

Les tests de logiciel

Sofien Benharchache Stella Zevio

Université Montpellier II

16 octobre 2014

1. Introduction
2. Méthodes de vérification et validation
3. Principes de base et mauvaises pratiques
4. Différentes classifications de tests
5. Frameworks et outils de test
6. Conclusion

1. Introduction

- 1.1 Qu'est-ce qu'un logiciel ?
- 1.2 Qu'est-ce qu'un test de logiciel ?
- 1.3 Qu'est-ce qu'un bug ?
- 1.4 Pourquoi tester un logiciel ?

Qu'est-ce qu'un logiciel ?

- ▶ Des documents de gestion de projet
- ▶ La spécification
- ▶ La conception
- ▶ Le code source
- ▶ L'exécutable

Qu'est-ce qu'un test de logiciel ?

- ▶ Propriétés
- ▶ Résultat
 - ▶ Validation dynamique

norme IEEE-STD729, 1983

Qu'est-ce qu'un bug ?

Erreur → Défaut → Anomalie

Pourquoi tester un logiciel ?

Éviter les bugs

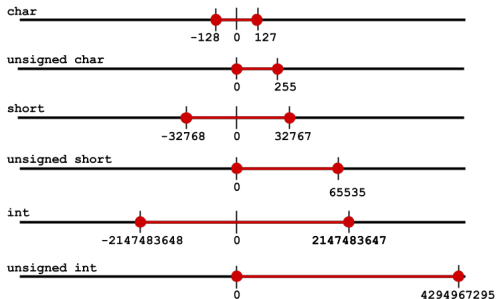
Figure: Sonde Mariner 1, 1962



Pourquoi tester un logiciel ?

Éviter les bugs

Figure: Ariane 5 vol 501, 1996



Pourquoi tester un logiciel ?

Éviter les bugs

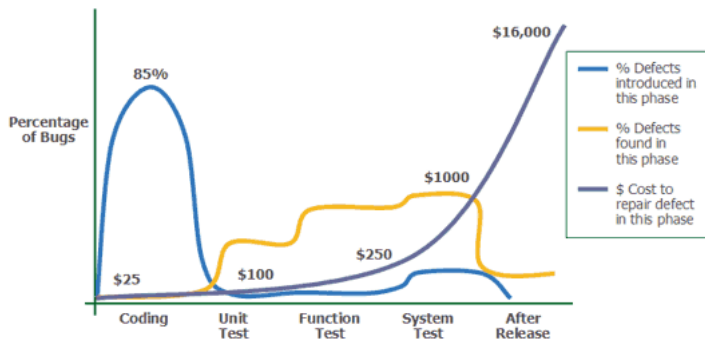
Figure: Therac-25, 1985-1987



Pourquoi tester un logiciel ?

Réduire les coûts

Figure: Applied Software Measurement, Capers Jones, 1996



Pourquoi tester un logiciel ?

Réduire les coûts

- ▶ 30-50% des coûts de développement
- ▶ 1/3 dans le planning

Pourquoi tester un logiciel ?

Assurer la qualité

- ▶ Capacité fonctionnelle
- ▶ Facilité d'utilisation
- ▶ Fiabilité
- ▶ Performance
- ▶ Maintenabilité
- ▶ Portabilité

Sommaire - Méthodes de vérification et validation

1. Introduction
2. Méthodes de vérification et validation
 - 2.1 Méthodes de vérification et validation
 - 2.2 Comparaison des méthodes

Méthodes de vérification et validation

Vérification et validation

Vérification *"are we building the product right ?"*

- ▶ Preuve formelle, model-checking

Validation *"are we building the right product ?"*

- ▶ Test

Preuve formelle, model-checking

- ▶ Exhaustifs
- ▶ Mise en œuvre difficile
- ▶ Limitations par rapport à la taille du système
- ▶ Distance entre le modèle et la réalité

Test

- ▶ Nécessaire mais pas suffisant
- ▶ Nécessite une exécution réelle du système
- ▶ Ne garantit pas l'absence d'erreur

Sommaire - Principes de base et mauvaises pratiques

1. Introduction
2. Méthodes de vérification et validation
3. Principes de base, bonnes et mauvaises pratiques
 - ▶ Mauvaises pratiques
 - ▶ Bonnes pratiques
 - ▶ Principes de base

Principes de base, bonnes et mauvaises pratiques

Mauvaises pratiques

Tenter d'isoler un bug à l'aide d'appels d'affichage.

