



05

Chapitre

Qualité logicielle

1I2AC1 : Génie logiciel et Conception orientée objet

Régis Clouard, ENSICAEN - GREYC

« Le débogage est deux fois plus difficile que l'écriture de code.
Donc, si vous écrivez du code de la manière
la plus intelligente possible, vous n'êtes, par définition,
pas assez intelligent pour le déboguer. »

Brian Kernighan

Définition de la qualité logicielle

■ 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)

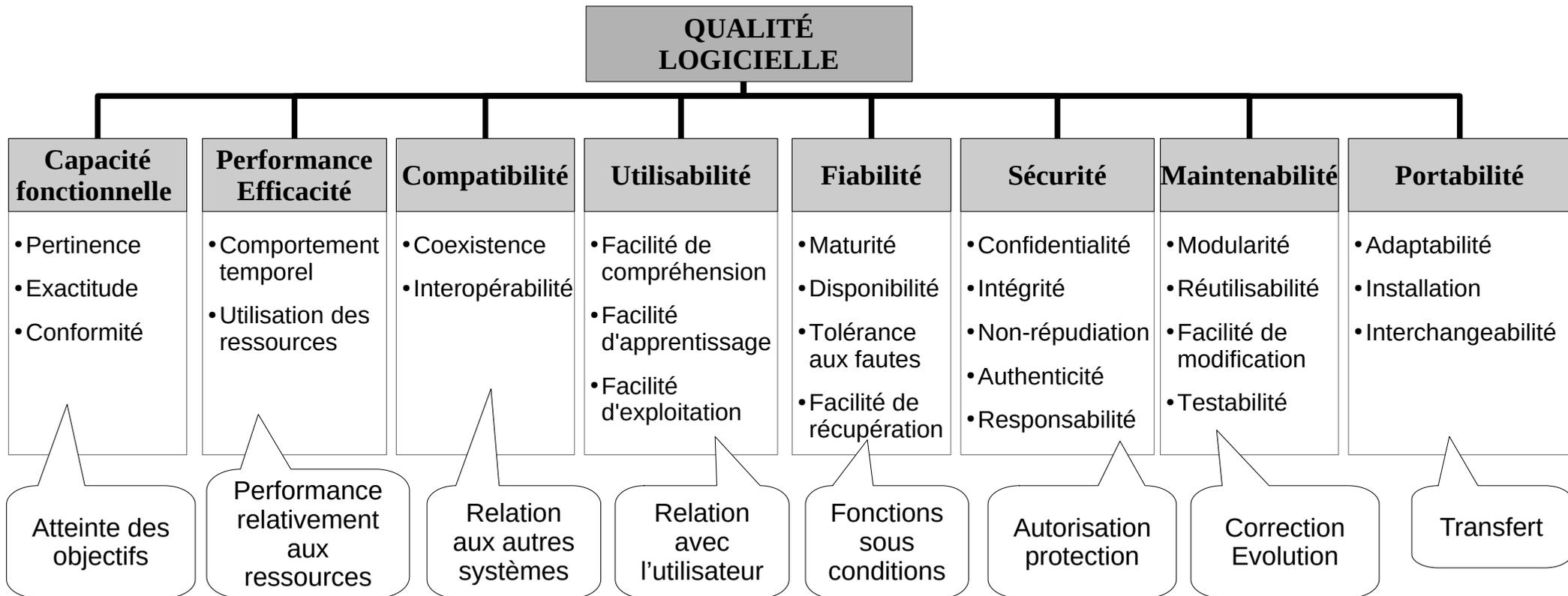
**La qualité logicielle
est régie
par des normes**

Qualité : normes ISO 9000

- 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 logiciel

Qualité logicielle : norme emblématique

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



Assurance qualité logicielle : norme emblématique

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

Bilan des normes

Inapplicabilité des indicateurs

■ ISO/CEI 25010

- Pas de réelle mesure objective de la qualité d'un logiciel à partir de ces indicateurs (hors performance)
 - ▶ Mesure **empirique** des indicateurs → qualité **subjective**
- 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
<ul style="list-style-type: none">• Pertinence• Exactitude• Conformité	<ul style="list-style-type: none">• Comportement temporel• Utilisation des ressources	<ul style="list-style-type: none">• Coexistence• Interopérabilité	<ul style="list-style-type: none">• Facilité de compréhension• Facilité d'apprentissage• Facilité d'exploitation	<ul style="list-style-type: none">• Maturité• Disponibilité• Tolérance aux fautes• Facilité de récupération	<ul style="list-style-type: none">• Confidentialité• Intégrité• Non-répudiation• Authenticité• Responsabilité	<ul style="list-style-type: none">• Modularité• Réutilisabilité• Facilité de modification• Testabilité	<ul style="list-style-type: none">• Adaptabilité• Installation• Interchangeabilité

