

CHAPITRE 1

INTRODUCTION A LA MESURE

DE LOGICIEL

PRÉSENTÉ PAR:

DR BOUNOUR NORA

21/02/2017

Introduction

- Le personnel de génie logiciel est confronté au défi de:
 - Gérer
 - Contrôler
 - Et Effectuer les changements nécessaires pour permettre au logiciel d'évoluer.
- Les ressources nécessaires pour faire ceci efficacement sont coûteuses. Un nombre de manière ont été suggérées pour garder ces coûts bas.

Introduction

- Le personnel doit viser les objectifs suivants:
 - Développement de logiciel compréhensible, maintenable et facile à vérifier et valider à travers le cycle de vie.
 - Le logiciel compréhensible aide à achever un ensemble de qualité telles que Evolvability et Verifiability.

Introduction

Nécessité de Mesurer

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

Introduction Problématique

1. Le logiciel arrive-t-il à faire ce qui lui est demandé ?
Dans quelle mesure y arrive-t-il ?
2. Quel est niveau de compréhensibilité, maintenabilité et evolvabilité, vérifiabilité ou autres qu'un produit logiciel doit manifester afin qu'il soit considéré comme acceptable?

Pourquoi Mesurer

- **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

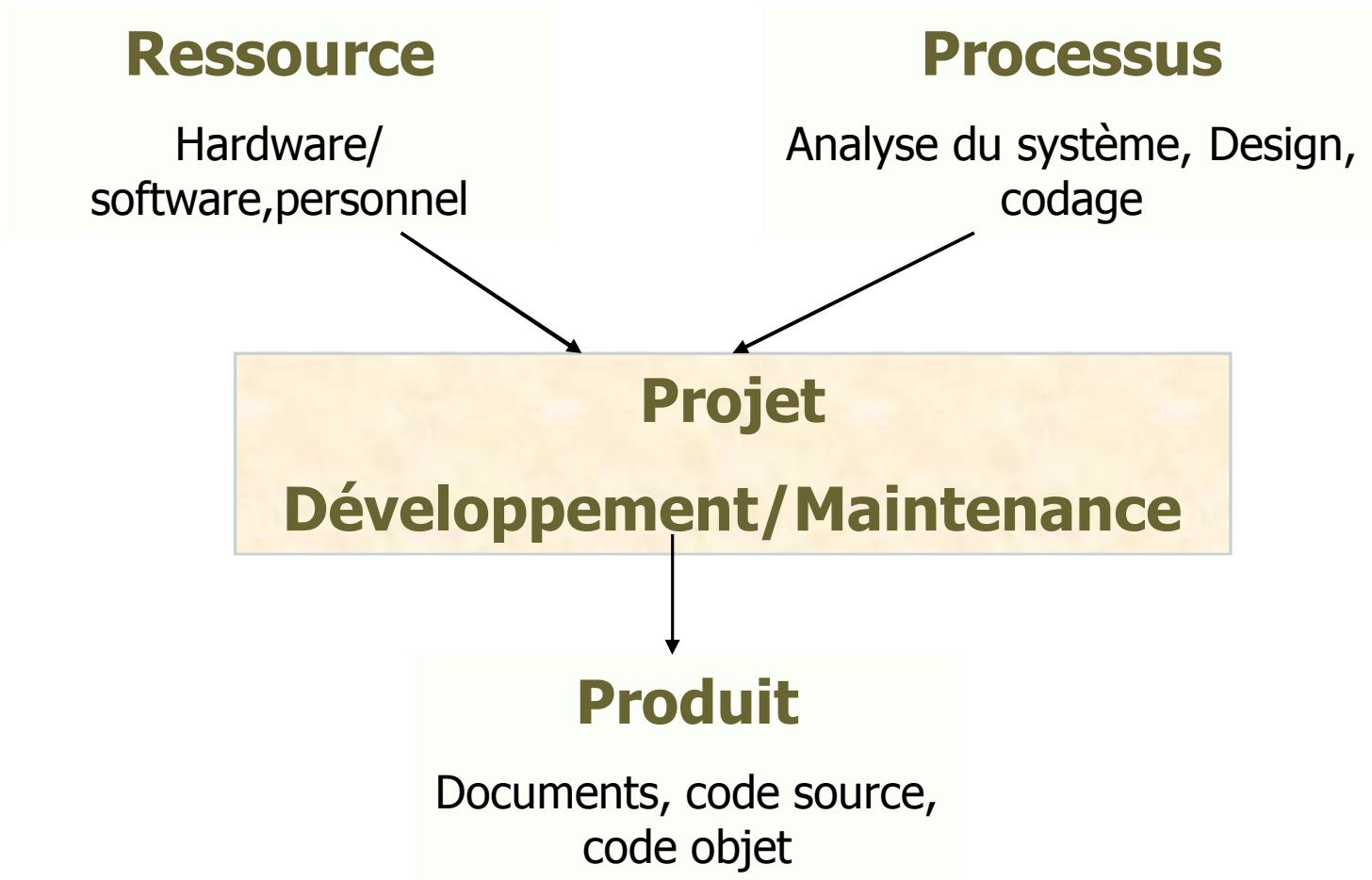
Introduction Problématique

- Pour répondre à ces questions, une bonne compréhension des issues concernées par les mesures est nécessaire sur les plans suivants :
 - Produits logiciels
 - Processus
 - Ressources nécessaires

Mesure de Logiciel

- Généralement, il existe trois entités relatives au logiciel et leurs attributs sont sujets de mesure:
 - Processus
 - Ressource
 - Produit

Relation entre ressource, processus et produit



Mesure de Logiciel

o Processus

- Toute activité reliée à un logiciel telle que Analyse de changement, spécification, conception codage et test.

o Ressource

- Une entrée au processus, par exemple personnel, hardware, et logiciel.

o Produit

- C'est un résultat intermédiaire ou final provenant d'un processus logiciel tel que la documentation du système, données de test, le code source et le code objet.

Mesures de logiciel

- Chaque entité des trois classes produits, processus et ressources possède des attributs.
- Une mesure peut être vue comme un **mapping** d'une **entité** (exemple, code source) en un **attribut** (par exemple, la complexité).

Processus de Mesure de Logiciel

- Processus quantifiant objectivement et empiriquement un attribut du système logiciel .

Aspects mesurables

- Deux types d'attributs peuvent être identifiées :
 - Attributs internes
 - Attributs externes

