

Université de Yaoundé I  
\*\*\*\*\*  
Faculté des sciences  
\*\*\*\*\*  
Département d'Informatique  
\*\*\*\*\*



The University of Yaoundé I  
\*\*\*\*\*  
Faculty of Sciences  
\*\*\*\*\*  
Department of Computer Science  
\*\*\*\*\*

# **METHODES D'ESTIMATION DES COUTS D'UN PROJET : Modélisation Algorithmique**

Sous l'encadrement de : **Dr ABESOLO**  
*(Enseignant au département  
d'informatique)*

**Année Académique  
2021-2022**

## LISTES DES MEMBRES

Nom(s) et Prénom(s)	Matricule
<b>TCHINDE WAFFO Cyrille-Junior</b>	<b>19M2392</b>
<b>TCHOUAMOU PAYONG Thierry Pavonne</b>	<b>19M2381</b>
<b>TCOUGWEN TEWO Brice Jordan</b>	<b>19M2412</b>
<b>TCHUENTE KAMMOGNE Patrice</b>	<b>162543</b>
<b>TCHUIMBOU KENMEGNE Jefferson</b>	<b>21S2821</b>
<b>TEYOU GHOMFO Martial</b>	<b>19M2364</b>
<b>TIATI A BIOMBI Ruud Dil Nelson</b>	<b>12Y440</b>
<b>TIOMELA TCHINDA Raïssa Flore</b>	<b>19M2380</b>
<b>TOUEZANG AUDRAN Cabral</b>	<b>19M2471</b>
<b>TSENG MEDILI George Ulrich</b>	<b>14Y530</b>
<b>TUEGUEM TIEKAM Melvis</b>	<b>18U2339</b>
<b>Yannick-Joseph MBIDA OBIANG</b>	<b>19M2476</b>
<b>YEMALEU WOTCHEU Jocelyn Parfait</b>	<b>19M2429</b>
<b>YOMKIL Philippe Papin</b>	<b>20V2897</b>
<b>ZANGA FOE Yves Emmanuel</b>	<b>16Y380</b>
<b>ZEKENG Arthur Xavier</b>	<b>18T2911</b>
<b>ZOGO ABOUMA Zozime Achaire</b>	<b>18N2824</b>
<b>BIKIM BI MSOGA Jean Petit Yvelos</b>	<b>19M2596</b>
<b>CHEUNWI Daniel Nganeck</b>	<b>19M2640</b>

## Table des matières

<b>LISTES DES MEMBRES .....</b>	<b>1</b>
<b>INTRODUCTION.....</b>	<b>3</b>
<b>Partie I : MODELISATION ALGORITHMIQUE.....</b>	<b>4</b>
<b>I.    PRINCIPE.....</b>	<b>4</b>
<b>II.    AVANTAGES .....</b>	<b>5</b>
<b>III.    INCONVENIENTS .....</b>	<b>5</b>
<b>Partie II : PRESENTATION DES DIFFERENTS MODELES .....</b>	<b>5</b>
<b>I.    Modèle COCOMO .....</b>	<b>5</b>
<b>1.    Projets organiques .....</b>	<b>5</b>
<b>2.    Projets de type semi-détaché .....</b>	<b>6</b>
<b>3.    Projets de type embarqué .....</b>	<b>6</b>
<b>4.    Modèle COCOMO de base .....</b>	<b>6</b>
<b>5.    Modèle COCOMO intermédiaire .....</b>	<b>7</b>
<b>6.    Modèle COCOMO détaillé .....</b>	<b>9</b>
<b>II.    Présentation des autres modèles .....</b>	<b>9</b>
<b>1.    SEL Model.....</b>	<b>9</b>
<b>2.    Walston-Felix Model .....</b>	<b>10</b>
<b>3.    Bailey-Basil Model.....</b>	<b>10</b>
<b>4.    Halstead Model .....</b>	<b>10</b>
<b>5.    Doty (for KLOC &gt; 9) .....</b>	<b>10</b>
<b>6.    IVR Model.....</b>	<b>11</b>
<b>Partie III : CAS PRATIQUE .....</b>	<b>11</b>

## INTRODUCTION

De nos jours, les logiciels sont devenus indispensables pour les entreprises et organisations. Ils sont utilisés pour effectuer différentes activités relatives à la gestion des opérations, la communication, la simulation, etc... Avec l'évolution des industries et de leurs besoins, les logiciels sont devenus de plus en plus grands et complexes. Cependant, le taux de réussite des projets de développement de ces logiciels n'a pas progressé car ils sont devenus plus difficiles à gérer et à réaliser. La bonne estimation de l'effort, des coûts et des délais est un facteur clé de succès des projets logiciels. Il existe différentes techniques d'estimation de l'effort de développement logiciel. Chaque technique a des spécificités et des caractéristiques relatives aux éléments impliqués tels que les informations sur les projets à estimer et le niveau d'expérience de la personne chargée de l'estimation, etc... Dans cet exercice nous nous attarderons exclusivement sur la modélisation algorithmique. Nous commencerons par donner le principe de la modélisation algorithmique ses avantages, ainsi que ses inconvénients, puis nous présenterons les différents modèles, nous finirons par un cas pratique celui utilisation d'un modèle COCOMO.

