

# Projet de thèse

## Mesure de la complexité fonctionnelle des logiciels

AUTOMNE 2003

Étudiant: DE TRAN-CAO

Directeurs de recherche:

➤ GHISLAIN LÉVESQUE

➤ JEAN-GUY MEUNIER

## Plan (45 min.)

- Introduction
- Revue de la littérature
  - Mesure du logiciel: concepts, problèmes, développement d'une méthode de mesure ...
  - Mesure de la complexité du logiciel: aspects de la complexité, classifications, complexité cognitive, mesure de la complexité.
  - Approches cognitives: modèles de la capacité humaine, modèle de la compréhension du programme, la complexité de tâche, modèle de la complexité de tâche de Wood.
- Projet de recherche
  - Sujet
  - Motivation
  - Problématiques et approches
  - Méthodologies et la démarche du projet
- Conclusion

## Introduction

- Titre: *Mesure de la complexité fonctionnelle des logiciels*
  - Entité: logiciel = ensemble de fonctionnalités
  - Attribut: complexité
- Complexité: le degré de difficulté dérivée de la fonctionnalité du logiciel.
- Mesure de la complexité: une valeur numérique dérivée de fonctionnalité du logiciel qui indique le degré de difficulté. Elle est un indicateur de l'effort pour réaliser une tâche sur le logiciel.
- ↳ Mesure de la complexité est un indicateur de l'effort de développement du logiciel. Elle doit être en corrélation avec l'effort de développement. Elle pourrait être utilisée pour estimer tôt l'effort de développement/le coût du logiciel.

3

## Revue de la littérature

- Mesure du logiciel
- Développement d'une mesure
- Mesure de la complexité du logiciel
- Approches cognitives

## Mesure du logiciel

- Une mesure du logiciel: méthode de déterminer quantitativement l'ampleur avec lequel un processus, un produit ou un projet de logiciel possède un certain attribut (Fenton, 1997).
- Une mesure doit être définie:
  - Entité
  - Attribut
- Classification
  - Entité: Processus, produit, ressource
  - Attribut: interne, externe
  - Mesure: direct, indirect.

5

## Mesure du logiciel

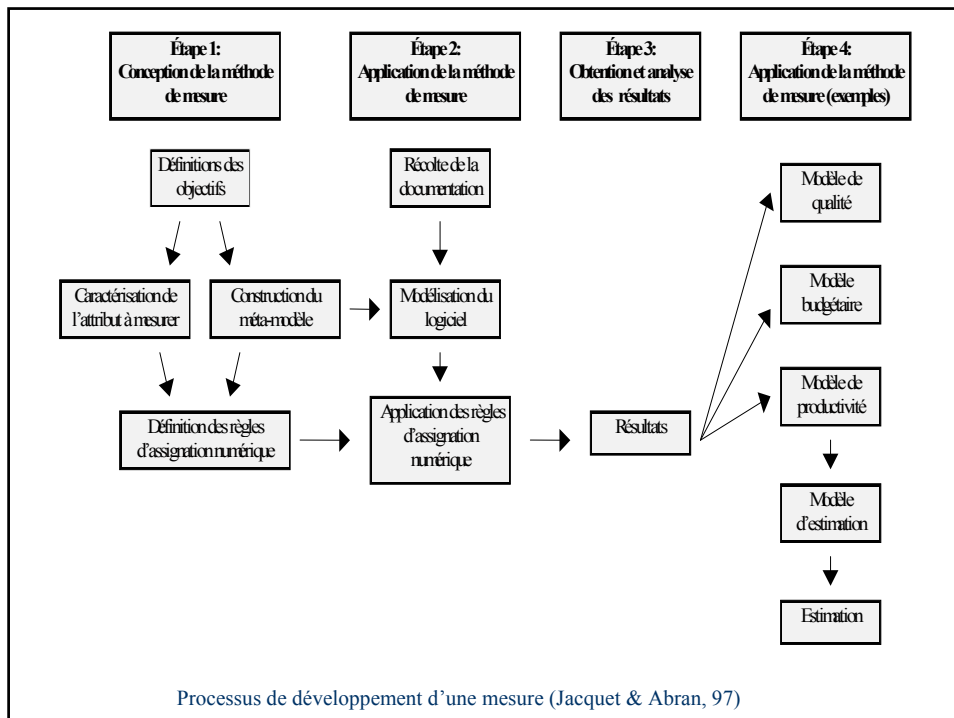
- Rôles de mesure
  - Comprendre l'état actuel du processus de production
  - Contrôler les activités du projet
  - Améliorer le processus de production ainsi que la qualité de produit.
- Problèmes
  - Logiciel n'est pas tangible
  - L'attribut mesuré (e.g la complexité) n'est pas bien compris.
  - Le monde empirique n'est pas clair / bien compris
  - Comment valider une mesure?