Aspects mesurables

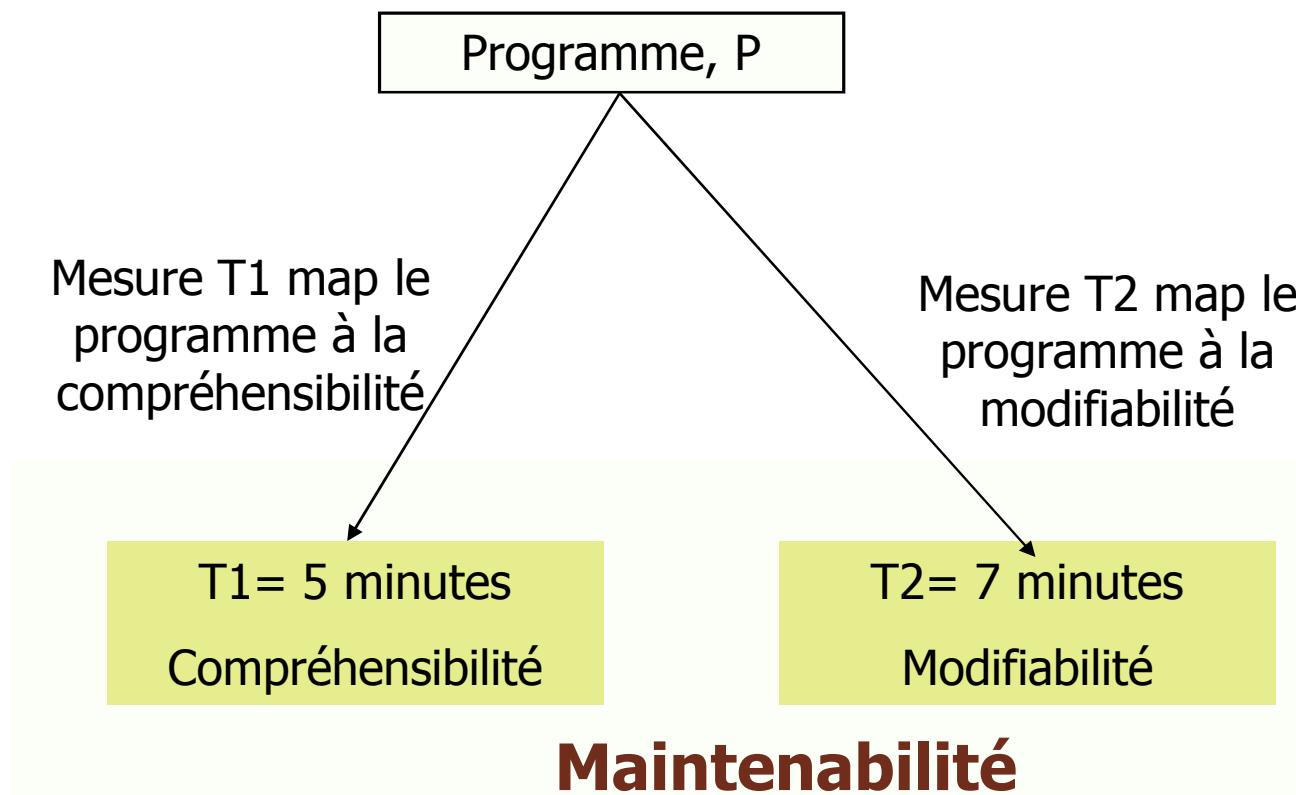
- Des attributs internes : attributs mesurables sur l'entité indépendamment de son environnement.
 - Ces attributs peuvent être mesurer en terme de processus produit, ou ressource.
Exemple: Complexité, modularité.
- Des attributs externes : attributs mesurables par rapport aux liens avec son environnement.
 - Ces attributs peuvent être mesurer seulement avec le respect de la relation du processus , produit ou ressource à son environnement.
 - **Exemple:** maintenabilité du code source d'un programme

Aspects mesurables

- Attributs internes de processus : effort ou durée du processus ou d'une activité, ...
- Attributs internes de produit : taille, complexité, couplage, cohésion, ...
- Attributs internes de ressource : personnel, matériels, méthodes, ...
- Attributs externes de produit : l'efficacité, la portabilité, la facilité de compréhension, ...

EXEMPLE

- Une entité peut avoir plus d'une mesure pour le même attribut; en d'autre terme plus d'un mapping.



Aspects mesurables

○ Quelques remarques

- Les attributs internes de produits sont souvent utilisés pour prédire les attributs externes
- Ces prédictions permettent de contrôler le développement
- Il est très difficile de définir objectivement des mesures qui dépendent de beaucoup d'autres mesures

Aspects mesurables

- Attributs internes de processus : effort ou durée du processus ou d'une activité, ...
- Attributs internes de produit : taille, complexité, couplage, cohésion, ...
- Attributs internes de ressource : personnel, matériels, méthodes, ...
- Attributs externes de produit : l'efficacité, la portabilité, la facilité de compréhension, ...