## Partie I : MODELISATION ALGORITHMIQUE

### I. PRINCIPE

Les techniques algorithmiques se basent sur des modèles élaborés à partir de bases de données historiques. Ces modèles expriment les relations mathématiques qui existent entre l'effort et les variables discriminantes considérées déterminantes de l'effort et appelées "conducteurs d'effort" ou "facteurs d'effort". Divers modèles d'estimation d'effort ont été proposés par des chercheurs et des praticiens au fur et à mesure de l'évolution du développement logiciel. Ils peuvent être classifiés, selon leur nature et les hypothèses sur lesquelles ils se reposent (normalité des données par exemple), en deux catégories :

- Modèles paramétriques : sont des modèles d'estimation ayant une forme bien définie et un nombre fixe de paramètres. Chaque modèle d'estimation paramétrique s'appuie sur une forme de fonction déterminée à priori et un nombre déterminé à priori de paramètres. Les paramètres des modèles d'estimation paramétriques sont appelés les coefficients. Ces coefficients sont directement déterminés à partir de l'apprentissage sur la base de données. Des exemples courants de modèles paramétriques sont **COCOMO (Constructive Cost Model)**, **Walston-Felix**, et la régression linéaire.
- Modèles non paramétriques : sont des modèles d'estimation ayant une forme non déterminée à priori et un nombre de paramètres non fixé à priori. Les modèles non paramétriques s'appuient sur moins d'hypothèses concernant la distribution des données et supposent que la forme de la fonction n'est pas définie à priori. Dans ces modèles, le nombre de paramètres est flexible et évolue pour s'accorder avec la complexité des données. En effet, ces modèles ont, en plus des coefficients, des hyperparamètres qui sont des variables contrôlant la configuration des modèles.

## **II. AVANTAGES**

Les méthodes algorithmiques ont l'avantage d'être objectives, répétables, ne nécessitant pas d'expérience, interprétable, et surtout avec une bonne puissance prédictive.

## **III. INCONVENIENTS**

Elles nécessitent des bases de données, elles ne sont pas flexibles surtout s'il s'agit des modèles basés sur la théorie et non sur les données.

## **Partie II : PRESENTATION DES DIFFERENTS MODELES**

### **I. Modèle COCOMO**

Barry Boehm a proposé COCOMO (Constructive Cost Model) en 1981. COCOMO est l'un des modèles d'estimation de logiciels les plus utilisés dans le monde. COCOMO prédit les efforts et le calendrier d'un produit logiciel en fonction de la taille du logiciel (nombre de ligne de code).

Les étapes nécessaires à l'élaboration de ce modèle sont :

- Obtenir une première estimation de l'effort de développement à partir de l'évaluation de milliers de lignes de code source livrées (KLOC).
- Déterminer un ensemble de facteurs multiplicateurs à partir de divers attributs du projet.
- Calculer l'estimation de l'effort en multipliant l'estimation initiale avec tous les facteurs multiplicateurs, c'est-à-dire multipliez les valeurs de l'étape 1 et de l'étape 2

Dans COCOMO, les projets sont classés en trois types :

- Organique ou biologique
- Semi-détaché ou Jumelé
- Embarqué

#### **1. Projets organiques**

Un projet de développement peut être traité de type organique, si le projet consiste à développer un programme d'application bien compris, la taille de l'équipe de développement est raisonnablement petite et les membres de l'équipe sont expérimentés dans le développement de méthodes de projets similaires.

Exemples : systèmes commerciaux simples, des systèmes simples de gestion des stocks et des systèmes de traitement de données.

## **2. Projets de type semi-détaché**

Un projet de développement peut être traité de type semi-détaché si le développement consiste en un mélange de personnel expérimenté et inexpérimenté. Les membres de l'équipe peuvent avoir une expérience limitée des systèmes connexes, mais peuvent ne pas être familiarisés avec certains aspects de la commande en cours d'élaboration. Ces projets sont généralement de grande taille.

Exemple : le développement d'un nouveau système d'exploitation (OS), d'un système de gestion de base de données (SGBD) et d'un système de gestion d'inventaire complexe.

## **3. Projets de type embarqué**

Un projet de développement est traité comme étant de type embarqué, si le logiciel en cours de développement est fortement couplé à du matériel complexe, ou si des réglementations strictes sur le mode opératoire existent.

