



05

Chapitre

Qualité logicielle

1I2AC1 : Génie logiciel et Conception orientée objet

Régis Clouard, ENSICAEN - GREYC

« Je ne suis pas un grand programmeur.
Je suis juste un bon programmeur avec de bonnes habitudes. »
Kent Beck (créateur de la méthode X-Programming)

Définition de la qualité logicielle

2

- Définition
 - La **qualité** logicielle englobe l'ensemble des caractéristiques d'un logiciel qui affecte sa capacité à satisfaire des besoins **exprimés** ou **implicites** en termes de fonctionnalités, délais et coûts.
 - L'**assurance qualité** (QA) est l'ensemble des activités planifiées et systématiques de toutes les actions nécessaires pour fournir une assurance que la qualité du logiciel est conforme aux exigences et aux attentes établies.
- Une appréciation globale de la qualité tient compte à la fois de :
 - Facteurs **externes** directement observables par l'utilisateur (MOA)
 - Facteurs **internes** observables par les ingénieurs du développement (MOE)

Plan du chapitre

3

1

La qualité logicielle
est régie
par des normes

Qualité : normes ISO 9000

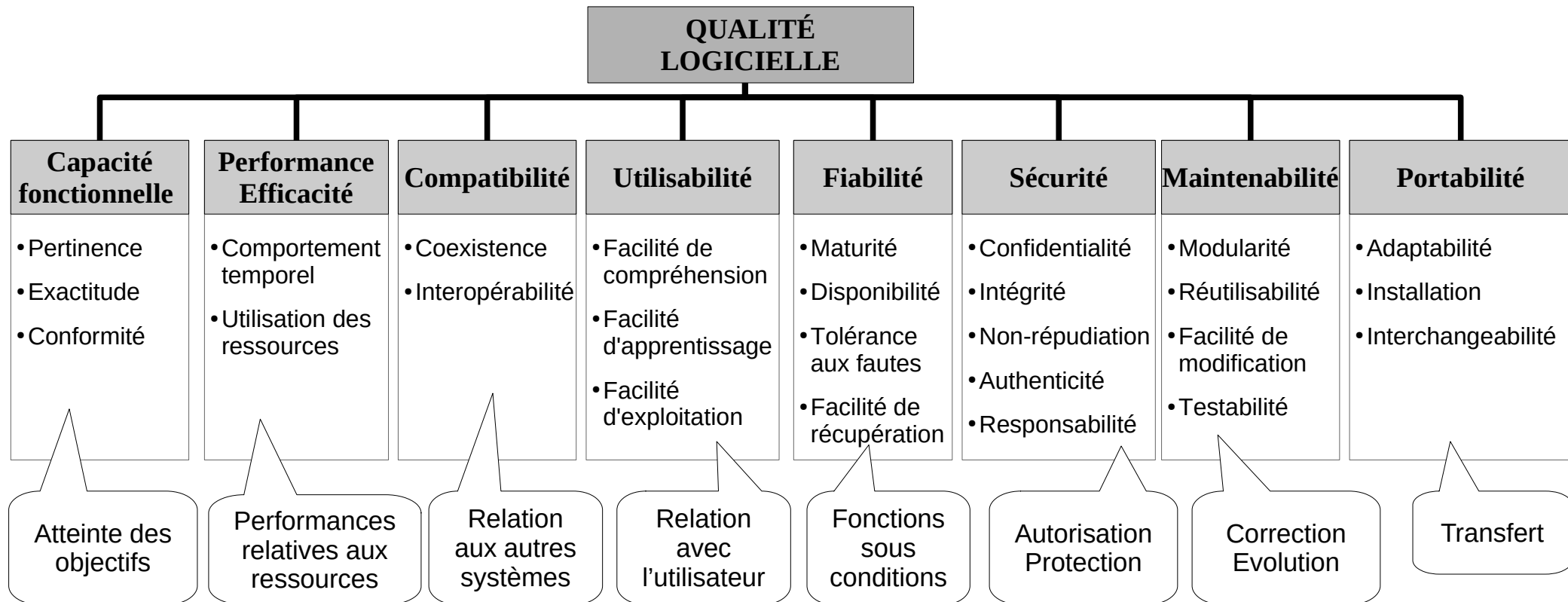
4

- Guide pour le management et l'organisation d'un projet (et plus généralement d'une entreprise)
- Éléments de base pour le pilotage d'un projet et des orientations reconnues comme étant les bonnes pratiques pour satisfaire les clients et s'améliorer
- Normes déclinées pour le génie logiciel → 2 exemples de norme

