

Qualité logicielle

Mottu - Le Traon – Baudry - Sunyé

Plan

- ▶ Définitions
- ▶ Erreurs
- ▶ Facteurs de qualité
- ▶ Métriques

Définitions

Qualité logicielle

«Conformité aux besoins explicites de fonctionnalité et de performance, aux normes de développement explicitement documentées et aux caractéristiques implicites qui sont attendues de tous les logiciels développés de façon professionnelle»

Software Engineering: A Practitioner's Approach [Roger Pressman](#).6
edition (April 2, 2004)

Qualité logicielle

- ▶ Le degré pour lequel un système, un composant ou un processus répond aux besoins spécifiés.
- ▶ Le degré pour lequel un système, un composant ou un processus répond aux attentes des utilisateurs.

[IEEE]

Qualité logicielle

- ▶ «Conformité aux besoins explicites de fonctionnalité et de performance, aux normes de développement explicitement documentées et aux caractéristiques implicites qui sont attendues de tous les logiciels développés de façon professionnelle»
- ▶ Le degré pour lequel un système, un composant ou un processus répond aux besoins spécifiés.
- ▶ Le degré pour lequel un système, un composant ou un processus répond aux attentes des utilisateurs.

Une question de point de vue

- ▶ Utilisateurs.
- ▶ Développeurs.
- ▶ Gestionnaires.

Une question de spécifications

- ▶ Fonctionnelle, extra-fonctionnelle
- ▶ Formelle, semi-formelle, informelle
- ▶ Commune, transmise, reprise, adaptable

Une question de système

- ▶ Logiciel, embarqué, enfoui
- ▶ Évolution, évolutif
- ▶ Fonctionnement continu, ponctuel
- ▶ Niveau de criticité

Vérification et validation

- ▶ Vérification

- ▶ Validation

Vérifier pour prévenir

- ▶ Une *erreur* du développeur introduit
- ▶ un *défaut* dans le système qui provoquera
- ▶ sa *défaillance* à l'exécution

Erreur

- ▶ Une erreur est une décision inappropriée ou erronée, faite par un développeur, qui conduit à l'introduction d'un défaut.

Défaut

- ▶ Un défaut est une imperfection dans un des aspects du système qui contribue, ou peut potentiellement contribuer, à la survenance d'une ou de plusieurs défaillances
- ▶ Parfois, il faut plusieurs défauts pour causer une défaillance.

Défaillance

- ▶ Une défaillance est un comportement inacceptable présenté par un système.
- ▶ La fréquence des défaillances reflète la fiabilité.

Erreurs

Types d'erreurs

- ▶ Erreur de codage.
- ▶ Erreur de processus.
- ▶ Erreur de documentation.
- ▶ Erreur de données.

Causes principales d'erreurs

- ▶ Mauvaise spécification des besoins.
- ▶ Échec de communication client/développeur.
- ▶ Écarts délibérés des besoins.
- ▶ Erreurs de conception
- ▶ Erreurs de codage
- ▶ Non-respect des règles de codage et de documentation
- ▶ Lacunes du processus de tests
- ▶ Erreurs de documentation.

Coût des erreurs

- ▶ 2002, National Institute of Standards and Technology :
 - ▶ Software bugs, or errors, are so prevalent and so detrimental that they cost the U.S. economy an estimated \$59.5 billion annually, or about 0.6 percent of the gross domestic product, according to a newly released study commissioned by the Department of Commerce's National Institute of Standards and Technology (NIST). At the national level, over half of the costs are borne by software users and the remainder by software developers/vendors.

The study also found that, although all errors cannot be removed, more than a third of these costs, or an estimated \$22.2 billion, could be eliminated by an improved testing infrastructure that enables earlier and more effective identification and removal of software defects. These are the savings associated with finding an increased percentage (but not 100 percent) of errors closer to the development stages in which they are introduced. Currently, over half of all errors are not found until "downstream" in the development process or during post-sale software use.

<http://www.nist.gov/director/planning/upload/report02-3.pdf>

Conséquences de défaillances

- ▶ **Mauvais résultat**
- ▶ **Plantage**
- ▶ **Crash**

- ▶ **PERTES**
 - ▶ Image
 - ▶ Financier
 - ▶ Humain

Coût financier

- ▶ Mariner I (1962)
- ▶ **Cost:** \$18.5 million
- ▶ **Disaster:** The Mariner I rocket with a space probe headed for Venus diverted from its intended flight path shortly after launch. Mission Control destroyed the rocket 293 seconds after liftoff.
- ▶ **Cause:** A programmer incorrectly transcribed a handwritten formula into computer code, missing a single superscript bar. Without the smoothing function indicated by the bar, the software treated normal variations of velocity as if they were serious, causing faulty corrections that sent the rocket off course.