Inapplicabilité des indicateurs

- 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
- Nous l'avons déjà constaté, la documentation est souvent inutile voire nuisible

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

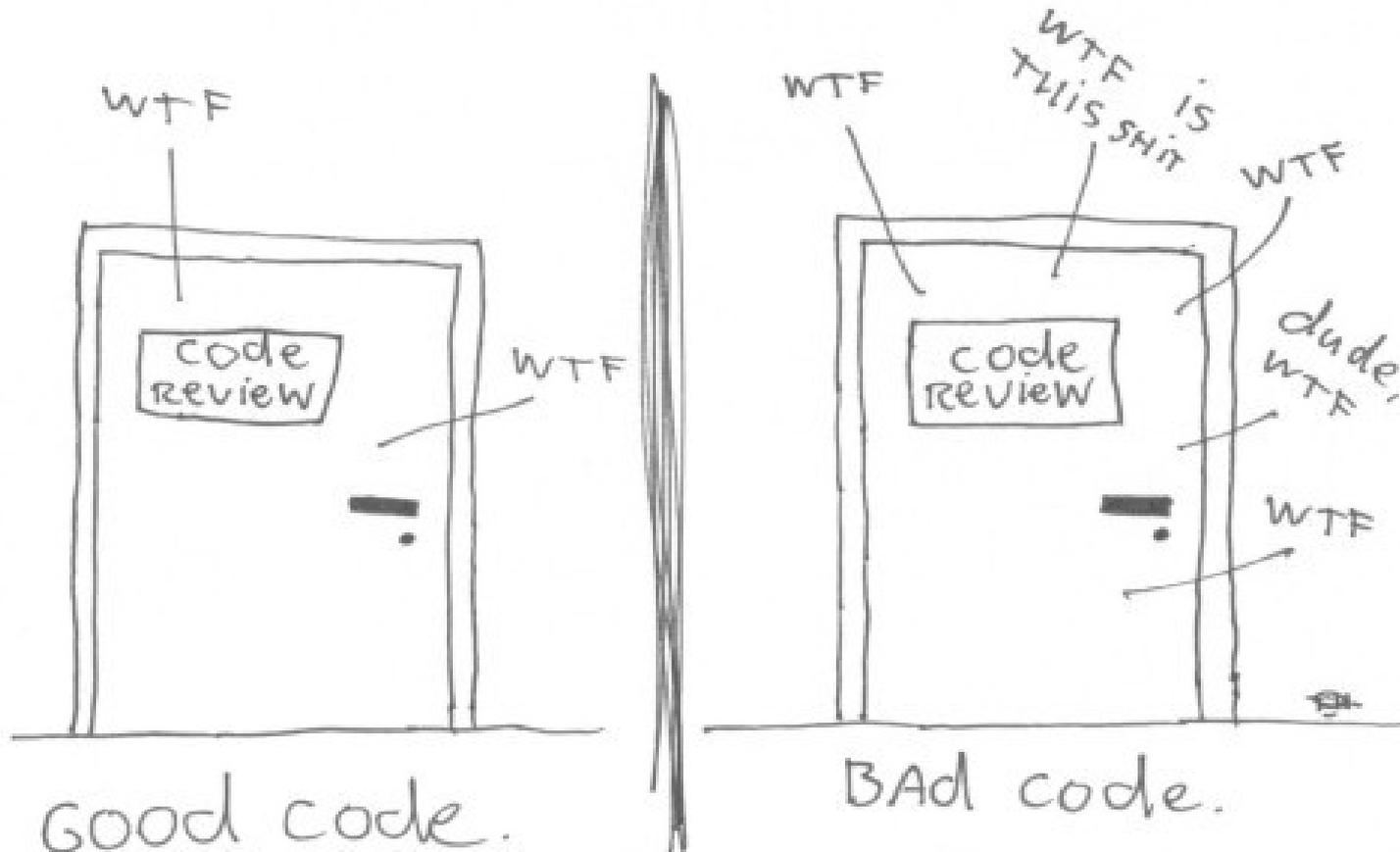
Qualité logicielle : facteurs internes

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**
 - 3) **Testé**

L'atelier de l'artisan

- 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*, *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

Exemple avec IntelliJ

Que retenir de ce chapitre

- 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.