

2.7 La méthode d'estimation COCOMO (COntstructive COst MOdel)

a- Estimation de charges : unités de mesure de l'effort

- Charge : quantité de travail qu'une personne peut réaliser

Notions utiles :

a- La notion de durée-elapse (« Elapse time »)

Cette notion désigne la durée totale du projet. Un projet qui commence le 1^{ier} janvier et se termine le 30 avril de la même année a une durée de quatre mois indépendamment des vacances, des jours fériés et du travail effectif de l'équipe de projet.

La notion de durée-effort

Cette notion désigne la durée du projet en termes d'effort et de travail effectif des ressources attribuées au projet. Cet effort sera désigné par une valeur exprimée en jours/homme (j/h), semaine/homme (s/h) ou mois/homme (m/h), c'est-à-dire par la durée élémentaire de travail d'une ressource

- Durée, dépend de : charge ET nb personnes

Exemple : 60 m/h peut être 1 personne pendant 5 ans ou
10 personnes pendant 6 mois ou
60 personnes pendant 1 mois

b- Taille du projet

Se mesure à sa charge :

ordre de grandeur (selon les normes ISO) :

charge < 6 m/h ⇒ très petit projet

6 m/h ≤ charge ≤ 12 m/h ⇒ petit projet

12 m/h ≤ charge ≤ 30 m/h ⇒ projet moyen

30 m/h ≤ charge ≤ 100 m/h ⇒ grand projet

100 m/h ≤ charge ⇒ très grand projet

c. Principe de Cocomo.

COCOMO est un modèle qui permet d'estimer le coût, l'effort et le temps nécessaire au développement d'un logiciel. Le modèle original de COCOMO a été édité la première fois par le Dr. Barry Boehm en 1981 et a reflété les pratiques en matière de développement de logiciel de cette époque. Durant les 15 années suivantes les techniques de développement de logiciel ont changé (traitements en temps réel, apparition des ateliers de génie logiciel très performants ...).

Ces changements ont commencé à faire apparaître des problèmes dans le modèle original de COCOMO. La solution était de réinventer le modèle pour les années 90. Ce nouveau modèle prend en compte les évolutions des techniques de développement survenues depuis les années 80. Il est aussi prêt pour les évolutions futures.

Selon la méthode Cocomo utilisée, on peut évaluer certains des critères de projets suivant :

- Effort
- La durée
- L'effectif
- La productivité

d. Le modèle de base Cocomo 81

Elle se base sur une approche algorithmique pour déterminer « l'effort » et le « temps de développement » d'une application. Elle est basé sur le nombre de lignes de code en Kilo (KLOC : Kilo line of code) ou le nombre de milliers d'instructions source livrées KDSI (Kilo Delivered Source Instruction).

Exemple :

1 ligne comprenant N instruction compte pour 1 LOC

1 déclaration sur N lignes compte pour N LOC

Dans cette méthode, ne sont pas prises en compte :

- Les commentaires
- Les instructions de test provisoire, utiles uniquement pour la phase de test
- Les lignes de code générées automatiquement

Le modèle COCOMO 81 est en fait constitué de trois modèles de calcul et distingue 3 types de projets :

Les trois modèles sont :

- le modèle de base
- le modèle intermédiaire
- le modèle détaillé ou expert

Les trois types de projets sont :

- *Organique :*

Application simple, réalisée par une équipe expérimentée ayant l'habitude de travailler ensemble, maîtrisant le langage et l'environnement de développement.

- *Semi-Détaché :*

Niveau intermédiaire, le projet n'est ni trop simple ni trop compliqué, l'équipe de développement a déjà réalisé quelques projets ensemble mais n'est pas totalement rodée. Les technologies et le domaine d'application sont un peu flous, mais pas de grosses difficultés.

- *Imbriqué :*

Techniques innovantes, organisation complexe, couplage fort avec beaucoup d'interactions. Technologie et domaine nouveau, équipe 'jeune'.

Par rapport au nombre de lignes de code en Kilo KLOC, on considère :

- ✓ Organique : <50 KLOC
- ✓ Semi détaché (Médian): <300 KLOC
- ✓ Imbriqué : >300 KLOC

d.1 Le modèle de base

Le modèle de base estime l'effort (le nombre de homme mois) en fonction du nombre de milliers d'instructions source livrées(KDSI), de la productivité (le nombre de lignes de code par personne par mois) et d'un facteur d'échelle qui dépend du type de projet.

