

# IFT3913

## Qualité du logiciel et métriques

### Chapitre 2

#### Qualité du produit logiciel

#### Plan du cours

- Introduction
- Qualité du logiciel
- Théorie de la mesure
- Mesure de la qualité du logiciel
- Études empiriques
- Mesure du produit logiciel
- Collection et analyse des métriques
- Test du logiciel



## Pourquoi évaluer la qualité d'un logiciel?

### ■ Situations

- Achat d'un logiciel (alternatives)
- Évaluation de logiciel en cours d'utilisation
- Contrôle de qualité durant le développement d'un logiciel

### ■ Questions

- Le logiciel arrive-t-il à faire ce qui lui est demandé?
- Dans quelle mesure y arrive-t-il?

3




## Pourquoi évaluer la qualité d'un logiciel ? (suite)

### ■ Dimensions

- La qualité n'est pas une notion unidimensionnelle (voir définitions du chapitre 1)
- Il est donc nécessaire
  - de définir les caractéristiques à évaluer (quoi)
  - de déterminer les techniques à utiliser pour évaluer chacune des caractéristiques (comment)

4



## Le « quoi » ou la définition des caractéristiques de qualité

- ISO/IEC 9126 propose 6 caractéristiques de qualité du produit logiciel
  - Capacité fonctionnelle (*Functionality*)
  - Fiabilité (*Reliability*)
  - Facilité d'utilisation (*Usability*)
  - Rendement (*Efficiency*)
  - Maintenabilité (*Maintainability*)
  - Portabilité (*Portability*)

5



## Capacité fonctionnelle

- Définition
  - Ensemble d'attributs portant sur l'existence d'un ensemble de fonctions et leurs propriétés. Les fonctions sont celles qui satisfont aux besoins exprimés ou implicites
- Sous-caractéristiques
  - Aptitude : présence et adéquation d'une série de fonctions pour des tâches données
  - Exactitude : fourniture de résultats ou d'effets justes ou convenus
  - Interopérabilité : capacité à interagir avec des systèmes donnés
  - Sécurité : aptitude à empêcher tout accès non autorisé (accidentel ou délibéré) aux programmes et données

6



## Fiabilité

- Définition
  - Ensemble d'attributs portant sur l'aptitude du logiciel à maintenir son niveau de service dans des conditions précises et pendant une période déterminée
- Sous-caractéristiques
  - Maturité : fréquence des défaillances dues à des défauts du logiciel
  - Tolérance aux fautes : aptitude à maintenir un niveau de service donné en cas de défaut du logiciel ou de violation de son interface
  - Possibilité de récupération : capacité à rétablir son niveau de service et de restaurer les informations directement affectées en cas de défaillance; temps et effort nécessaire pour le faire

7



## Facilité d'utilisation

- Définition
  - Ensemble d'attributs portant sur l'effort nécessaire pour l'utilisation et l'évaluation individuelle de cette utilisation par un ensemble défini ou implicite d'utilisateurs
- Sous-caractéristiques
  - Facilité de compréhension : effort que doit faire l'utilisateur pour reconnaître la logique et sa mise en œuvre
  - Facilité d'apprentissage : effort que doit faire l'utilisateur pour apprendre son application
  - Facilité d'exploitation : effort que doit faire l'utilisateur pour exploiter et contrôler l'exploitation de son application

8



## Rendement

### ■ Définition

- Ensemble d'attributs portant sur le rapport existant entre le niveau de service d'un logiciel et la quantité de ressources utilisées, dans des conditions déterminées

### ■ Sous-caractéristiques

- Comportement vis-à-vis du temps : temps de réponses et de traitement; débits lors de l'exécution de sa fonction
- Comportement vis-à-vis des ressources : quantité de ressources utilisées; durée de leur utilisation lorsqu'il exécute sa fonction

9



## Maintenabilité

### ■ Définition

- Ensemble d'attributs portant sur l'effort nécessaire pour faire des modifications données

### ■ Sous-caractéristiques

- Facilité d'analyse : effort nécessaire pour diagnostiquer les déficiences et causes de défaillance ou pour identifier les parties à modifier
- Facilité de modification : effort nécessaire pour modifier, remédier aux défauts ou changer d'environnement
- Stabilité : risque des effets inattendus des modifications
- Facilité de test : effort nécessaire pour valider le logiciel modifié


