

COCOMO

Bourhis David
Caballero Flor
Coeurdoux Fabian
Rolland Lydie

Introduction

- Période économique néfaste :
Réduction des coûts

=> méthode de gestion :
COCOMO

Présentation de COCOMO

- COnstructive COst MOdel
- Premier modèle : 1981 par Barry Boehm
- Modèle arithmétique : relations mathématiques entre les variables

Principes de COCOMO (1)

- 3 types de modèles :
 - le modèle de base
 - le modèle intermédiaire
 - le modèle détaillé
- Estimer un projet :
 - Le code sans les commentaires
 - Du codage à l'intégration

Principes de COCOMO (2)

- Découpée en 3 catégories de projets :
 - Organique : <50KLOC
 - Semi détaché : <300KLOC
 - Imbriqué : >300KLOC

Objectifs de COCOMO

- Evaluer des critères de projets :
 - Effort
 - La durée
 - L'effectif
 - La productivité

Modèle de base (1)

- Estime :
 - l'effort
 - la durée du projet

Modèle de base (2)

Equations pour calculer l'effort et la productivité selon le type de projet

Types de projet	Effort	Productivité
Organique	$MM = 2,4 * (KLOC)^{1,05}$	$TDEV = 2,5 * (MM)^{0,38}$
Semi-détaché	$MM = 3 * (KLOC)^{1,12}$	$TDEV = 2,5 * (MM)^{0,35}$
Imbriqué	$MM = 3,6 * (KLOC)^{1,20}$	$TDEV = 2,5 * (MM)^{0,32}$

Exemple : projet organique

Taille produite	Effort	Productivité	Moyenne de personnel
2 KLOC (Petit)	5.0 MM	4.6	1.1
8 KLOC (Intermédiaire)	21.3 MM	8.0	2.7
32 KLOC (Moyen)	91.0 MM	14.0	6.5
128 KLOC (Grand)	392.0 MM	24.0	16.0

Moyenne de personnel = Effort / Productivité

Modèle intermédiaire (1)

Equations pour calculer l'effort et la productivité selon le type de projet

Types de projet	Effort	Productivité
Organique	$MM = 3,2 * (KLOC)^{1,05}$	$TDEV = 2,5 * (MM)^{0,38}$
Semi détaché	$MM = 3,0 * (KLOC)^{1,12}$	$TDEV = 2,5 * (MM)^{0,35}$
Imbriqué	$MM = 2,8 * (KLOC)^{1,20}$	$TDEV = 2,5 * (MM)^{0,32}$

Modèle intermédiaire (2)

- Introduit 15 facteurs correcteurs
- Regroupé en 4 catégories :
 - Attributs du produit
 - Attributs de l'ordinateur
 - Attributs du personnel
 - Attributs du projet

Facteurs correcteurs

Facteurs de productivité	Evaluation					
	Très bas	Bas	Nominal	Haut	Très haut	Extrêmement haut
Attributs du produit						
RELY	.75	.88	1.00	1.15	1.40	
DATA		.94	1.00	1.08	1.16	
CPLX	.70	.85	1.00	1.15	1.30	1.65
Attributs de l'ordinateur						
TIME			1.00	1.11	1.30	1.66
STOR			1.00	1.06	1.21	1.56
VIRT		0.87	1.00	1.15	1.30	
TURN		0.87	1.00	1.07	1.15	
Attributs du personnel						
ACAP	1.46	1.19	1.00	.86	.71	
AEXP	1.29	1.13	1.00	.91	.82	
PCAP	1.42	1.17	1.00	.86	.70	
VEXP	1.21	1.10	1.00	.90		
LEXP	1.14	1.07	1.00	.95		
Attributs du projet						
MODP	1.24	1.10	1.00	.91	.82	
TOOL	1.24	1.10	1.00	.91	.83	
SCED	1.23	1.08	1.00	1.04	1.10	