Aspects mesurables

- Attributs internes de processus : effort ou durée du processus ou d'une activité, ...
- Attributs internes de produit : taille, complexité, couplage, cohésion, ...
- Attributs internes de ressource : personnel, matériels, méthodes, ...
- Attributs externes de produit : l'efficacité, la portabilité, la facilité de compréhension, ...

Comment évaluer la qualité d'un logiciel ?

- La qualité n'est pas une notion Unidimensionnelle
- Plusieurs Dimensions
 - Il est donc nécessaire de définir quelles caractéristiques évaluer (**quoi**)
 - de décider quelles techniques utiliser pour évaluer chacune des caractéristiques (**comment**)

COMMENT ?



OBJECTIF

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*)

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

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

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 oeuvre
- 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

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

MAINTENABILITÉ

○ Définition

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

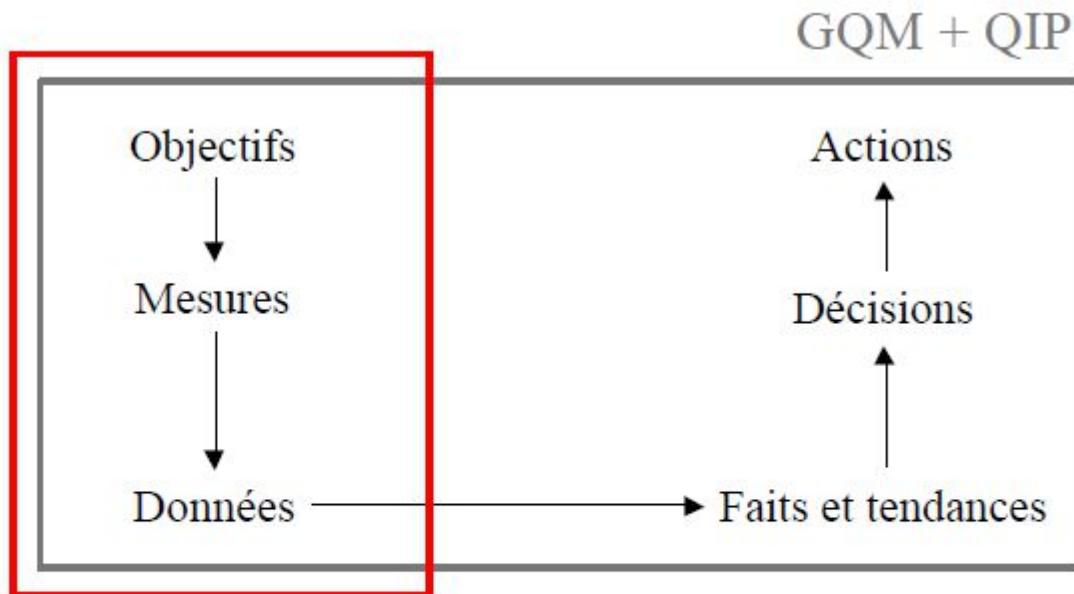
○ Sous-caractéristiques

- 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

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.

Comment ?



ISO9126 + ISO14589

ou

SCOPE

Comment ?

- ISO/IEC 9126 propose également des 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
- La méthode GQM permet de choisir les indicateurs adéquats

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.

PROCESSUS D'ÉVALUATION (9126)

2. La préparation de l'évaluation (suite)

- la sélection des métriques de qualité. Ces dernières doivent correspondre aux caractéristiques énumérées plus haut.
- La définition des taux de satisfaction. Les échelles de valeurs doivent être divisées en portions correspondant aux niveaux de satisfaction des exigences
- La 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.