10



## Portabilité

- Définition
  - Ensemble d'attributs portant sur l'aptitude du logiciel à être transféré d'un environnement à l'autre
- Sous-caractéristiques
  - Facilité d'adaptation : possibilité d'adaptation à différents environnements donnés sans que l'on ait recours à d'autres actions ou moyens que ceux prévus à cet effet pour le logiciel considéré
  - Facilité d'installation : effort nécessaire pour installer le logiciel dans un environnement donné
  - Conformité aux règles de portabilité : conformité aux normes et aux conventions ayant trait à la portabilité
  - Interchangeabilité : possibilité et effort d'utilisation du logiciel à la place d'un autre logiciel donné dans le même environnement

11



## Le « comment » ou la définition du processus d'évaluation

- ISO/IEC 9126 propose également un ensemble de grandes lignes pour un processus d'évaluation de la qualité
- ISO/IEC 14598 propose un cadre plus précis pour l'évaluation du produit logiciel
- Le projet SCOPE définit un cadre complet pour l'évaluation

12



## Processus d'évaluation (9126)

- Processus d'évaluation est composé de trois étapes
  1. La définition des exigences de qualité
    - L'objectif de cette première étape est de spécifier les exigences en termes de caractéristiques de qualité. Ces exigences peuvent varier d'un composant du produit à un autre
  2. La préparation de l'évaluation.
    - À ce niveau, l'objectif est d'initier l'évaluation et de mettre au point ses bases. Ceci est fait en trois sous-étapes

13



## Processus d'évaluation (9126)

2. La préparation de l'évaluation (suite)
  - Sélection des métriques de qualité
    - Ces dernières doivent correspondre aux caractéristiques énumérées plus haut
  - Définition des taux de satisfaction
    - Les échelles de valeurs doivent être divisées en portions correspondant aux niveaux de satisfaction des exigences
  - Définition des critères d'appréciation
    - Ceci inclut la préparation de la procédure de compilation des résultats par caractéristique. Il est possible aussi de prendre en compte dans cette procédure des aspects de gestion tels que le temps ou les coûts

14



## Processus d'évaluation (9126)

### 3. La procédure de l'évaluation.

- À ce stade, l'évaluation est faite en termes de
  - Mesure. Les métriques sélectionnées sont appliquées au produit, donnant ainsi des valeurs.
  - Notation. Pour chaque valeur mesurée, une note (de satisfaction) est attribuée.
  - Appréciation. En utilisant les critères d'appréciation, un résultat global de l'évaluation du produit est obtenu. Ce résultat est confronté aux aspects de gestion (temps et coûts) pour la prise de décision.

15



## Directives complémentaires (14598)

- L'objectif de cette norme est de fournir
  - les directives d'identification, d'implantation et d'analyse des métriques nécessaires au processus d'évaluation du produit final,
  - les directives de définition des indicateurs qui permettent des évaluations partielles pendant le cycle de développement
- Cette norme donne entre autres
  - les informations générales sur les indicateurs de qualité des logiciels;

16





## Directives complémentaires (14598)

- les critères de sélection de ces indicateurs;
- les directions pour l'évaluation des données de mesurage;
- les directions pour l'amélioration du processus de mesurage;
- des exemples de types de graphes d'indicateurs;
- des exemples d'indicateurs qui peuvent être utilisés pour les caractéristiques de qualité de la norme ISO/IEC 9126

17



## Un exemple de cadre d'évaluation SCOPE

- SCOPE (*Software CertificatiOn Programme in Europe*) est un projet européen ESPRIT
- Objectifs
  - Définir les procédures d'attribution d'un label de qualité à un logiciel quand celui-ci satisfait un certain ensemble d'attributs de qualité
  - Développer des technologies nouvelles et efficaces d'évaluation, à des coûts raisonnables, permettant l'attribution de ce label
  - Promouvoir l'utilisation des technologies modernes de l'ingénierie des logiciels. Celles-ci, étant utilisées durant le développement des logiciels, contribuent à l'attribution de ce label