6

## Mesure du logiciel

- Approches
  - Axiomes (Weyuker, 1988)
  - GQM (goal – question – metric, Basili, 1984)
- Processus de développement d'une mesure (Jacquet & Abran, 1997)
  - **Étape 1:** Construction de la méthode de mesure
  - **Étape 2:** Application de la méthode
  - **Étape 3:** Obtention des résultats
  - **Étape 4:** Exploitation des résultats

7



## Mesure de la complexité du logiciel

- Interprétations différentes
  - Taille:
    - Longueur du code source (LOC, KLOC)
    - Taille fonctionnelle (FPA, Feature Point, MARK II, COSMIC-FFP etc.)
  - Complexité structurelle:
    - Structure du programme: Nombre cyclomatique, McCabe, 76
    - Structure du design: couplage et cohésion, Yourdon, 79; Fan-in & fan-out, Henry et al. 1981
    - Structure de donnée: software science, Halstead, 77.
- Pas de consensus sur “qu’est-ce que la complexité du logiciel?”

9

## Mesure de la complexité du logiciel

### Classification

- *Complexité du problème*: la difficulté inhérente du problème, par exemple, l'espace du problème des échecs.
- *Complexité algorithmique*: l'ordre de grandeur du temps ou de l'espace de mémoire requis pour exécuter une implémentation.
- *Complexité structurelle*: la difficulté de la structure du logiciel, par exemple la complexité du design.
- *Complexité cognitive*: la difficulté de comprendre le logiciel ou l'effort requis pour réaliser une tâche sur le logiciel.

10

## Mesure de la complexité du logiciel

### Définitions

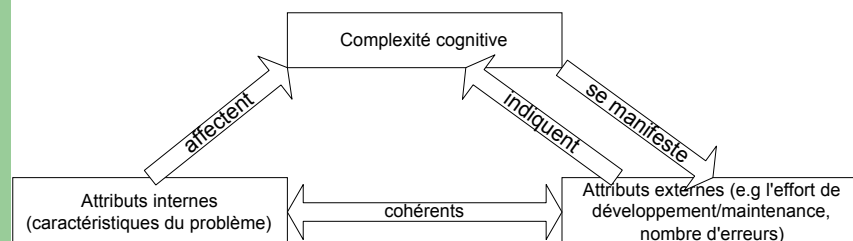
- Software complexity is a characteristic of the software interface which influences the resources another system will expend while interacting with the software (Curtis, 79)
- Software complexity is as a measure of resources expended by a system [human or other] while interacting with a piece of software (Basili, 81).
- The cognitive complexity of software refers to those characteristics of software that affect the level of resources used by a person performing a given task on it (Henderson-Seller, 96).

- ☛ quantifier les caractéristiques du logiciel qui affecte le niveau d'effort requis pour réaliser une certaine tâche sur le logiciel.
- ☛ mesure de la complexité doit être en corrélation avec l'effort de réaliser une tâche sur le logiciel.