Coût financier

- ▶ Mars Climate Orbiter(1998)
- ▶ **Cost:** \$125 million - 327M\$.
- ▶ **Disaster:** After a 286-day journey from Earth, the Mars Climate Orbiter fired its engines to push into orbit around Mars. The engines fired, but the spacecraft fell too far into the planet's atmosphere, likely causing it to crash on Mars.
- ▶ **Cause:** The software that controlled the Orbiter thrusters used imperial units (pounds of force), rather than metric units (Newtons) as specified by NASA.
(more)

Coût financier

- ▶ The Virtual Case File (2005)
- ▶ Système d'information du FBI
 - ▶ abandonné en avril 2005 : coût 170 M \$
 - ▶ mauvaise spécification, exigences mal exprimées
 - ▶ réutilisation dans un contexte inadapté
 - ▶ trop d'acteurs concurrents
- ▶ Système de gestion des opérations de l'armée américaine en Afghanistan et en Irak (DCGS-A, 2011)
 - ▶ 2.7 milliards de dollars
 - ▶ Ne fonctionne pas : incompatibilité, absence de réponse, plantage
 - ▶ Fonctionnalité principale défectueuse : partage de données
 - ▶ Nouvelle version en développement

Coût financier

- ▶ **USS Yorktown (1998)**
 - ▶ Une division par zéro coupe les moteurs
- ▶ **Retournement d' un avion de chasse au passage de l' équateur**
- ▶ **Ariane 5 (1996)**
 - ▶ Mauvaise réutilisation
- ▶ **Un quart des cartes bancaires allemandes bloquées le 1er janvier 2010 pendant près d'une semaine**
 - ▶ # 30 millions de cartes
 - ▶ Des millions d'euros de dommages et de perte d'image

Coût humain

▶ Therac-25 (1985)

- ▶ Entre 1985 et 1987, le Therac-25 fut impliqué dans au moins six accidents durant lesquels des patients reçurent des doses massives de radiation, parfois de l'ordre de plusieurs centaines de grays. Au moins cinq patients décédèrent des suites de l'irradiation.

▶ **Causes :**

- ▶ L'Atomic Energy of Canada Limited (AECL) ne disposait pas d'un organisme indépendant pour l'évaluation du code source
- ▶ L'AECL avait négligé certaines étapes liées au test du logiciel
- ▶ La documentation du système n'expliquait pas de manière adéquate les codes indiquant des erreurs
- ▶ Le personnel de l'AECL, au début, n'avait pas tenu compte des plaintes ou n'y croyait pas.

Coût humain

- ▶ Therac-25 (1985)

- ▶ **Autres causes:**

- ▶ Absence de dispositif physique pour bloquer le flux si la cible n'était pas en place.
- ▶ Réutilisation de code d'autres modèles qui possédaient cette protection physique.
- ▶ Le logiciel ne pouvait pas vérifier l'état des capteurs.
- ▶ Problème de concurrence (race condition) si l'opérateur changeait les paramètres trop rapidement.
- ▶ Utilisation et incrémentation d'un drapeau. Parfois un dépassement se produisait et désactivait certains tests de sécurité.
- ▶ Programmation en assembleur.

Coût humain

- ▶ Système de missiles Patriot (1991)
- ▶ Cost: 28 soldiers dead, 100 injured
- ▶ Disaster: During the first Gulf War, an American Patriot Missile system in Saudi Arabia failed to intercept an incoming Iraqi Scud missile. The missile destroyed an American Army barracks.
- ▶ Cause: A software rounding error incorrectly calculated the time, causing the Patriot system to ignore the incoming Scud missile.

Coût humain

- ▶ Un canon-robot anti-aérien sud-africain tue neuf soldats dans un mode tout automatique qui avait été rajouté (2007)

Coût indirect et d'image

- ▶ Playstation 3 : le système de jeux en ligne croyait qu'il y avait un 29 février 2010, ce qui bloqua le fonctionnement des jeux en ligne toute la journée du 1 mars 2010
- ▶ iPhone 4 : problème de réveil
 - ▶ après le passage à l'heure d'hiver à l'automne 2010
 - ▶ après le passage à 2011
- ▶ serveurs de Juniper du monde entier (7 nov. 2011)
 - ▶ plantent 6 minutes avec un effet cascade
 - ▶ bug révélé par une mise à jour

Erreurs non révélées

- ▶ Beaucoup de défaillances ne sont pas reconnues
- ▶ D'autres sont masquées
 - ▶ Produit lancé, trop coûteux à rappeler