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

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 ;

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

UN CADRE D'ÉVALUATION, SCOPE

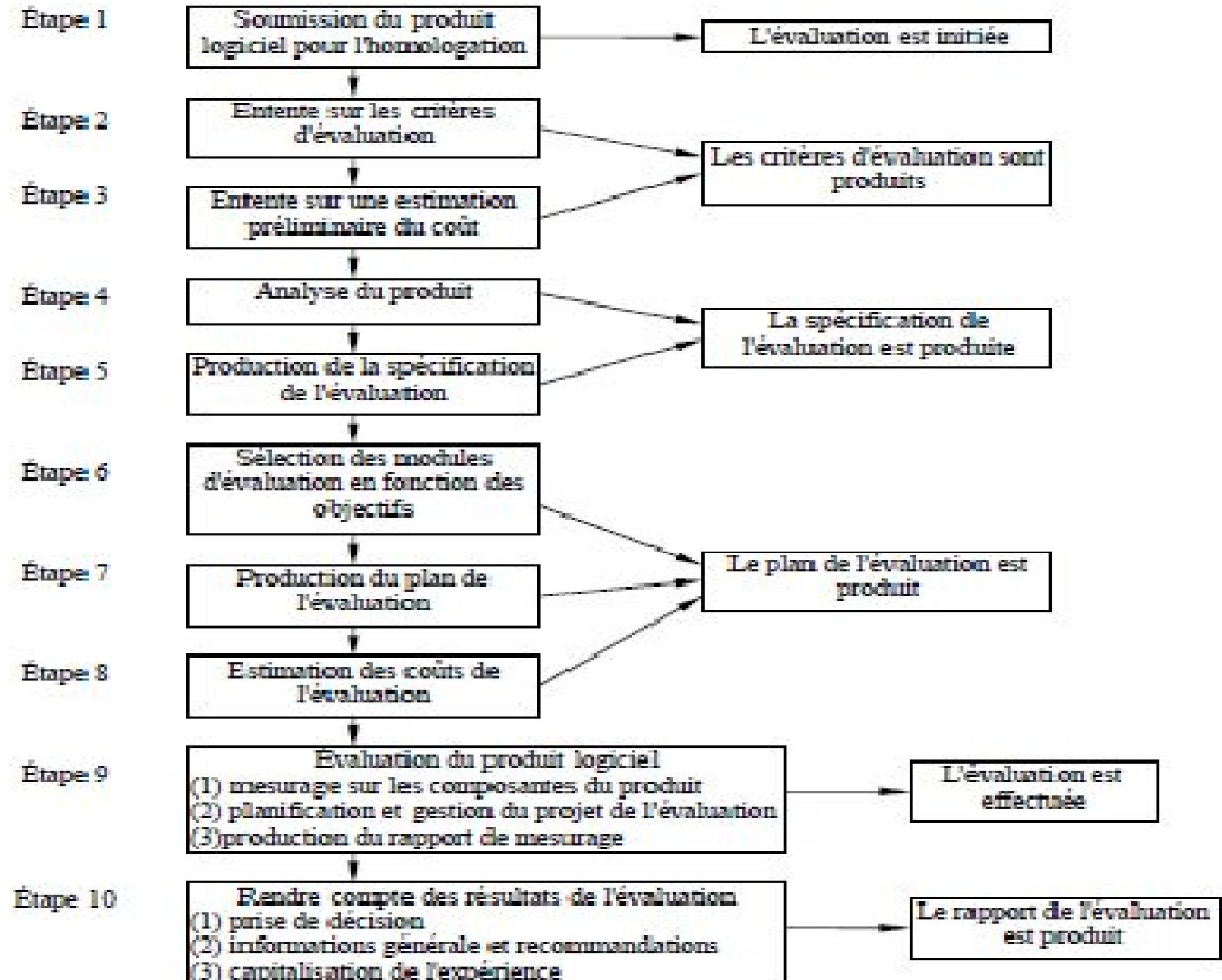
- Résultat : définition d'un cadre d'évaluation comprenant
 - Un processus
 - Une méthode
 - Des techniques

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



MODÈLE SCOPE

Documents produits

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

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".

Méthode SCOPE □

- L'évaluation peut se faire selon le niveau de détail

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

TECHNIQUES SCOPE



○ Choix des techniques pour chaque niveau

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