Exemple : ATM, contrôle du trafic aérien.

Pour les trois catégories de produits, Boehm fournit un ensemble d'expressions différent pour prédire l'effort (en unité de personne-mois) et le temps de développement à partir de la taille de l'estimation en KLOC (Kilo Line of code). L'estimation des efforts prend en compte la perte de productivité due aux vacances, repos hebdomadaire, pauses café, etc.

Selon Boehm, l'estimation du coût du logiciel doit se faire en trois étapes :

- Modèle de base
- Modèle intermédiaire
- Modèle détaillé

## **4. Modèle COCOMO de base**

Boehm propose 5 classes de projets

- Les petits projets : 2 KLOC
- Les projets intermédiaires : 8 KLOC
- Les projets moyens : 32 KLOC
- Les grands projets : 128 KLOC

- Les très grands projets : 512 KLOC

Le modèle COCOMO de base fournit une taille précise des paramètres du projet. Les expressions suivantes donnent le modèle d'estimation de base de COCOMO :

$$\text{Effort} = a * (\text{KLOC})^b \text{ H-M}$$

$$\text{Tdev} = c * (\text{efforts})^d \text{ Mois}$$

- KLOC est la taille estimée du produit logiciel indiquée en kilo lignes de code,
- a, b, c et d sont des constantes pour chaque groupe de produits logiciels,
- Tdev est le temps estimé pour développer le logiciel, exprimé en mois, Effort est l'effort total requis pour développer le produit logiciel, exprimé en Homme-mois (H-M).

**Tableau d'estimation de l'effort et du temps de développement d'un projet**

Type de projet	Estimation de l'effort	Estimation de la durée
Organique	$\text{Effort} = 2.4 * (\text{KLOC})^{1.05}$	$\text{Tdev} = 2.5 * (\text{Effort})^{0.38}$
Semi-détaché	$\text{Effort} = 3.0 * (\text{KLOC})^{1.12}$	$\text{Tdev} = 2.5 * (\text{Effort})^{0.35}$
Embarqué	$\text{Effort} = 3.6 * (\text{KLOC})^{1.20}$	$\text{Tdev} = 2.5 * (\text{Effort})^{0.32}$

## 5. Modèle COCOMO intermédiaire

Le modèle de base de COCOMO considère que l'effort est uniquement fonction du nombre de lignes de code et de quelques constantes calculées en fonction des différents systèmes logiciels. Le modèle

COCOMO intermédiaire reconnaît ces faits et affine les estimations initiales obtenues grâce au modèle COCOMO de base en utilisant un ensemble de 15 inducteurs de coûts basés sur divers attributs du génie logiciel. Classification des inducteurs de coûts et de leurs attributs

Attributs du produit

RELY : sureté du produit

DATA : taille de la base de données

CPLX : complexité du produit



#### Attributs de la machine

TIME : contraintes de temps d'exécution

STOR : contraintes de mémoire principale

VIRT : changement dans l'ensemble software + hardware

TURN : temps de retour d'une tâche soumise à l'ordi

#### Attributs du personnel

ACAP : aptitude de l'analyste

AEXP : expérience de ces applications

PCAP : aptitude du programmeur

VEXP : expérience de l'ensemble software + hardware

LEXP : expérience du langage de programmation

#### Attributs du projet

MODP : pratique des techniques modernes de programmation

TOOL : utilisation d'outils

SCED : agenda imposé

#### Pour l'obtenir la valeur de l'effort

- Pour chaque attribut on associe un facteur multiplicateur
- On multiplie ces 15 facteurs
- On détermine l'effort nominal lié au projet

Le tableau suivant nous donne une estimation de l'effort nominale pour chacun des trois types de projets

Type de projet	Estimation de l'effort nominal
Organique	$\text{Effort}_{\text{nominal}} = 3.2 * (\text{KLOC})^{1.05}$
Semi-détaché	$\text{Effort}_{\text{nominal}} = 3.0 * (\text{KLOC})^{1.12}$
Embarqué	$\text{Effort}_{\text{nominal}} = 2.8 * (\text{KLOC})^{1.20}$

- On multiplie le produit de ces 15 facteurs par l'effort nominal  
 $\text{Effort} = \text{ProduitFacteur} * \text{Effort}_{\text{nominal}}$

Les formule des durées de développements restent inchangées.

## 6. Modèle COCOMO détaillé

COCOMO détaillé intègre toutes les qualités de la version standard avec une évaluation de l'effet du générateur de coûts sur chaque méthode du processus d'ingénierie logicielle. Le modèle détaillé utilise divers multiplicateurs d'effort pour chaque propriété d'inducteur de coût. Dans COCOMO détaillé, l'ensemble du logiciel est différencié en plusieurs modules, puis nous appliquons COCOMO dans divers modules pour estimer l'effort puis additionner l'effort.