11

## Mesure de la complexité du logiciel

### Conception



12

## Recherche sur la complexité du logiciel: l'approche cognitive

- Comprendre le processus de la compréhension et les limites de la capacité humaine pour comprendre la complexité.
- Modèles cognitifs de la complexité
  - Limite de capacité humaine
    - Modèle mémoire court-terme, long-terme
    - Mémoire court-terme:  $7 \pm 2$  (Miller, 56)
    - Durée de mémorisation: 20-30s  $\Rightarrow$  densité de l'information est élevé, on perd de l'information.
  - Processus de compréhension du code: *chunking – tracing*
    - *Chunk* de code: pièce de code ayant une sémantique unitaire qui peut être représentée par un symbole.
    - Composer le programme: programmeur forme les *chunks* et relations entre les *chunks* (Chunking)
    - Débogage (*debug*)/maintenance: programmeur reforme les *chunks* et relations entre les *chunks* (Tracing)
    - la complexité est mesurée en se basant sur la relation entre les chunks: McCabe, 76, Henry 81.
- Traitement de l'information: Entropy (David, 88), soft. Science (Halstead,77)

13

## La Complexité de tâche

- La complexité de tâche  $\leftrightarrow$  la difficulté de réaliser une tâche.
- Deux approches:
  - Psychologique: la complexité est définie en termes de la capacité de l'individu.
  - Objective: la complexité est définie sur des caractéristiques de la tâche.
    - ✦ Le nombre d'indices d'information (*information cues*)
    - ✦ Inter-relations entre les indices d'information
    - ✦ Principe de l'intégration des informations

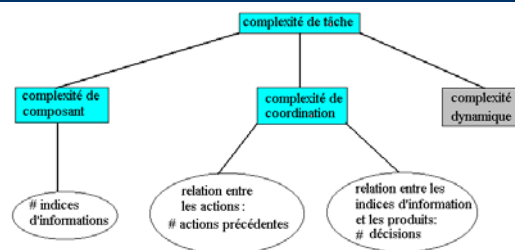
14

## Modèle de la complexité de tâche

- Le modèle de la complexité de tâche de Wood est basé sur le traitement de l'information pendant la réalisation de tâche
  - Réalisation de tâche = traitement de l'information
  - La complexité de tâche = la complexité de traitement de l'information.
- Modèle conceptuel de tâche:
  - Produits (product).
  - Actions requises (required act)
  - Indices d'information (information cues)

15

## Mesure de la complexité de tâche



Complexité de composant:  $\Sigma$  indices d'information.

Complexité de coordination:  $\Sigma$ actions précédentes

$\Sigma$  tournant

Complexité dynamique: changement de la complexité avec le temps

16



## Projet de recherche

- Sujet
- Motivation
- Problématiques & approches
- Méthodologies & la démarche
- Contributions

## Sujet de recherche

- Titre: *Mesure de la complexité fonctionnelle des logiciels*
- Complexité: le degré de difficulté dérivée de la fonctionnalité du logiciel.
- Mesure de la complexité: quantifier les caractéristiques du problème décrivant dans les documents de l'analyse du logiciel (e.g exigences de l'utilisateur, la spécification du système, la conception du système)
- Attribut intéressé: complexité cognitive du logiciel interprétée comme l'effort requis pour développer le logiciel.
- But: mieux estimer l'effort de développement du logiciel.

## Motivation

- Les mesures basées sur 'code' ne permettent pas d'estimer tôt l'effort de développement.
- Les mesures basées sur 'design' font référence à la qualité du design plutôt que l'effort de développement.
- Les mesures basées sur la fonctionnalité permettent d'estimer tôt l'effort, mais
  - Évaluation subjective (e.g FPA)
  - Ne couvrent suffisamment pas la complexité (e.g COSMIC-FFP)
  - N'estiment pas efficacement l'effort (taux d'erreurs est encore élevé)

19

## Motivation

Mesure	N	Corrélation R <sup>2</sup>	PRED(0.25)	MRRE	Par
COSMIC-FFP	15	80%	53%	40%	Abran Alain
COSMIC-FFP	19	75%	53%	51%	Abran Alain
COSMIC-FFP	15	85%	50%	48%	Tran-Cao De
FPA (Kemerer model)	10	N/a	N/a	81%	Tran-Cao De
FPA (Matson model)	10	N/a	N/a	54%	Tran-Cao De
LOC (SLIM)	15	89%	N/a	772%	Srinisavan
LOC (COCOMO)	15	70%	N/a	610%	Srinisavan
FP	15	58%	N/a	103%	Srinisavan

But: chercher des mesures (de la complexité) qui permettent d'améliorer la performance d'estimation d'effort de développement.

