

Le modèle COCOMO intermédiaire

Le modèle COCOMO Intermédiaire est une extension du modèle COCOMO de base. Il en diffère principalement par la prise en compte de 15 facteurs, en plus de la taille (mesurée en KLOC) et du type de logiciel (mesuré sur une échelle de trois valeurs: Simple (Organique), Moyen (Semi-détaché) et complexe (Intégré)). Ces 15 facteurs justifient la variation entre les coûts de deux logiciels de même taille. Cette variation n'est pas révélée ni expliquée par le modèle COCOMO de base.

L'équation d'estimation du modèle de l'effort COCOMO Intermédiaire en Homme-Mois (HM) d'un projet logiciel est:

$$Charge = a \times KLOC^b \times \prod_{i=1}^{15} C_i$$

L'équation de l'estimation de la durée est :

$$Durée = b \times Charge^c$$

Les paramètres a , b , c , et d du modèle COCOMO intermédiaire, en fonction du type du projet, sont présentés dans le tableau suivant :

Type de projet	Simple	Moyen	Complexe
a	3,20	3,00	2,80
b	1,05	1,12	1,20
c	2,50	2,50	2,50
d	0,38	0,35	0,32

C_i représente la valeur du facteur d'effort i . Il existe 15 facteurs d'effort qui sont présentés dans le tableau suivant :

Attributs du Produit
RELY (<i>Required Software Reliability</i>): Fiabilité requise
DATA (<i>Data Base Size</i>): Taille de la base de données
CPLX (<i>Product Complexity</i>): Complexité du logiciel
Attributs du Matériel
TIME (<i>Execution Time Constraint</i>): Contrainte du temps d'exécution
STOR (<i>Main storage Constraint</i>): Contrainte de la taille mémoire
VIRT (<i>Virtual machine Volatility</i>): Instabilité de la plateforme
TURN (<i>Computer Turnaround Time</i>): Temps de restitution de l'ordinateur
Attributs du Personnel
ACAP (<i>Analyst Capability</i>): Compétence des analystes
PCAP (<i>Programmer Capability</i>): Compétence des programmeurs
AEXP (<i>Application Experience</i>): Expérience du domaine d'application
VEXP (<i>Virtual Machine Experience</i>): Expérience dans la plateforme
LEXP (<i>Programming Language Experience</i>): Expérience du langage de programmation
Attributs du Projet
MODP (<i>Modern Programming Practices</i>): Pratiques des méthodes de programmation
TOOL (<i>Use of Software Tools</i>): Utilisation d'outils logiciels
SCED (<i>Required Development Schedule</i>): Contraintes de planification

Chaque facteur est évalué avec une *note* (valeurs linguistiques), ensuite, la note est convertie à une *valeur* (*facteur d'ajustement*). Les notes qu'un facteur d'effort peut prendre sont :

Très faible, Faible, Moyen, Élevé, Très élevé, et Extra-élevé

Les évaluations possibles des 15 facteurs et les valeurs correspondantes sont représentées dans les tableaux suivant :

Facteur	Evaluation					
	Très faible	Faible	Moyen	Elevé	Très élevé	Extra-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	
TURN		0,87	1	1,07	1,15	
ACAP	1,46	1,19	1	0,86	0,71	
AEXP	1,29	1,13	1	0,91	0,82	
PCAP	1,42	1,17	1	0,86	0,7	
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	

I. Attribut du produit (*Product Attributes*)

1. Fiabilité requise (*Required Reliability* : RELY)

Ce facteur représente l'effet des exigences concernant la fiabilité, i.e. l'aptitude dans laquelle le logiciel peut accomplir de manière satisfaisante ses fonctions prévues.

Évaluation (RELY)	Définition
Très faible	L'effet d'une défaillance du logiciel est tout simplement l'inconvénient que les développeurs doivent réparer la faute.
Faible	L'effet d'une défaillance du logiciel est une perte de niveau faible et facilement récupérable par les utilisateurs.
Moyen	L'effet d'une défaillance du logiciel est une perte modérée pour les utilisateurs, mais une situation à laquelle on peut se remettre sans peine extrême.
Elevé	L'effet d'une défaillance du logiciel peut être une perte financière majeure ou une difficulté humaine importante.
Très élevé	L'effet d'une défaillance du logiciel peut être la perte de la vie humaine

2. Taille de la base de données (*Data Base Size: DATA*)

Cette mesure tente de capturer l'impact des exigences de données volumineuses sur l'effort de développement. La raison pour laquelle la taille de la base de données est importante à considérer, est l'effort requis pour générer les données de test qui seront utilisées pour tester le programme. Ce facteur est mesuré par :

$$D/P = (\text{Taille de la base de données en octets ou caractères}) / (\text{Taille des programmes en LOC})$$

Évaluation (DATA)	Définition
Faible	$D/P < 10$
Moyen	$10 \leq D/P \leq 100$
Elevé	$100 \leq D/P \leq 1000$
Très élevé	$D/P > 1000$

3. Complexité du produit (*Product Complexity: CPLX*)

Ce facteur est une évaluation de la moyenne subjective de quatre types de fonctions de contrôle : (1) Opérations de contrôle, (2) Opérations de calcul, (3) Opérations dépendantes des périphériques, et (4) Opérations de gestion de données. (L'évaluation varie de *Très bas* à *Très élevé*).