Qualité logicielle : norme emblématique

5

- **ISO/CEI 25010 : Liste les indicateurs clés de performance**
 - 8 aspects et 35 caractéristiques **mesurables** → **KPI** (Key Performance Indicator)



Assurance qualité logicielle : norme emblématique

6

- **ISO/CEI 12207 : Préconisation pour garantir un logiciel de qualité**
 - Modèle de pilotage basé sur le cycle en V
 - Documentation exhaustive pour chaque phase

Plan du chapitre

7

1

La qualité logicielle
est régie
par des normes

2

Bilan des normes

Inapplicabilité des indicateurs

8

■ ISO/CEI 25010

- Pas de réelle mesure objective de la qualité d'un logiciel à partir de ces indicateurs (hors mesure de performance)
 - ▶ Mesure **empirique** des indicateurs → évaluation **subjective** de la qualité
- Les KPI sont inutilisables pour mesurer la qualité d'un logiciel
 - ▶ C'est un leurre de vouloir mesurer la qualité d'un logiciel à partir de KPI

Capacité fonctionnelle	Performance Efficacité	Compatibilité	Utilisabilité	Fiabilité	Sécurité	Maintenabilité	Portabilité
• Pertinence • Exactitude • Conformité	• Comportement temporel • Utilisation des ressources	• Coexistence • Interopérabilité	• Facilité de compréhension • Facilité d'apprentissage • Facilité d'exploitation	• Maturité • Disponibilité • Tolérance aux fautes • Facilité de récupération	• Confidentialité • Intégrité • Non-répudiation • Authenticité • Responsabilité	• Modularité • Réutilisabilité • Facilité de modification • Testabilité	• Adaptabilité • Installation • Interchangeabilité

Inapplicabilité des indicateurs

9

- Mesurer la qualité d'un logiciel à partir d'indicateurs, c'est comme noter la qualité d'un devoir de philosophie à partir de :
 - ▶ Nombre de fautes d'orthographe
 - ▶ Longueur des phrases
 - ▶ Nombre de citations

Inapplicabilité de l'assurance qualité

10

■ ISO/CEI 12027

- Nous l'avons déjà démontré :
 - ▶ le cycle de vie en V n'offre pas les meilleures garanties de succès des projets
 - ▶ La documentation est, au mieux, inutile, au pire, nuisible.

Plan du chapitre

11

1

La qualité logicielle
est régie
par des normes

2

Bilan des normes

3

Le point de vue
de l'agilité

Qualité logicielle : Facteurs externes

12

- Le seul juge de la qualité externe d'un logiciel est l'**utilisateur**
- Mise en pratique
 - Collaboration constante entre la MOE et la MOA
 - Cycle de développement itératif et incrémental à base de MVP