Principes de base, bonnes et mauvaises pratiques

Bonnes pratiques

1. Objectifs
2. Données de test, résultats attendus
3. Exécution, collecte des résultats
4. Comparaison, établissement d'un verdict

Principes de base et mauvaises pratiques

Principes de base

1. Ne pas tester ses propres programmes
2. Chercher les erreurs
3. Définir les résultats attendus avant d'exécuter un test
4. Inspecter minutieusement les traces
5. Prendre en compte les entrées invalides ou incohérentes
6. Vérifier que le programme n'en fait pas plus que sa spécification

Sommaire - Différentes classifications de tests

1. Introduction
2. Méthodes de vérification et validation
3. Principes de base et mauvaises pratiques
4. Différentes classifications de tests
 - ▶ Classification selon le niveau
 - ▶ Cycle de développement en V d'un logiciel
 - ▶ Classification selon le niveau d'accessibilité
 - ▶ Boîte noire
 - ▶ Boîte blanche
 - ▶ Classification selon les caractéristiques

Classifications de tests

Classification selon le niveau

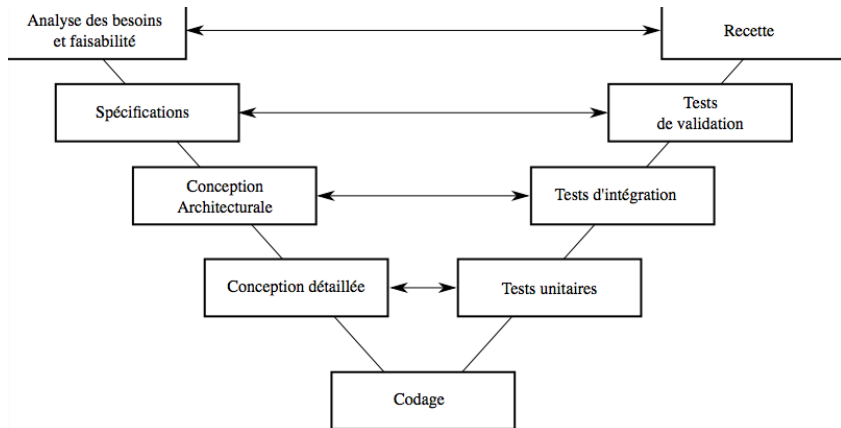
- ▶ Test unitaire
- ▶ Test d'intégration
- ▶ Test système
- ▶ Test d'acceptation ou recette

Comité français du test logiciel

Classification selon le niveau

Cycle de développement en V

Figure: Cycle de développement en V d'un logiciel



Classification de tests

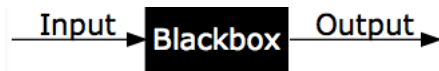
Classification selon le niveau d'accessibilité

- ▶ Boîte noire
- ▶ Boîte blanche

Classification selon le niveau d'accessibilité

Boîte noire

Figure: Boîte noire



Classification selon le niveau d'accessibilité

Boîte blanche

Code source déjà écrit.

Classification de tests

Classification selon les caractéristiques

- ▶ Performance
- ▶ Robustesse
- ▶ Vulnérabilité
- ▶ Non régression

Liste non exhaustive

Sommaire - Frameworks et outils de test

1. Introduction
2. Méthodes de vérification et validation
3. Principes de base et mauvaises pratiques
4. Différentes classifications de tests
5. Frameworks et outils de test

- ▶ Frameworks (xUnit)
- ▶ Objets de type mock (programmation orientée objet)
- ▶ JMeter
- ▶ SoapUI...
- ▶ Outils de couverture de code (TestWellCTC++, Emma, Xdebug...)

1. Introduction
2. Méthodes de vérification et validation
3. Principes de base et mauvaises pratiques
4. Différentes classifications de tests
5. Frameworks et outils de test
6. Conclusion

Conclusion

Tester, c'est très important

Tester, c'est :

- ▶ Une activité à part entière
- ▶ Avoir une vision globale d'un logiciel à toutes les étapes de son développement

Tester, ça doit être :

- ▶ Une activité rigoureuse
- ▶ De plus en plus automatique

Conclusion

Tester, c'est très important

Testez vos programmes !