Pour plus de détails : <http://www.functionpointmodeler.com/fpm-infocenter/topic/com.functionpointmodeler.fpm.help/ditafiles/concepts/cc-57.html>

II. Attribut du matériel (*Computer Attributes*)

4. Contrainte du temps d'exécution (*Execution Time Constraint: TIME*)

Ce facteur est une mesure de la contrainte du temps d'exécution imposée à un système logiciel. La note est exprimée en fonction du pourcentage du temps d'exécution disponible qui devrait être utilisé par le système.

Évaluation (TIME)	Définition
Moyen	$\leq 50\%$ d'utilisation du temps d'exécution disponible
Elevé	70% d'utilisation du temps d'exécution disponible
Très élevé	85% d'utilisation du temps d'exécution disponible
Extra élevé	95% d'utilisation du temps d'exécution disponible

5. Contrainte du stockage principale (*Main storage Constraint : STOR*)

Ce facteur représente le pourcentage du stockage principal qui devrait être utilisé par le logiciel. Le stockage principal se réfère au stockage direct (disques durs, mémoire vive).

Évaluation (STOR)	Définition
Moyen	$\leq 50\%$ d'utilisation du stockage disponible
Elevé	70% d'utilisation du stockage disponible
Très élevé	85% d'utilisation du stockage disponible
Extra élevé	95% d'utilisation du stockage disponible

6. Instabilité de la plateforme (*Virtual machine Volatility: VIRT*)

La «plateforme» désigne le matériel (ordinateurs, réseau) et le logiciels (SE , SGBD, etc.) que la production du logiciel appelle pour effectuer ses tâches. Les notations sont définies en fonction de la fréquence relative des changements majeurs et mineurs:

- Changement majeur : affecte de manière significative environ 10% des modules en cours de développement.
- Changement mineur : affecte sensiblement environ 1% des modules en cours de développement.

Évaluation (VIRT)	Définition
Faible	Changement majeur tous les 12 mois
Moyen	Changement majeur tous les 6 mois; Mineur: 2 semaines
Elevé	Majeur: 2 mois; Mineur: 1 semaine.
Très élevé	Majeur: 2 semaines; Mineur: 2 jours.

7. Temps de restitution de l'ordinateur (*Computer Turnaround Time: TURN*)

Ce facteur représente le niveau du temps de réponse de l'ordinateur utilisé par l'équipe de projet qui développe le produit logiciel. Le temps de réponse est le temps moyen depuis le moment où le développeur lance l'exécution jusqu'à ce que les résultats soient restitués aux développeurs.

Évaluation (TURN)	Définition
Faible	Interactif.
Moyen	temps d'exécution <4 heures.
Elevé	4 à 12 heures.
Très élevé	> 12 heures.
Extra-élevé	Pas de note - par défaut à très élevé

Remarque :

Compte tenu de l'augmentation considérable des temps d'exécution des processeurs et des capacités de stockage, les contraintes du temps d'exécution (TIME) et du stockage (STOR), et le Temps de restitution (TURN) sont devenues moins importants. Cependant, de nombreuses applications continuent de se développer pour consommer toutes les ressources disponibles.

III. Attributs du personnel (*Personnel Attributes*)

8. Compétence des analystes (*Analyst capability : ACAP*)

Les analystes sont des employés qui travaillent sur l'élaboration et la validation des exigences, la conception générale et la conception détaillée, et sont fortement impliqués dans l'intégration et le test d'intégration. Les notes pour la capacité des analystes sont exprimées en termes de *percentiles* (mesure statistiques) par rapport à la population globale des analystes logiciels. Les principaux attributs à considérer sont la capacité, l'efficacité, la rigueur et la capacité de communiquer et de coopérer.

Cette évaluation **ne doit pas inclure l'expérience** (prise en compte dans d'autres facteurs) et devrait être basée sur la capacité des analystes en tant qu'équipe plutôt que les individus.

