
Coûts et durée des projets informatiques

pratique des modèles d'estimation

Jacques Printz
Christiane Deh
Bernard Mesdon
Nicolas Trèves

 **hermes**

Lavoisier

Table des matières

Chapitre 1. Introduction : la bonne pratique des modèles d'estimation	11
1.1 Avertissement.	11
1.2. Vade-mecum du chef projet	16
1.2.1. Le chef de projet est à un carrefour de conflits humains et d'intérêts économiques contradictoires	16
1.2.2. Le chef de projet doit avoir en tête quelques données simples	17
1.2.3. La relation économique coût-qualité-fonctionnalité-délai [CQFD] se stabilise dans l'ordre suivant	20
1.2.4. La productivité du développement dépend beaucoup plus de la compétence des personnes et de l'équipe que des outils	22
1.2.5. Le plan de développement requerra toute l'attention du chef de projet	23
1.2.6. Le chef de projet doit toujours se préoccuper et, au minimum, être bien informé du contexte organisationnel du MOA	24
1.2.7. Le chef de projet s'investira totalement, et surtout prioritairement, sur les phases critiques du cycle de développement là où l'incertitude est maximale	24
Chapitre 2. Les modèles d'estimation basés sur les instructions et les fonctions : COCOMO 81 et COCOMO II	27
2.1. Introduction.	27
2.2. Contexte d'utilisation du modèle.	29
2.2.1. Typologie des programmes informatiques	29

6 Coûts et durée des projets informatiques

2.2.2. Cycle de développement de référence et phases du modèle COCOMO	34
2.3. Le modèle COCOMO 81	35
2.3.1. Equations de base du modèle	35
2.3.2. Ventilation de l'effort par phase du projet	37
2.3.3. Répartition de l'effort et de la durée par phase	41
2.3.4. Répartition par activités de l'effort	43
2.4. Cas des applications composites	46
2.4.1. Combinaison de <i>S</i> , <i>P</i> , <i>E</i> programmes	46
2.4.2. La réutilisation de composants logiciel.	48
2.4.3. Les progiciels et les COTS	50
2.5. Approfondissement de la notion de KISL.	51
2.5.1. Les facteurs de coût	51
2.5.2. Influence des langages de programmation objet	55
2.5.3. Prise en compte de l'architecture et de l'intégration	57
2.6. Les modèles COCOMO intermédiaires et COCOMO détaillés.	59
2.6.1. Le modèle intermédiaire	59
2.6.2. Le modèle détaillé	61
2.7. Le modèle COCOMO II.	64
2.7.1. Motivations	64
2.7.2. Les hypothèses de base du modèle COCOMO II.	67
2.7.3. Le modèle nominal COCOMO II	70
2.7.4. Remarques des auteurs à propos de COCOMO II	73
2.8. Bibliographie.	74
2.9. Annexes	75
2.9.1. Annexe COCOMO 81	75
2.9.2. Annexe COCOMO II	85
Chapitre 3. Les points de fonction.	91
3.1. Qu'est-ce qu'une mesure de taille fonctionnelle ?	91
3.1.1. La vue utilisateur	91
3.1.2. Ce n'est pas une mesure d'estimation	92
3.1.3. Plusieurs méthodes	92
3.1.4. Un standard pour certifier ces méthodes : ISO 14143	93
3.2. Les points de fonction IFPUG 4.1	94

3.2.1. Description générale	94
3.2.2. Détail de la méthode de calcul	97
3.2.3. Méthode de calcul simplifié	105
3.2.4. Mesure des projets et des applications	106
3.3. Les autres méthodes	107
3.3.1. MkII Function Point Analysis	107
3.3.2. COSMIC Full Function Points	110
3.3.3. Experience Pro	111
3.3.4. Boeing 3D	112
3.4. Taille fonctionnelle et estimation de projet	112
3.4.1. La problématique	112
3.4.2. Quelle mesure de productivité ?	113
3.5. Autres domaines d'utilisation des mesures de taille fonctionnelle	114
3.5.1. Dans le cadre des projets	114
3.5.2. Au niveau d'un service informatique ou d'une entreprise	115
3.6. Mise en place des points de fonction dans une organisation	117
3.6.1. Objectifs recherchés	117
3.6.2. Types d'organisation	117
3.6.3. Phases de mise en place	118
3.7. Bibliographie	119
3.8. Abréviations et acronymes	119
Chapitre 4. Etudes de cas	121
4.1. Introduction	121
4.2. Typologie	122
4.2.1. Applications de gestion	122
4.2.2. Projets d'infrastructures logicielles et applications techniques	123
4.2.3. Intégration de systèmes	124
4.3. Démarche générale	125
4.3.1. Généralités	125
4.3.2. Les étapes de l'estimation	126
4.3.3. Choix du facteur de productivité et du facteur d'échelle	126
4.3.4. Choix des coefficients correcteurs	127
4.4. Présentation des cas	128
4.4.1. Facteurs de productivité utilisés	129