Les formules de calcul :

TYPE DE PROJET	Effort en homme mois(HM)	Temps de développement
ORGANIQUE	$2.4(KDSI)^{1.05}$	$2.5(HM)^{0.38}$
MEDIAN	$3.0(KDSI)^{1.12}$	$2.5(HM)^{0.35}$
IMBRIQUE	$3.6(KDSI)^{1.20}$	$2.5(HM)^{0.32}$

Tableau 1 formules d'estimation COCOMO pour le modèle de base en fonction de KDSI

Exemples :

Effort (HM)	2KDSI	8KDSI	32KDSI	128 KDSI	512 KDSI
Organique	5	21,3	91	392	
Médian	6,5	31	146	687	3250
Imbriqué	8,3	44	230	1216	6420

TDEV(mois)	2KDSI	8KDSI	32KDSI	128 KDSI	512 KDSI
Organique	4.6	8	14	24	
Médian	4.8	8,3	14	24	42
Imbriqué	4.9	8,4	14	24	41

Le temps de développement commence après les spécifications fonctionnelles et s'arrête après l'intégration. De ces chiffres on peut déduire la productivité et le nombre moyen de personnes sur le projet FSP

productivité = (KDSI/HM)

FSP=HM/TDEV

On peut ensuite calculer la distribution de l'effort par phases (en %)

- RPD (Requirements and Preliminary Design): Conception globale et Plan d'intégration
- DD (Detail Design) : Conception détaillée
- CUT (Code and Unit Test) : Programmation et Tests unitaires
- IT (Integration and Test) : Intégration

<i>PROJET ORGANIQUE</i>	2 KDSI	8 KDSI	32 KDSI	128 KDI	512 KDSI
RPD	16	16	16	16	
DD	26	25	24	23	
CUT	42	40	38	36	
IT	16	19	22	25	
<i>PROJET MEDIAN</i>					
RPD	17	17	17	17	17
DD	27	26	25	24	23
CUT	37	35	33	31	29
IT	19	22	25	28	31
<i>PROJET IMBRIQUE</i>					
RPD	18	18	18	18	18
DD	28	27	26	25	24
CUT	32	30	28	26	24
IT	22	25	28	31	34

Tableau 2 : Distribution de l'effort par phases en pourcentage

<i>ORGANIQUE</i>	2 KDSI	8 KDSI	32 KDSI	128 KDI	512 KDSI
RPD	19	19	19	19	
DD et CUT	63	59	55	51	
IT	18	22	26	30	
<i>MEDIAN</i>					
RPD	24	25	26	27	28
DD et CUT	56	52	48	44	40
IT	20	23	26	29	32
<i>IMBRIQUE</i>					
RPD	30	32	34	36	38
DD et CUT	48	44	40	36	32
IT	22	24	26	28	30

Tableau 3 : Distribution du temps de développement par phases en pourcentage

d.2 Le modèle intermédiaire

Le modèle de base ne prend en compte que le nombre de lignes source et induit des discontinuités un peu brutales au niveau du personnel entre chaque phase du cycle de vie ; ce qui peut perturber l'organisation du projet.

Le modèle intermédiaire introduit 15 facteurs de productivité (appelés 'cost drivers'), représentant un avis subjectif du produit, du matériel, du personnel, et des attributs du projet. Chaque facteur prend une valeur nominative de 1,

et peut varier selon son importance dans le projet. Ces facteurs sont à rapprocher des caractéristiques générales du produit. Les 15 facteurs sont multipliés entre eux pour donner un facteur d'ajustement qui vient modifier l'estimation donnée par la formule de base.

