

# Résume Test Logiciel

# Pourquoi tester?

- Parce que tout être humain peut faire une erreur (méprise) qui produit un défaut (bogue) dans le code, dans un logiciel ou un système, ou dans un document
- Et si un défaut du code est exécuté, il peut générer une défaillance (ou pas)

# Pourquoi tester ?

## I. Limiter le coût des bogues

- Baisse de réputation
- Économique:
  - Perte de contrat
  - Application de pénalités
  - estimation à 64 milliards de \$/an rien qu'aux U.S.A. (2002)
- Humains, environnementaux, etc.

# Pourquoi tester?

- Quelques exemples:
  - Ariane 5 (1996):
    - problème de conversion de nombres,
    - Coût : 370 millions de dollars
    - [http://fr.wikipedia.org/wiki/Vol\\_501\\_d'Ariane\\_5](http://fr.wikipedia.org/wiki/Vol_501_d'Ariane_5)
  - Machine de radio-thérapie Therac-25, (1985-87) :
    - problème de synchronisation.
    - Coût : plusieurs vies
    - <http://fr.wikipedia.org/wiki/Therac-25>
  - Panne d'électricité aux États-Unis et au Canada (2003)
    - Impact sur 55 millions d'habitants.
    - Coût : 6 milliards de dollars

# Pourquoi tester?

- USS Yorktown (1998)
  - Une division par zéro coupe les moteurs
- Ariane 5 (1996)
  - Mauvaise réutilisation
- Mars orbiter (1999)
  - Plusieurs unités de mesure
- Système de guidage (2002)
  - Initialisation erronée
- The Patriot and the Scud
  - mauvais arrondi dans une soustraction

# Pourquoi tester?

2. Assurer la qualité d'une application
  - Pour atteindre les objectifs de normes/lois (ex: DO = I78B pour l'aviation)
  - Pour atteindre un bon rapport qualité/prix
  - Pour maintenir la confiance du client

# Définition(s) du test

- *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)

# Objectifs du test

- Trouver des défauts
- Augmenter le niveau de confiance en la qualité d'un logiciel
- Fournir aux décideurs *go/no go* un état le plus précis possible de la qualité du logiciel
- Prévenir des défauts



# Les 7 principes généraux

1. Les tests montrent la présence de défauts
  - Jamais leur absence
2. Les tests exhaustifs sont impossibles
  - Exemple Sopra: test d'une IHM contenant 20 écrans, 4 menus, 3 options, 10 champs, 2 types de données, 100 valeurs possibles
  - $20 \times 4 \times 3 \times 10 \times 2 \times 100 = 480\,000$  cas de tests
  - Un expert (10s par test!) mettrait 180 jours
3. Il faut tester le plus tôt possible

# Les 7 principes généraux

4. Les défauts sont généralement regroupés
  - Loi de Pareto: 80% des défauts sont concentrés dans 20% des modules
5. Paradoxe du pesticide
  - Les mêmes tests ne décèlent plus de défauts
6. Les tests dépendent du contexte
  - Les tests de sécurité dépendent de l'application testée

# Les 7 principes généraux

## 7. L'illusion de l'absence d'erreur

- Trouver et corriger des défauts ne sert à rien si le système conçu est inutilisable (opérativité, conformité, attractivité, performance)

# Tester <> Déboguer

- Tester et déboguer, ce n'est pas la même activité et les responsables sont différents:
  - Les testeurs testent
  - Les développeurs déboguent
- Déboguer: activité de développement permettant d'analyser, trouver et supprimer les causes de la défaillances