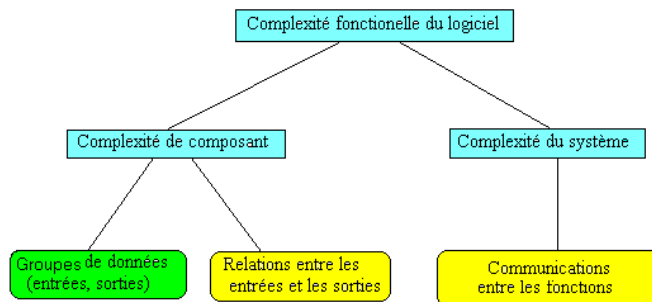
Corrélation R<sup>2</sup> ≥ 75% , PRED (.25) ≥ 60% , MMRE ≤ 40%

20

## Problématiques et approches

### 1 - définition de la complexité, mesure de la complexité

- « Mesure de la complexité du logiciel » est un « misnomer ». Une interprétation de ce terme est la difficulté pour mettre à jour, comprendre et modifier le logiciel (ZUSE, 91)
- Proposition d' un modèle conceptuel (framework)



21

## Problématiques et approches

### 2- la base théorique de la méthode

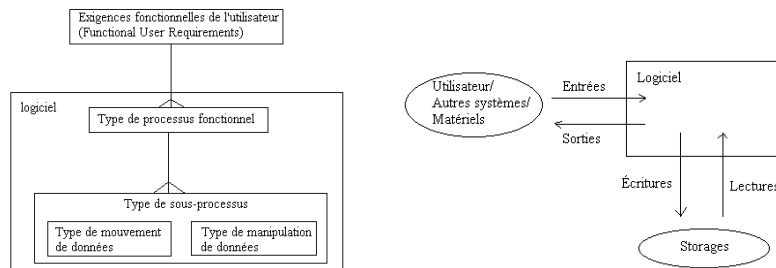
- Logiciel = ensemble de fonctions (processus fonctionnels) qui réalisent une tâche / des tâches.
- La complexité de logiciel = la complexité de tâche = la complexité de processus de traitement de l'information pour la réalisation de tâche.
- La théorie de la complexité de tâche de Wood
  - Vérifier le modèle conceptuel de la complexité fonctionnelle du logiciel.
  - Guider la proposition des mesures.

22

## Problématiques et approches

### 3- Modélisation du logiciel pour mesurer la complexité

- Logiciel n'est pas un produit tangible
- Modèle fonctionnel du logiciel de COSMIC
- Taille fonctionnelle =  $\Sigma(\text{entrées} + \text{sorties} + \text{lectures} + \text{écritures})$ 
  - Ne pas tenir compte de manipulations de données
  - Ne pas tenir compte de relations entre les processus fonctionnels.

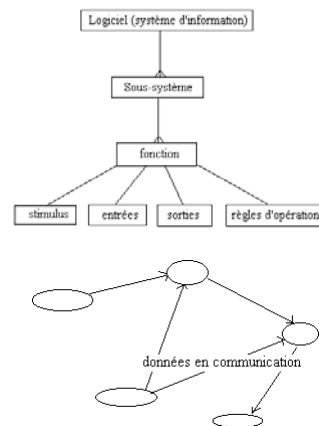


23

## Problématiques et approches

### 3- Modélisation du logiciel pour mesurer la complexité

- Mouvements de données (comme COSMIC-FFP)
- La manipulation de données est modélisée par les règles d'opération
  - Une règle d'opération est une condition / une contrainte sur des entrées pour produire les sorties.
  - Par exemple:  $F1 = a+b$   
 $F2 = |a+b| = a+b, \text{ si } a+b \geq 0$   
 $= -(a+b) \text{ si } a+b < 0.$
- La relation de données entre les fonctions modélisée par les transitions de données ou le partage des données.