Facteurs de productivité

- Logiciel

- RELY: Fiabilité requise
- DATA: Volume des données manipulées
- CPLX: Complexité du produit

- Matériel

- TIME: Contraintes de temps d'exécution
- STOR: Contraintes de taille mémoire
- VIRT: Instabilité de la mémoire

- Personnel

- ECAP: Aptitude de l'équipe
- AEXP: Expérience du domaine
- VEXP: Expérience de la machine virtuelle
- LEXP: Maîtrise du langage

- Projet

- MODP: Pratique de développement évolués
- TOOL: Utilisation d'outils logiciels
- SCED: Contraintes de délais

Multiplicateurs	Très bas	Bas	Nominal	Elevé	Très élevé	Extrêmement élevé
RELY	0,75	0,88	1	1,15	1,4	
DATA		0,94	1	1,08	1,16	
CPLX	0,7	0,85	1	1,15	1,3	1,65
TIME			1	1,11	1,3	1,66
STOR			1	1,06	1,21	1,56
VIRT		0,87	1	1,15	1,3	
ECAP	1,44	1,18	1	0,86	0,7	
AEXP	1,29	1,13	1	0,91	0,82	
VEXP	1,21	1,1	1	0,9		
LEXP	1,14	1,07	1	0,95		
MODP	1,24	1,1	1	0,91	0,82	
TOOL	1,24	1,1	1	0,91	0,83	
SCED	1,23	1,08	1	1,04	1,1	

Tableau 4 : Coefficients multiplicateurs pour chacun des facteurs de productivité

TYPE DE PROJET	Effort en homme mois(HM)	Temps de développement
ORGANIQUE	3.2(KDSI) ^{1.05}	2.5(HM) ^{0.38}
MEDIAN	3.0(KDSI) ^{1.12}	2.5(HM) ^{0.35}
IMBRIQUE	2.8(KDSI) ^{1.20}	2.5(HM) ^{0.32}

Tableau 5 : Coefficients pour le calcul d'effort et de temps dans le modèle intermédiaire

Calcul de l'effort (HM) dans le modèle intermédiaire :

Identifier le mode de développement (organique, médian, imbriqué) . Ceci donne 4 coefficients au modèle : p (productivité), e (échelle appliquée à la taille du logiciel), c (constante du modèle) et d (échelle appliquée au temps de développement) conformément au tableau 5.

Estimer le nombre de lignes du code source (en KDSI), puis calculer le nombre d'homme*mois par la formule appropriée du tableau 5. On obtient l'effort HM base

Estimer les 15 facteurs de productivité et calculer le facteur d'ajustement (a) en multipliant les facteurs ensemble conformément au tableau 4

Multiplier l'effort 'nominal' par le facteur d'ajustement :

$$HM = HM \text{ base } * a$$

Calculer le temps de développement en utilisant le tableau 5:

$TDEV = c(HM)^d$ (on notera que les coefficients restent inchangés par rapport au modèle de base)

Dans le modèle intermédiaire, les coefficients multiplicateurs s'appliquent uniformément sur l'ensemble des phases.

d.3 Modèle expert

Le modèle expert inclut toutes les caractéristiques du modèle intermédiaire avec une estimation de l'impact de la conduite des coûts sur chaque étape du cycle de développement : définition initiale du produit, définition détaillée, codage, intégration . De plus, le projet est analysé en terme d'une hiérarchie : module, sous système et système. Il permet une véritable gestion de projet, utile pour de grands projets. Le modèle expert ne sera pas décrit.

e. Le modèle Cocomo II

COCOMO II peut être calibré pour mieux correspondre aux projets de l'entreprise. COCOMO II tient compte de la réutilisation et est constitué de trois modèles :

Modèle de composition d'application :

Ce modèle est utilisé pour les projets fabriqués à l'aide des toolkits d'outils graphiques. Il est basé sur les nouveaux 'Object Points'.

Modèle avant projet :

Modèle utilisé pour obtenir une estimation approximative avant de connaître l'architecture définitive. Il utilise un sous ensemble de facteurs de productivité (cost drivers). Il est basé sur le nombre de lignes de code ou les points de fonction non ajustés.

Modèle post-architecture :

Il s'agit du modèle le plus détaillé de COCOMO II. A utiliser après le développement de l'architecture générale du projet. Il utilise des facteurs de productivité (cost drivers) et des formules.

Les facteurs utilisés sont classés en : Facteurs d'échelle, Urgence, Flexibilité de développement, résolution d'architecture/Risque, Cohésion d'équipe et Maturité de Processus.