Exemple explicatif

- RELY : fiabilité requise pour le logiciel
- Projet de type organique de 10 KLOC
 - $MM = 3.2 * (10)^{1.05} = 36$
- Choix du facteur correcteur selon mon exigence
 - très faible fiabilité : $MM * 0.75 = 27$
 - très forte fiabilité : $MM * 1.4 = 50.4$

Etapes d'utilisation (1)

- Identifier le mode de développement
- Estimer le nombre de LOC
- Calculer le nombre de homme-mois :
$$MM_{base} = p * (KLOC)^e$$

Etapes d'utilisation (2)

- Estimer les 15 facteurs de productivité
- Calculer le facteur d'ajustement « a »
- Multiplier l'effort « nominal » par le facteur d'ajustement $MM = MM_{base} * a$

Le modèle détaillé de COCOMO (1)

- Evolution du modèle intermédiaire
- Inclut les 4 phases d'un projet :
 - Développement/conception
 - Finition de la conception
 - Tests au codage
 - Tests à l'implémentation

Le modèle détaillé de COCOMO (2)

- Suit une hiérarchie à trois niveaux :
 - Système
 - Sous-système
 - Module

- Dispose de 2 diagrammes :
 - Système / sous-système
 - Module

Diagramme sous-système/systeme

SHEP niveau sous-système

	1	2	8	Product		Computer attributes				Personnel		Project			32	20	33	34	35	36
				21	22	23	24	25	26	27	28	29	30	31	EAF	MM	MM			TOTALS
	SS NO	Subsystem	LOC	RELY	DATA	TIME	STOR	VIRT	TURN	ACAP	AEXP	MODP	TOOL	SCED	SS	MOD	EST	\$K/MM	\$K	AVGS
PD																				
DD																				
CUT																				
IT																				
			9	Total LOC												PD				
			10	(MM) _{NOM}						Phase						DD				
			11	(EDS/MM) _{NOM}						fraction						CUT				
										PD						IT				
										DD						Total				
										CUT							MM	Sched	\$K	
										IT										

Diagramme niveau module

SHEP niveau module

	3	4	5	6	7	14	15	16	17	18	13	19	37	
										EAF	MM	MM	MM	Percent
	SS NO	MOD NO	Module	LOC	AAF	CPLX	PCAP	VEXP	LEXP	M	NOM	MOD	EST	%
PD														
DD														
CUT														
IT														

Le modèle détaillé de COCOMO (3)

- Trois grandes opérations :
 - Regroupement d'informations
 - Evaluation
 - Calcul

Le modèle détaillé de COCOMO (4)

- Détermination du modèle détaillé:
 - Numéros et noms des modules et sous-systèmes
 - Nombre d'instructions
 - Effort nominal total puis par module

Le modèle détaillé de COCOMO (5)

- Facteur d'ajustement du diagramme module
- Effort modifié du niveau sous-système
- Facteur d'ajustement niveau sous-système
- Effort estimé
- Calculs finaux

COCOMO par l'exemple (1)

- Descriptions du projet :
 - Modèle COCOMO de base
 - Organique
 - 32000 lignes de codes

COCOMO par l'exemple (2)

- Calcul de l'effort :

$$MM = 2.4(KLOC)^{1.05}$$

- Calcul du temps de développement :

$$TDEV = 2.5(MM)^{0.38}$$

COCOMO par l'exemple (3)

- Calcul de l'effort :

$$MM = 2.4(32)^{1.05} = 91 \text{ hommes-mois}$$

- Calcul du temps de développement :

$$TDEV = 2.5(91)^{0.38} = 14 \text{ mois}$$

- Soit une productivité de :

$$\frac{32000}{91} = 352 \text{ lignes de codes}$$

COCOMO par l'exemple (4)

		32 Kloc
		...
Programmation	...	62
...		...
Programmation	...	55

COCOMO par l'exemple (5)

- Effort programmation :
 $(0.62)(91) = 56$ hommes-mois
- Temps prévu de programmation :
 $(0.55)(14) = 7.7$ mois

