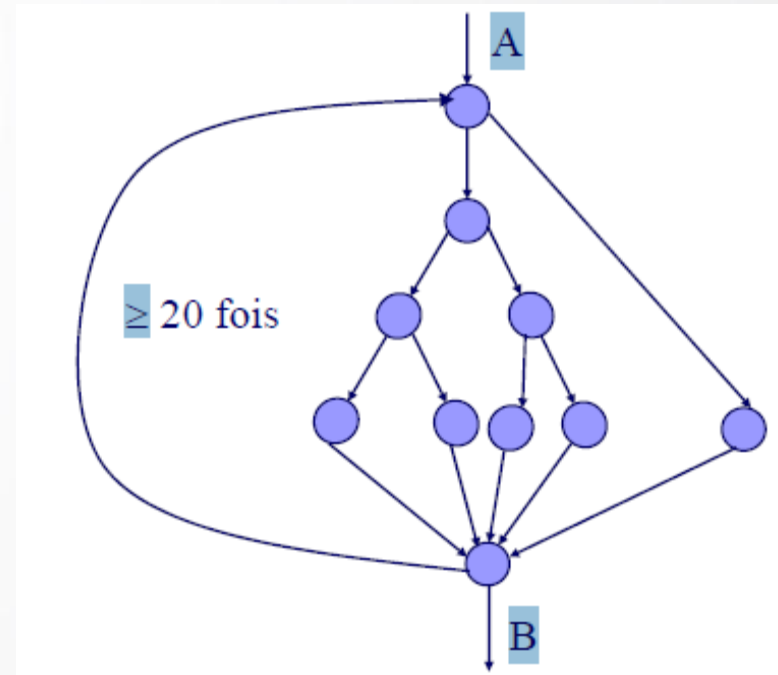
- Test fonctionnel
- Tester les fonctionnalités du produit : Est-ce que le produit répond aux exigences du client?




► Méthodes de tests (2/3)

■ Méthode Boite blanche

- Test structurel.
- Test de fonctionnement et de structure de code : Est-ce que le code produit est bien fonctionnel?
- Test d'optimisation et de structure de code.
- Processus :
 1. Critères de test.
 2. Tous les chemins.
 3. Toutes les branches.
 4. Toutes les instructions.