24

## Problématiques et approches

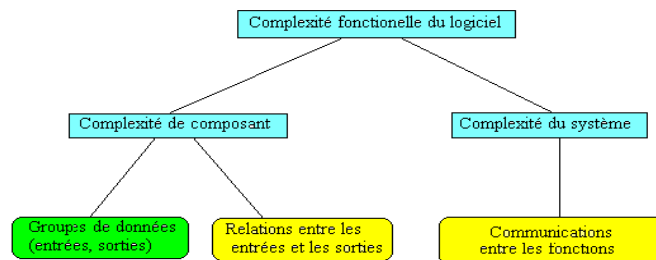
### 4- problème d'évaluation subjective de la complexité

- L'être humain utilise des mesures symboliques comme 'simple', 'complexe', 'très complexe'
- Modèle d'estimation utilise des paramètres numériques
- Une des solutions: assignation des valeurs numériques aux présentations symboliques, par exemple simple=3, complexe=5...
- Complexité est évaluée subjectivement, non répétable.
- Les mesures qui quantifient objectivement des caractéristiques des logiciels sont préférables.

25

## Problématiques et approches

### Mesures proposées



- $\Sigma$ (mouvements de données)  
comme COSMIC-FFP

$\Sigma$ (conditions)

- $\Sigma$ (groupes de données entrées, sorties)

❖ Entropie du système.

❖ Degré d'interconnexion moyen

## Problématiques et approches 5 - validation

- Validation, selon Fenton:
  - La mesure capture l'attribut que l'on veut mesurer, et
  - La mesure quantifie correctement l'attribut mesuré = mesure satisfait la **condition de représentation**.



1,25m 1,70m

Mesure  $\mu$  est valide si et seulement si:

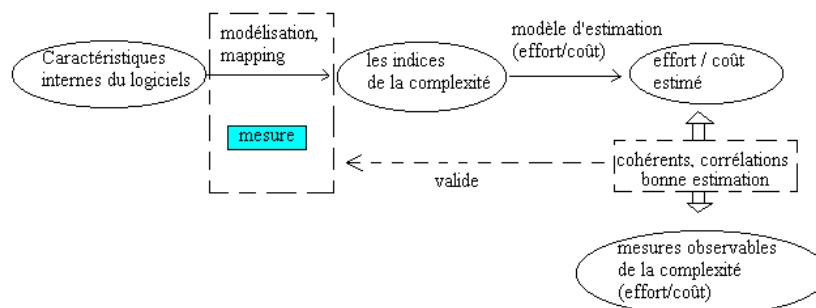
$$A > B \leftrightarrow \mu(A) > \mu(B).$$

❖ Ce type de la validation inspirée de mesures des objets physiques.

❖ Comment valider une mesure de l'intelligence / de la complexité?

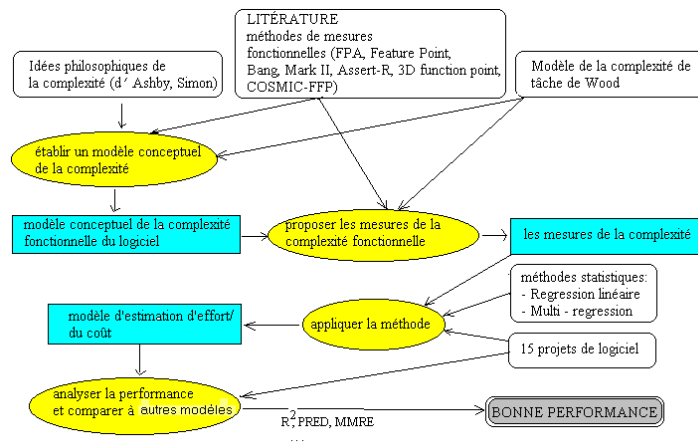
27

## Problématiques et approches 5 - validation



28

## Méthodologies et la démarche du projet



29

## Contribution du projet

- Établir une nouvelle méthode de mesure qui
  - Est basée sur une théorie de la complexité.
  - Offre des indices tôt de l'effort de développement du logiciel
  - Évite des évaluations subjectives
- Introduire la complexité de tâche comme la base théorique de mesure de la complexité fonctionnelle.
- Valider le modèle de la complexité de tâche dans l'environnement de développement du logiciel.
- Améliorer la performance d'estimation d'effort et du coût

30

## Conclusion

- Validation de la méthode: approche pragmatique.
  - Ne peut pas prouver « *measure right thing in the right way* »
  - Prouver que les mesures sont utiles, conformes à des normes IEEE 14143-1.
- Le projet est réalisable.
- Les contributions sont pertinentes

31

## Discussion & Questions