COCOMO par l'exemple (6)

- Nombre de personnes à temps plein :

$$\frac{56}{7.7} = 7.3 \text{ personnes}$$

COCOMO par l'exemple (7)

- Estimation Maintenance :

$$ACT = \frac{4000 + 2400}{32000} = 0.2$$

COCOMO par l'exemple (8)

- Effort pour la Maintenance :

$$(MM)_{AM} = 1.0(ACT)(MM) = 18$$

- Temps plein Maintenance

$$(FSP)_{AM} = \frac{(MM)_{AM}}{12} = 1.5$$

COCOMO II

- En 1998
- Plus adapté à l'aspect réutilisation des composants
- Composé de 3 modèles

COCOMO II : les modèles

- **Modèle de composition d'application**
utilisé pour les projets fabriqués à l'aide d'outils graphique
- **Modèle avant projet** : obtenir une estimation approximative avant de connaître l'architecture définitive
- **Modèle post-architecture** :
 - le plus détaillé
 - à utiliser après le développement de l'architecture générale du projet

[Historique]

- 1979 Mise au point et publiée par Albrecht d'IBM
- 1986 Création de l'IFPUG (International Function Point Users Group)
- 1992 Création de la FFUP (French Function Point User's Group)
- 1994 Dernière version (4.1)

Objectifs et avantages

■ Objectif

Mesurer la taille fonctionnelle des logiciels du point de vue de leurs utilisateurs

■ Avantages

- Méthode indépendante de la technologie
- Résultat simple: un nombre de points
- Rapide à calculer

Comment compter les points (1)

■ Les 5 étapes (1)

Détermination du périmètre de l'application

Comment compter les points (2)

■ Périmètre de l'application

○ Frontières de l'application

Le plus petit ensemble de fonctionnalités formant un tout cohérent pour l'utilisateur

○ Identification de l'utilisateur

Dépendante des objectifs de la mesure et modifie le périmètre de la mesure

○ Règles et procédures des frontières du comptage

La frontière déterminée sur la base du point de vue utilisateur

Comment compter les points (3)

■ Les 5 étapes (2)

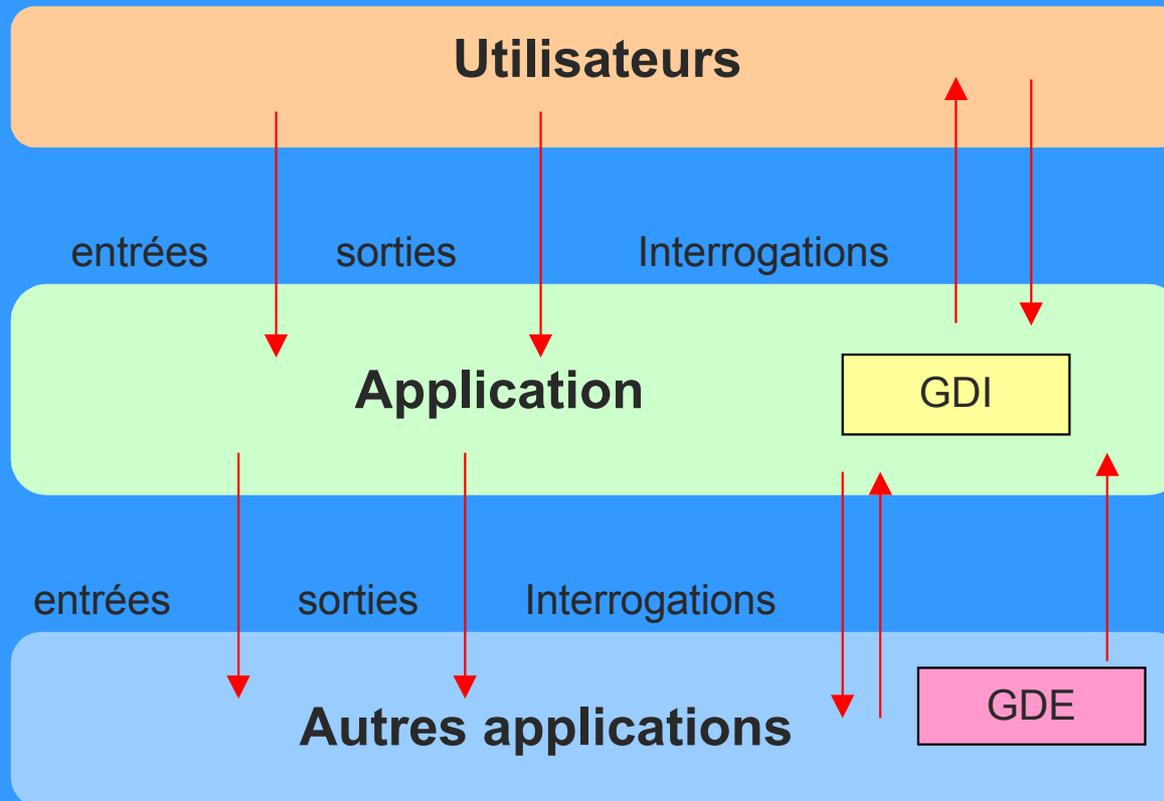
Détermination du périmètre de l'application