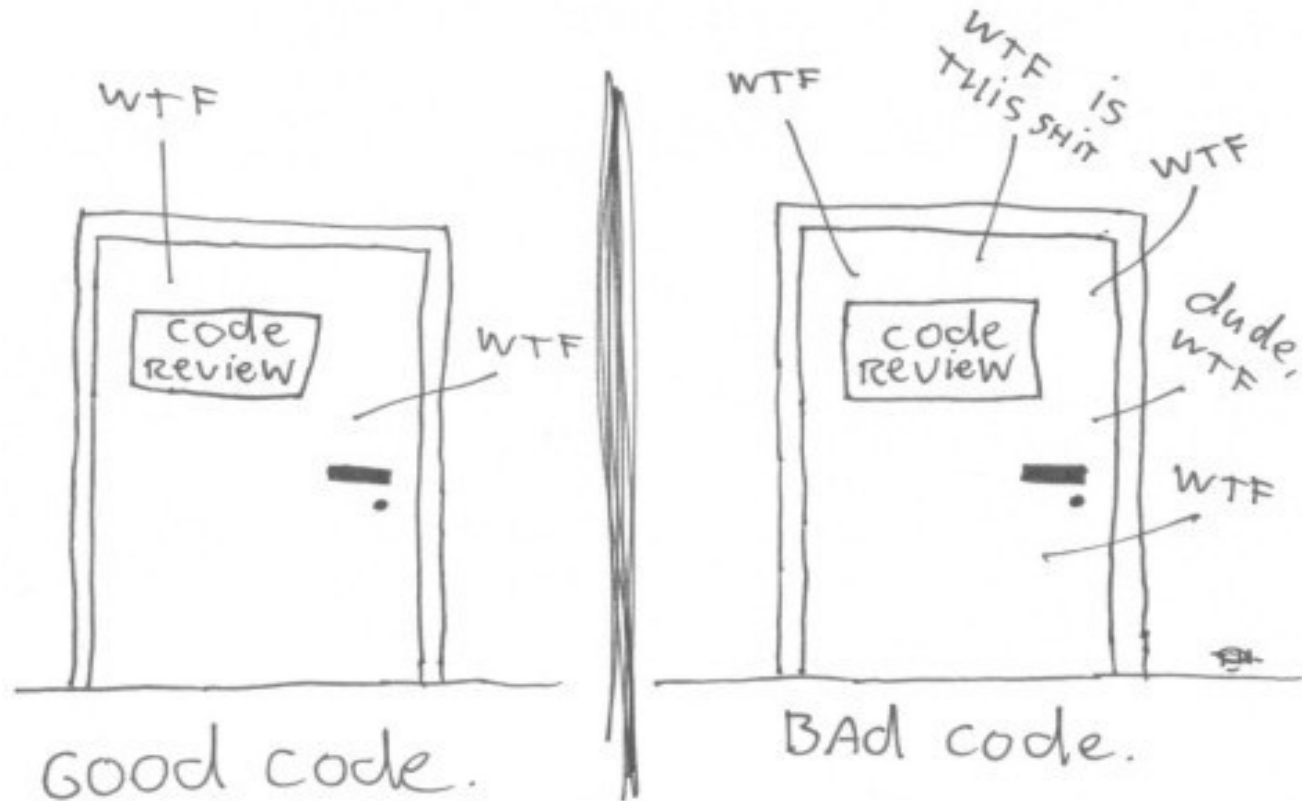
Qualité logicielle : Facteurs internes

13

- La **qualité interne d'un logiciel** résulte de la **qualité de son code**
 - Il ne suffit pas qu'un logiciel soit efficace, il faut en plus qu'il soit bien conçu
 - Le meilleur juge de la qualité est le développeur qui reprend le code ...

The ONLY VALID MEASUREMENT OF CODE QUALITY: WTFs/minute *

* Saperlipopettes / minute



Assurance qualité : artisanat du logiciel

15

- En agilité, le développeur se voit comme un artisan du logiciel
- **Software craftsmanship**
 - Élever le développement logiciel au rang de discipline scientifique
 - ▶ Pratique d'ingénieurs
 - ▶ Professionnalisme
 - ▶ Valeurs
 - Partage de compétences
 - Humilité
 - Pragmatisme
 - Le code est un chef-d'œuvre dont l'équipe doit être fière

La qualité du code selon l'artisan du logiciel

16

- Qualité recherchée (et non mesurée) pour le code :
 - 1) **Simplicité** (principe *KISS : Keep It Simple, Stupid*)
 - 2) **Propreté** (*clean code*)
 - 3) **Testé**

L'atelier de l'artisan

17

- Un environnement de développement professionnel
 - IDE (eg, *IntelliJ*, *Visual Studio*)
 - Gestion de version (eg, *Git*)
 - Moteurs de production (eg, *Makefile* (*cmake*), *Gradle*, *Maven*)
 - Outils d'intégration continue (eg, *GitLab CI/CD*, *Jenkins*)
 - Des outils de déverminage et de profilage de code (eg, *valgrind*, *gprof*)
- Des analyseurs de la propreté du code (pas des métriques mais des alertes)
 - Outils d'inspection d'*IntelliJ*
 - *SonarQube* : une référence dans le domaine de l'analyse de code

Plan du chapitre

18

1

La qualité logicielle
est régie
par des normes

2

Bilan des normes

3

Le point de vue
de l'agilité

4

Exemple d'atelier pour
le génie logiciel avec
IntelliJ IDEA

Que retenir de ce chapitre

19

- La qualité logicielle est impossible à mesurer
- Il existe bien des normes mais concrètement elles n'apportent rien d'objectif pour produire des logiciels de qualité
- L'agilité considère alors que la qualité d'un logiciel est la résultante de la qualité de son code
- Pour cela, il faut utiliser des environnements de développement sophistiqués qui permettent d'obtenir des critiques du code.