8 Coûts et durée des projets informatiques

4.4.2. Vue synthétique des cas	130
4.5. PP.GEST/GESTSTAT : petit projet de gestion en environnement PC	132
4.5.1. Méthode analytique	132
4.5.2. Application GESTSTAT	133
4.6. GP.GEST/GESTMAT : grand projet de gestion	135
4.6.1. Etape 1 : cadrage du projet	135
4.6.2. Etape 2 : modélisation fonctionnelle	136
4.6.3. Etape 3 : estimation des charges de développement	138
4.6.4. Etape 4 : estimation des charges additionnelles	141
4.6.5. Etape 5 : charge projet brute	141
4.6.6. Etape 6 : facteurs d'ajustement et charge nette	142
4.6.7. Etape 7 : délais et calendrier de développement	142
4.6.8. Etape 8 : analyse de risques	143
4.6.9. Etape 9 : analyse de la valeur	143
4.7. GP.TECH/POWERSIMU : grand projet du domaine technique.	146
4.7.1. Etape 1 : cadrage du projet	146
4.7.2. Etape 2 : modélisation fonctionnelle	146
4.7.3. Etape 3 à 6 : estimation des charges de développement	147
4.7.4. Etape 7 : délai de développement, dérive du projet	148
4.7.5. Etape 8 : analyse de risques	151
4.8. TGP.TECH/POWERCTRL : très grand projet du domaine technique.	152
4.8.1. Etape 1 : cadrage du projet	152
4.8.2. Etape 2 : modélisation fonctionnelle	152
4.8.3. Etape 3 : estimation des charges de développement	155
4.8.4. Etape 4 : estimation des charges additionnelles	156
4.8.5. Etape 5 : charge projet brute	156
4.8.6. Etape 6 : facteurs correcteurs et charge nette	157
4.8.7. Etape 7 : délais et calendrier de développement	157
4.8.8. Etape 8 : analyse de risques	157
4.8.9. Etape 9 : analyse de la valeur	157
4.9. TGP.INTSYS/GROUPWORK : très grand projet d'intégration de systèmes	157
4.9.1. Etape 1 : cadrage du projet	158
4.9.2. Etape 2 : modélisation fonctionnelle	158
4.9.3. Etape 3.1 : estimation des charges de développement des domaines métier	160

4.9.4. Etape 3.2 : charges de développement des services génériques et adaptations socle 165

4.9.5. Etape 4 : charges additionnelles 165

4.9.6. Etape 5 : charge projet brute 165

4.9.7. Etape 6 : facteurs d'ajustement et charge nette 166

4.9.8. Etape 7 : délais et calendrier de développement 166

4.9.9. Etape 9 : analyse de risques 166

Introduction : la bonne pratique des modèles d'estimation logiciel

Le développement logiciel est un processus complexe et multidisciplinaire. L'estimation des coûts et des délais est une tâche cruciale pour garantir le succès d'un projet. Cette introduction présente les bonnes pratiques pour la mise en œuvre de modèles d'estimation logiciel.

Le développement logiciel est un processus complexe et multidisciplinaire. L'estimation des coûts et des délais est une tâche cruciale pour garantir le succès d'un projet. Cette introduction présente les bonnes pratiques pour la mise en œuvre de modèles d'estimation logiciel.

Le développement logiciel est un processus complexe et multidisciplinaire. L'estimation des coûts et des délais est une tâche cruciale pour garantir le succès d'un projet. Cette introduction présente les bonnes pratiques pour la mise en œuvre de modèles d'estimation logiciel.

Le développement logiciel est un processus complexe et multidisciplinaire. L'estimation des coûts et des délais est une tâche cruciale pour garantir le succès d'un projet. Cette introduction présente les bonnes pratiques pour la mise en œuvre de modèles d'estimation logiciel.

Cet ouvrage présente les trois techniques les plus couramment utilisées pour estimer les projets logiciels. C'est le compagnon du volume *Productivité des programmeurs*.

Le vade-mecum du chef de projet est fondé sur l'expérience intuitive et concrète de nombreux chefs de projets. C'est un modèle qui fixe des ordres de grandeur.

Le second modèle, COCOMO, récemment mis à jour avec COCOMO II, est le plus complet de tous les modèles d'estimation, et il doit faire partie de la boîte à outils de tout chef de projet.

La méthode de comptage des points de fonctions, fondée sur les flux de données, permet quant à elle une analyse quantitative dès les phases expression de besoin – spécification fonctionnelle du cycle de développement. Les PF dimensionnent l'application du point de vue de l'utilisateur. La taille ainsi mesurée permet une estimation sur des bases quantitatives.

Ce volume se termine par une série d'études de cas, toutes issues de projets réels que nous avons expertisés en détail. Ces études montrent comment on peut, dans la pratique, combiner ces différents modèles et augmenter la fiabilité de l'estimation des projets en termes de Coût, Qualité, Fonctionnalité et Délai (CQFD).

Les auteurs

Jacques Printz est titulaire, depuis 1995, de la chaire de génie logiciel du CNAM où il a créé le Centre de Maîtrise des Systèmes et du Logiciel. Il enseigne également à l'Ecole Centrale de Paris.

Christiane Deh, Bernard Mesdon et Nicolas Trèves participent aux enseignements de la chaire et aux diverses activités du CMSL.

hermes
Science
— publications —

www.hermes-science.com