Identification du composant

Comment compter les points (4)

■ Les différents composants des points de fonction



Comment compter les points (5)

■ Les 5 étapes (3)

Détermination du périmètre de l'application



Identification du composant



Evaluation de la complexité de chaque composant

Comment compter les points (6)

■ La complexité des composants

Exemple : Complexité des GDI et GDE

- déterminé par le nombre de SLD et de DE
- SLD = sous groupements de GDI (ou GDE) fondés sur la vue logique de l'utilisateur des données.
- DE = zones non récurrentes que l'utilisateur peut reconnaître

SLD \ DE	1-19	20-50	51 et plus
1	Faible	Faible	Moyen
2 à 5	Faible	Moyen	Elevé
6 ou plus	Moyen	Elevé	Elevé

Comment compter les points (7)

■ Les 5 étapes (4 et 5)

Détermination du périmètre de l'application



Identification du composant



Evaluation de la complexité de chaque composant



Calcul du poids en Points de fonction de chaque Composant



Calcul du nombre brut de Points de fonction

Comment compter les points (8)

■ Calcul du poids en fonction et du nombre brut

Type de composant	Complexité	Nombre	Total par complexité	Totaux par types de composant
GDI	F	—	x 7 =	—
	M	—	x 10 =	—
	E	—	x 15 =	—
GDE	F	==	x 5 =	==
	M	==	x 7 =	==
	E	—	x 10 =	—
ENT	F	—	x 3 =	—
	M	—	x 4 =	—
	E	—	x 6 =	—
SOR	F	==	x 4 =	==
	M	==	x 5 =	==
	E	—	x 7 =	—
INT	F	—	x 3 =	—
	M	—	x 4 =	—
	E	—	x 6 =	—

Nombre de points de fonction bruts =

Différents outils d'évaluation (1)

■ Outils associés à la méthode COCOMO

- **BYL** (Before You Leap), développé par le groupe de Gordon
- **WICOMO** (Wang Institute Cost Model, développé à l'institut de Wang)
- **DECPlan** (développé par Digital Equipment Corporation)

Différents outils d'évaluation (2)

■ SLIM

Outil associé à la méthode Rayleigh-Putnam

■ ESTIMACS

Macro modèle d'évaluation

■ L'outil SPQR/20

Développé par Software Productivity Research, Inc

Différents outils d'évaluation (3)

- Aide à l'évaluation
- Résultats utilisés en tant que “ point de repères ”
- Pas seule source pour une évaluation

Conclusion

- Positif :
 - Estimations fiables
 - Cocomo est une référence
 - Profits des projets précédents
- Négatif :
 - Ne gère pas les composants existants
 - Estimation nombre lignes de code

[Questions ?]

