



# 08

## Chapitre

# Test logiciel

**1I2AC1 : Génie logiciel et Conception orientée objet**

Régis Clouard, ENSICAEN - GREYC

« Durant le débogage, les novices insèrent du code correctif,  
alors que les experts suppriment du code défectueux. »

**Richard Pattis**

# Plan du chapitre

---

1  
Généralités  
sur les tests

2  
Tests  
unitaires

3  
Code testable

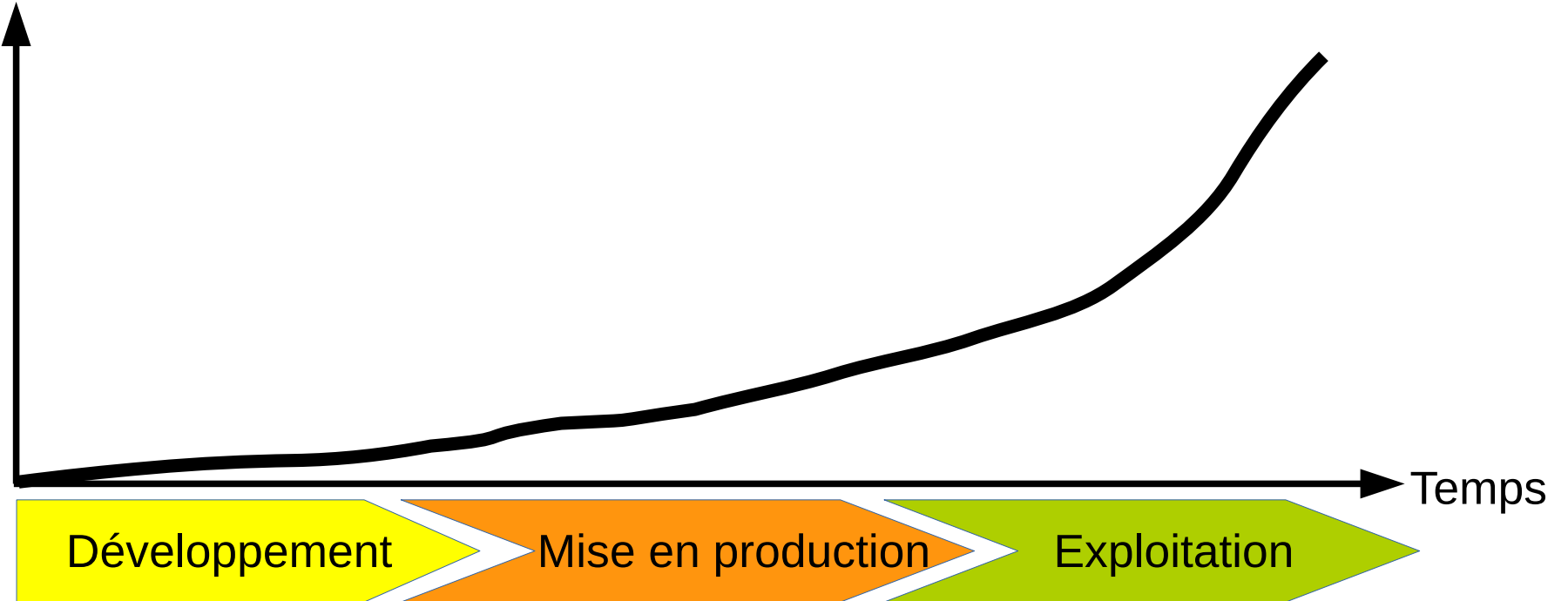
5  
Développement  
dirigé par  
les tests

4  
Frameworks de test  
JUnit  
Mockito

# Quand tester ?

- Le plus tôt possible
  - Coût d'un bug dans le cycle de vie d'un logiciel :

Charge de travail



# Quand tester ?

---

- Dans l'approche **classique** (ie, prédictive) les tests sont réalisés en fin de projet après la programmation de toutes les fonctionnalités
- Dans l'approche **agile**, les tests sont réalisés :
  - Après l'écriture du code d'une fonctionnalité
  - Après la découverte d'un bug
  - Pendant le développement d'une fonctionnalité
    - ▶ Méthode de développement
      - 1) Programmation vite faite, pas propre : *Quick and Dirty*
      - 2) Programmation des tests
      - 3) Refonte du code : *Refactor*

# TDD : Test Driven Development

- Et si les tests étaient programmés avant la fonctionnalité ?
  - TDD : développement dirigé par les tests
  - Nouvelle méthode de développement
    - 1) Programmation des tests
    - 2) Programmation Vite faite, Pas propre : *Quick and Dirty*
    - 3) Refonte du code : *Refactor*
- Le TDD est plus que simplement inverser l'ordre
  - Les tests définissent le cahier des charges de la fonctionnalité
  - Règle d'or du TDD :
    - ▶ « Ne jamais écrire une ligne de code fonctionnel sans qu'une ligne de code de test ne l'exige. »
    - ▶ Éviter d'ajouter du code non nécessaire : principe YAGNI

# Démo TDD : bowling

# Que retenir de ce chapitre ?

---

- Les tests sont une obligation
- Ils doivent forcément accompagner le code d'une application
- Il y a différents types de test selon le niveau considéré
  - Unitaire
  - Intégration
  - Système
- Les tests peuvent être écrits avant ou après l'écriture d'une fonctionnalité ou la correction d'un bug
- Le code des tests étant du code, il doit donc être propre au même titre que le code source