Les six phases de COCOMO détaillé sont :

1. Planification et exigences
2. Structure du système
3. Structure complète
4. Code et test du module
5. Intégration et test
6. Modèle de coût constructif

L'effort est déterminé en fonction du devis programme, et un ensemble d'inducteurs de coûts sont donnés en fonction de chaque phase du cycle de vie du logiciel.

## II. Présentation des autres modèles

Il existe plusieurs modèles qui ont été développés pour l'estimation de l'effort d'un projet logiciel. Ces modèles ont été élaborés en analysant un grand nombre de projets réalisés par différentes organisations. Ces modèles donnent évidemment des résultats différents selon le type de projet logiciel.

La forme générale de l'effort et de la durée peut alors s'écrire comme :

$$\text{EFFORT} = a(\text{KLOC})^b$$

$$\text{Duration} = c(E)^d$$

Où E est l'Effort, KLOC est la taille mesurée en millier de lignes de codes,  $a$ ,  $b$ ,  $c$  et  $d$  sont des paramètres constants qui sont déterminés par l'analyse de régression et la résolution des équations linéaires en utilisant une collection de résultats de projets.

### 1. SEL Model

Le « Software Engineering Laboratory » (SEL) de l'Université de Maryland a établi un modèle pour estimer l'effort d'un logiciel appelé « SEL Model », qui se définit comme suit :

$$\text{EFFORT} = 1.4 * (\text{Size})^{0.93}$$

$$\text{Duration} = 4.6 * \text{KLOC}^{0.26}$$

## **2. Walston-Felix Model**

En l'an 1977, Walston et Felix ont développé un modèle pour estimer l'effort en considérant 60 projets collectés de la division de « IBM's Federal System ». Ils fournissent également une relation entre les lignes de code source livrées et constituent une participation aux changements orientés vers le client, contraintes mémoires etc. En accord avec le modèle de Walston et Félix, l'effort est calculé comme suit :

$$\text{EFFORT} = 5.2 * \text{KLOC}^{0.91}$$

$$\text{Duration} = 4.1 * \text{KLOC}^{0.36}$$

## **3. Bailey-Basil Model**

Bailey-Basil a développé ce modèle en tenant compte des lignes de code source livrées et formule la relation :

$$\text{EFFORT} = 5.5 * \text{KLOC}^{1.16}$$

## **4. Halstead Model**

Ce modèle fut développé par Halstead entre les lignes de code sources livrées et formule la relation :

$$\text{EFFORT} = 0.7 * \text{KLOC}^{1.50}$$

## **5. Doty (for KLOC > 9)**

En accord avec Doty, la formule pour l'estimation de l'effort logiciel est :

$$\text{EFFORT} = 5.288 * \text{KLOC}^{1.047}$$

## 6. IVR Model

Ce projet est développé pour plusieurs projets d'externalisation de processus d'affaires

$$\text{EFFORT} = 3.4 * (\text{taille projet})^{1.15}$$

$$\text{Tdev} = 2.2 * (\text{Effort})^{0.31}$$

## Partie III : CAS PRATIQUE

Projet se création d'un nouveau système d'exploitation.

Etape d'évaluation du cout du projet

- Déterminer le type du projet : Il s'agit d'un projet de type semi-détaché
- Evaluer la taille du projet : projet de grande taille (environ 320 KLOC)
- Calculer l'effort et la durée de développement du projet :  
 $\text{Effort} = 3.0 * (320)^{1.12} \rightarrow \text{Effort} = 1918 \text{ HM}$   
 $\text{Tdev} = 2.5 * (1918)^{0.35} \rightarrow \text{Tdev} = 35 \text{ Mois (Environ 3ans)}$   
 $\text{Equipe} = \text{Effort}/\text{Tdev} \rightarrow \text{Equipe} = 1918/35 = 54.8 \text{ H (55 personnes)}$

Pour déterminer le cout du projet il suffit de multiplier L'effort par la valeur en monnaie de la paye heuristique des membres de l'équipe.

On a : 1 HM = 152 Heures

$$\text{Cout} = \text{Effort} * \text{PayeParHeure}$$

## CONCLUSION

Arrivé au terme de notre exercice ou il a été question de présenter estimation d'un projet par modélisation algorithmique. Basé sur le cycle de développement logiciel en cascade ce modèle présente de nombreux avantages tel que son objectivité mais présente aussi des inconvénients tel que son inflexibilité et son abstraction. Il serait donc judicieux d'utiliser en plus du modèle algorithmique une autre méthode d'estimation afin de comparer les résultats obtenus par les 2ce qui améliore grandement la prévision du cout du projet. Comme autre méthode nous pouvons citer entre autres l'estimation par analogie ou encore celle du jugement des experts qui se basent sur des données réelles.