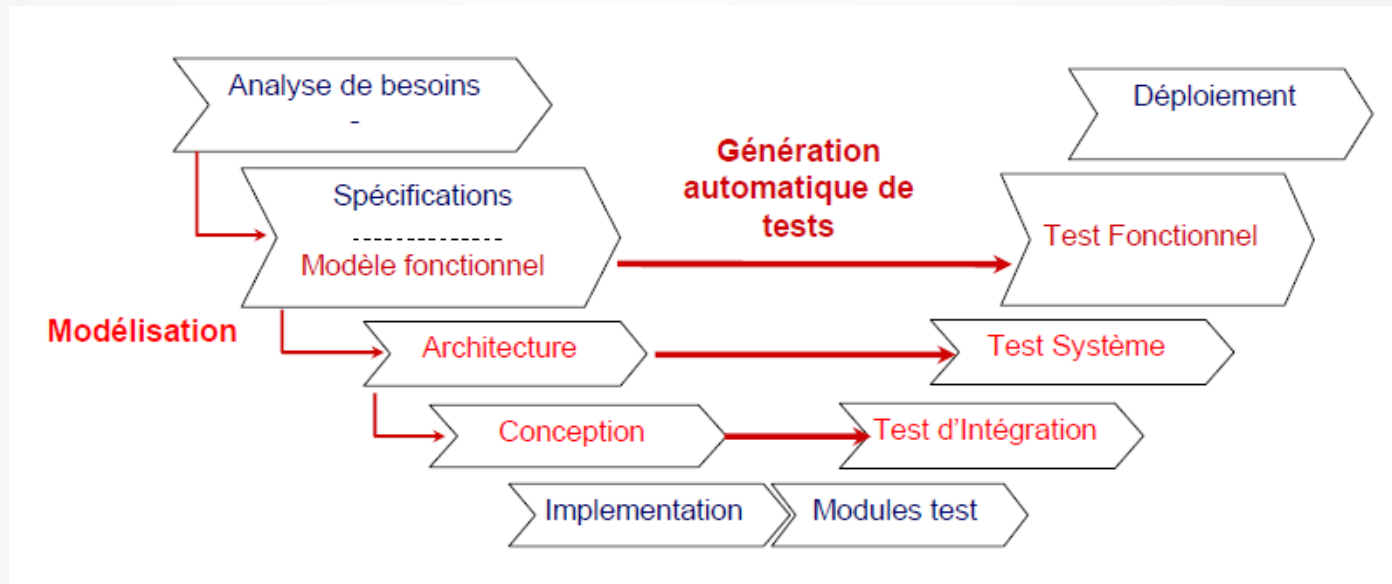


Méthodes de tests (3/3)

- **Boite Grise**

- Les techniques fonctionnelles et structurelles sont utilisées de façon complémentaire.
- Sélectionner un bout de code et tester son comportement fonctionnel.

Test à partir d'un modèle de Cycle de vie





► Types de tests (1/5)

- Test nominal (de bon fonctionnement)
 - Entrer des données volontairement valides.
 - ➔ Test to pass.
- Test de robustesse (de défense)
 - Entrer des données volontairement invalides.
 - ➔ Test to fail.



► Types de tests (2/5)

- Test de performance

- Déterminer la performance d'une application en la soumettant à des demandes de ressources anormales (test de stress) ou à des valeurs aux limites (test de charge).

- Test unitaire

- Tester les fonctions (ou les modules) de code par les programmeurs.

➡ Test nécessitant une connaissance approfondie du code de l'application.



► Types de tests (3/5)

■ Test d'intégration

- Valider le bon fonctionnement d'une ou de plusieurs parties (modules de code, librairies, applications individuelles) développées indépendamment avec le reste de l'application.

■ Test Système

- Tester l'entière application par la méthode boîte noire dans l'environnement qui imite la situation réelle d'utilisation de l'application.



► Types de tests (4/5)

- **Test de non régression (Tests liés au changement)**
 - Reprendre un ensemble de cas de tests après avoir fixé les bogues ou les modifications du logiciel ou de l'environnement.
 - Les outils de test automatisé peuvent être extrêmement utiles pour ce type de test.
- **Test d'administration**
 - Se Focaliser sur les aspects d'administration (tests des backups et restaurations; reprise après sinistre; gestion des utilisateurs; tâches de maintenance; chargements de données et tâches de migration, etc.)



► Types de tests (5/5)

■ Test d'utilisabilité

- Valider si l'application est facile à utiliser
- Test subjectif dépendant des utilisateurs finaux.
- Les programmeurs et les testeurs ne sont pas impliqués

■ Test de sécurité

- Détecter les intrusions et les failles de sécurité par le biais d'une vérification périodique de vulnérabilité de sécurité



Le testing: un métier

- Le test des logiciels est un métier à part entière.
- Activité dans le cycle de développement où l'on peut voir toutes les fonctionnalités d'un produit logiciel.
- Le test logiciel est le maillon principal dans la chaîne d'assurance qualité produit.
- Le test logiciel pourra avoir pour but de qualifier un logiciel ou certifier un produit.
- le processus de test fait intervenir : responsables de tests, concepteurs de tests, testeurs.



Exemple d'une fiche de test



Test Sheet		By : mkh
Description : Importing the external reference data		Test sheet : Gen_DbInst_001
Tested modules : IRIS Database		
Context : Database service name initialised Iris user created on the oracle service name Dumped external data file is given Condition of success : All expected results are observed		
STEP	ACTION	EXPECTED RESULT
1	Launch Imp command and supply the irisName account login	Connection succeeded
2	Supply the path/name of the dumped external data And respond to all appearing questions	Import of the external data finished with no error
3	Connect as irisName/irisName	Connection succeeded
4	Select * from tab	Following entries are displayed: ALPS_B_NUMBER_GROUP TABLE ALPS_ENGINEERING_ROUTE TABLE ALPS_RD_CARRIER TABLE ALPS_RD_CID_DEFINITIONS_VIEW TABLE
Notes : this step is relative to XXX soft platform test For AZURE this step will be replaced by running a script to configure database links to external data in which case when starting: select * from USER_DB_LINKS; all needed data tables are displayed		