Évaluation (ACAP)	Définition
Très faible	15 ^{ème} percentile
Faible	35 ^{ème} percentile
Moyen	55 ^{ème} percentile
Elevé	75 ^{ème} percentile
Très élevé	90 ^{ème} percentile

9. Compétence des programmeurs (Programmer Capability: PCAP)

Ce facteur représente la capacité des programmeurs qui travailleront sur le produit logiciel. Les notes sont exprimées en termes de percentiles par rapport à la population globale des programmeurs. Les principaux facteurs qui devraient être pris en compte dans le classement sont la capacité, l'efficacité, la rigueur et la capacité de communiquer et de coopérer. L'évaluation **ne devrait pas tenir compte du niveau d'expérience des programmeurs** (elle est couverte par d'autres facteurs) et elle devrait être basée sur la capacité des programmeurs en tant qu'équipe et non comme individu.

Évaluation (PCAP)	Définition
Très faible	15 ^{ème} percentile
Faible	35 ^{ème} percentile
Moyen	55 ^{ème} percentile
Elevé	75 ^{ème} percentile
Très élevé	90 ^{ème} percentile

10. Expérience du domaine d'application (Application Experience: AEXP)

Ce facteur représente le niveau d'expérience aux applications équivalentes de l'équipe du projet qui développe le produit logiciel.

Évaluation (AEXP)	Définition
Très faible	<= Expérience de 4 mois.
Faible	1 an d'expérience.
Moyen	3 ans d'expérience.
Elevé	6 ans d'expérience.
Très élevé	12 ans d'expérience

11. Expérience dans la plateforme (Virtual Machine Experience: VEXP)

Ce facteur représente l'expérience de l'équipe du projet avec la plateforme matérielle et logicielle utilisée pour accomplir le développement du logiciel, par exemple l'ordinateur, le système d'exploitation, et le système de gestion de base de données (le langage de programmation n'est pas considérée comme faisant partie de ce facteur).

Évaluation (VEXP)	Définition
Très faible	<= 1 mois d'expérience.
Faible	4 mois d'expérience.
Moyen	1 an d'expérience.
Elevé	3 ans d'expérience.
Très élevé	Pas d'évaluation - par défaut High.

12. Expérience du langage de programmation (Programming Language Experience: LEXP)

Ce facteur représente le niveau d'expérience au langage de programmation que l'équipe du projet utilise pour développer le logiciel. Les notes sont définies en fonction de la durée d'expérience de l'équipe du projet avec le langage de programmation à utiliser.

Évaluation (LEXP)	Définition
Très faible	<= 1 mois d'expérience.
Faible	4 mois d'expérience.
Moyen	1 an d'expérience.
Elevé	3 ans d'expérience.

IV. Attributs du Projet

13. Pratiques des méthodes de programmation (Modern Programming Practices: MODP)

Ce facteur représente l'effet des pratiques de programmation modernes utilisées dans le développement du logiciel. Les pratiques spécifiques incluses ici sont :

- Application de méthodologies de développement logiciel.
- Inspection logiciel (documents de spécifications/conception, code source, tests, etc.).
- Structuration du code (commentaires, indentation du code, etc.),
- Utilisation de bibliothèques, etc.

Évaluation (MODP)	Définition
Faible	Début de l'utilisation.
Moyen	Certains l'utilisent.
Elevé	Usage généralisé.
Très élevé	Utilisation courante.

14. Utilisation d'outils logiciels (Use of Software Tools: TOOL)

Ce facteur représente l'effet de l'utilisation des outils logiciels dans le développement du produit logiciel, tels que les outils de de conception, les outils de génération semi-automatique du code, les outils de génération de données de test, etc.

Évaluation (TOOL)	Définition
Très faible	Peu d'outils – un éditeur de code, un compilateur, un débogueur.
Faible	Simple outils CASE frontend et backend.
Moyen	Configuration normale d'outils- Des outils CASE de base du cycle de développement.
Elevé	Des outils CASE performants du cycle de développement.
Très élevé	Des outils CASE performants et intégrés du cycle de développement.

15. Contraintes du calendrier (Required Development Schedule: SCED)

Ce facteur représente le niveau de contrainte de calendrier imposé à l'équipe du projet par rapport au calendrier nominal, i.e., le calendrier considéré comme idéal pour réaliser le projet. Les notes sont en fonction du pourcentage par rapport au calendrier normal.

Évaluation	Définition
Très faible	75% du normal. Techniques d'accélération sévères seront nécessaires. Probablement inaccessible sans diminution de la qualité.
Faible	85% du nominal. Accélération modérée - possiblement réalisable avec une aide significative.
Moyen	100% - Calendrier nominal
Elevé	130% du nominal. Possibilité d'ennui et tendance au perfectionnisme.
Très élevé	160% de la valeur nominale. Etirement sévère du calendrier, problèmes liés à la continuité et l'instabilité du personnel.