Facteurs de qualité

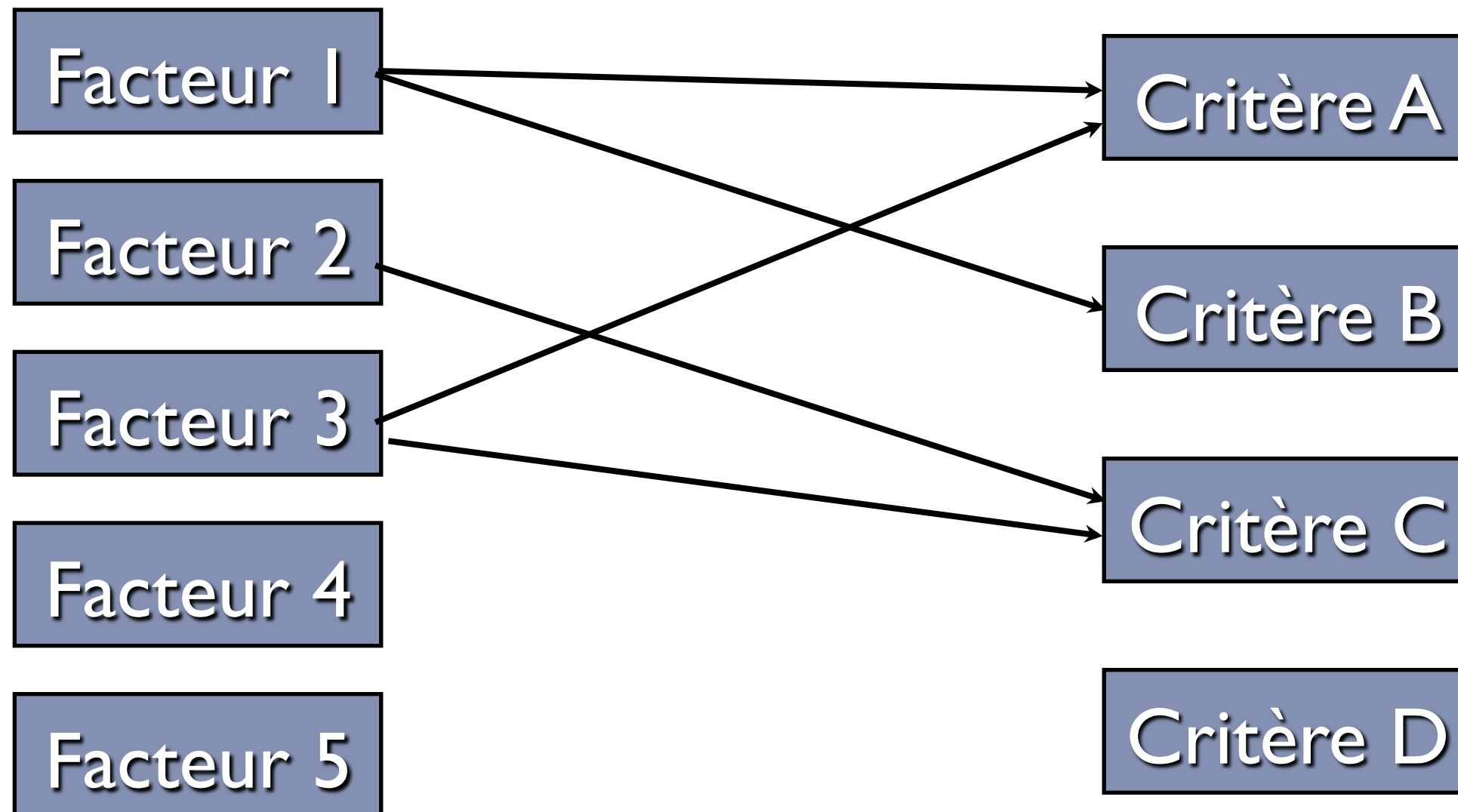
Modèles de qualité logicielle

- ▶ Pour savoir si la qualité s'est améliorée, il faut la mesurer.
- ▶ Et comment mesurer la qualité ?
- ▶ Nécessité d'utiliser des modèles prédictifs de qualité.

Modèle de qualité

- ▶ Ensemble de vues d'un produit.
- ▶ Chaque vue est décomposée en plusieurs facteurs, généralement des attributs externes de qualité.
- ▶ Un facteur est décomposé en plusieurs critères.
- ▶ Chaque critère est défini par un ensemble de métriques.