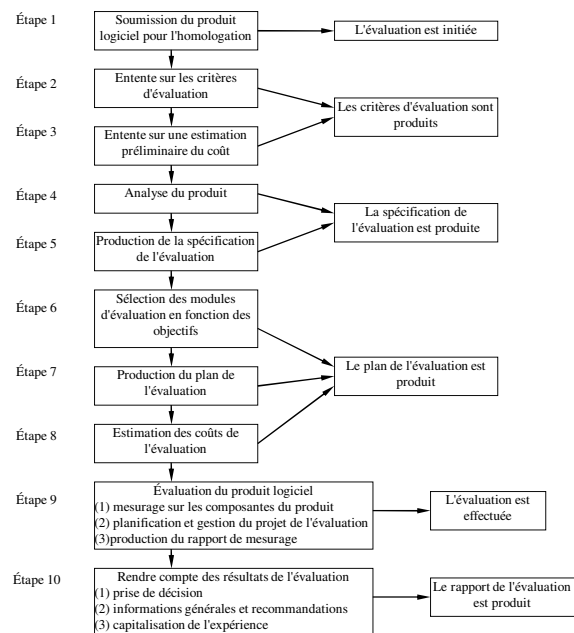
18

## Un exemple de cadre d'évaluation SCOPE

- Résultat : définition d'un cadre d'évaluation comprenant
  - un processus
  - une méthode
  - des techniques

19

## Processus SCOPE



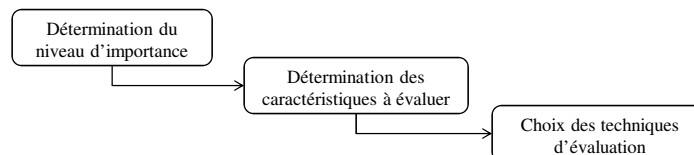
20

## Processus SCOPE

- Documents produits
  - Les critères d'évaluation
  - La spécification de l'évaluation
  - Le plan de l'évaluation
  - Le rapport d'évaluation

21

## Méthode SCOPE



- La méthode d'évaluation s'appuie sur trois types d'analyse techniques
  - L'analyse statique qui consiste à examiner le code pour évaluer les caractéristiques de qualité.
  - L'analyse dynamique qui consiste entre autres à simuler le déroulement de l'application pour effectuer des mesures.
  - L'inspection qui concerne particulièrement les interfaces « utilisateur ».

22

## Méthode SCOPE

- L'évaluation peut se faire selon le niveau d'importance

Niv.	Environnement	Personnes	Économie	Application
D	petit dommage à la propriété	pas de risques pour les personnes	perte économique négligeable	loisirs, domestiques
C	dommage à la propriété	peu de personnes touchées	perte économique significative	alarmes de feu, contrôle de processus
B	dommage environnemental réparable	menace pour des vies humaines	grande perte économique	systèmes médicaux, systèmes financiers
A	dommage environnemental irréparable	des personnes mortes	désastre financier	systèmes de transport, systèmes du nucléaire

23

## Techniques SCOPE

- Choix des techniques pour chaque niveau

	Niveau D	Niveau C	Niveau B	Niveau A
<b>Capacité fonctionnelle</b>	test fonctionnel (boîte noire)	+ inspection des documents (listes de contrôle)	+ test des composantes	+ preuve formelle
<b>Fiabilité</b>	facilités des langages de programmation	+ analyse de la tolérance aux fautes	+ modèle de croissance de la fiabilité	+ preuve formelle
<b>Facilité d'utilisation</b>	inspection des interfaces utilisateurs	+ conformité aux normes sur les interfaces	+ test en laboratoire	+ modèle mental de l'utilisateur
<b>Rendement</b>	mesurage du temps d'exécution	+ test avec bancs d'essai ( <i>benchmarks</i> )	+ complexité algorithmique	+ analyse des performances
<b>Maintenabilité</b>	inspection des documents (listes de contrôle)	+ analyse statique	+ analyse du processus de développement	+ évaluation de la traçabilité
<b>Portabilité</b>	analyse de l'installation	+ conformité avec les règles de programmation	+ évaluation des contraintes de l'environnement	+ évaluation de la conception des programmes

24