Facteurs et critères

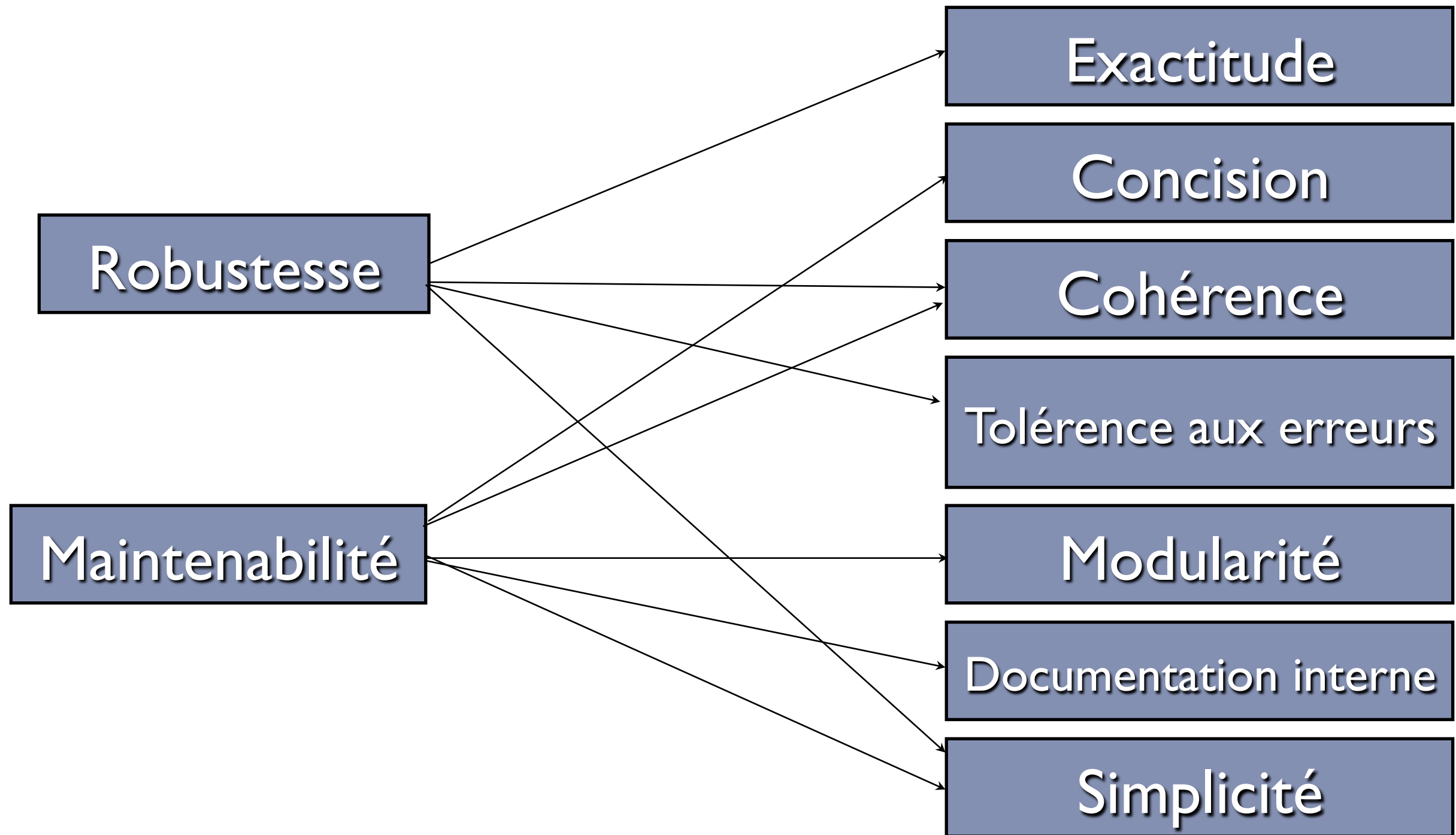


Facteurs de qualité [McCall 1977]

Correctness	Conformité
Reliability	Robustesse
Efficiency	Efficacité
Usability	Maniabilité
Integrity	Sécurité
Maintenability	Maintenabilité

Flexibility	Adaptabilité
Testability	Testabilité
Portability	Portabilité
Reusability	Réutilisabilité
Interoperability	Interopérabilité

Le modèle de McCall



Métriques

Classes de métriques

- ▶ **Produit** : nombre d'erreurs, temps moyen entre défaillances, projection et croissance de la robustesse, complexité.
- ▶ **Processus** : nombre de revues et d'inspections, capacité à estimer, mesures liées à la documentation, nombre de tests.

Classes de métriques

- ▶ Code : lignes de code, complexité cyclomatique, taux de commentaires, taux d'assertions.
- ▶ Spécification : cohésion et couplage des modules, taille et fréquence de communication de données.

Modèle de McCall

Métriques

- ▶ Environ 40 métriques.
- ▶ Mesurer un facteur revient à considérer une liste de conditions à vérifier .
- ▶ La liste de conditions peut s'appliquer aux besoins (R), à la conception (D) et à l'implémentation (I)
- ▶ Une condition est vraie ou fausse.

Métriques des logiciels à objets

Chidamber 96

- ▶ WMC Weighted Methods Per Class
- ▶ DIT Depth of Inheritance Tree
- ▶ NOC Number of Children
- ▶ CBO Coupling between Object Classes
- ▶ RFC Response for a Class
- ▶ LCOM Lack of Cohesion of Methods