

CENTRALE L Y O N	<h1>COCOMO II</h1> <p>Constructive Cost Model II Avec Costar</p> <p>BTD/GL/GP</p> <p style="text-align: right;">1</p>
---------------------	---------------------------------------------------------------------------------------------------------------------------

CENTRALE L Y O N	<h2>Brève histoire de COCOMO (1/2)</h2>
Bertrand DAVID : Génie Logiciel	<p>1981 : Version initiale de COCOMO a été proposée par Dr. Barry Boehm's textbook <i>Software Engineering Economics</i>. Ce modèle est maintenant appelé "COCOMO 81".</p> <p>1986 : Mise à disposition de la version Costar 1.0.</p> <p>1987 : Propositions de versions Ada COCOMO et Incremental COCOMO (proceedings, Third COCOMO Users Group Meeting, Software Engineering Institute).</p> <p>1988, 1989 : Raffinement de Ada COCOMO. Mise à disposition de la version Costar 2.0.</p> <p>1990 : Mise à disposition de la version Costar 3.0.</p> <p>1993 : Mise à disposition de la version Costar 4.0.</p> <p>1995, 1996 : Parution de premiers papiers décrivant COCOMO II.</p> <p>1997 : Première calibration du modèle COCOMO II par Dr. Boehm, appelée "COCOMO II.1997".</p> <p>Mise à disposition de la version Costar 5.0 supportant COCOMO II.</p> <p>1998 : Seconde calibration du modèle COCOMO II appelée "COCOMO II.1998".</p>
BTD/GL/GP	2

CENTRALE L Y O N	<h2>Brève histoire de COCOMO (2/2)</h2>
Bertrand DAVID : Génie Logiciel	<p>1999 : COCOMO II.1998 en renommé en COCOMO II.1999, puis en COCOMO II.2000 (Tous les trois modèles sont identiques).</p> <p>2000 : Publication du livre <i>Software Cost Estimation with COCOMO II</i> (Dr. Barry Boehm, et al) comportant la description comment appliquer le dernier modèle d'estimation. La plupart des concepts issus de <i>Software Engineering Economics</i> reste applicable pour les projets modernes.</p> <p>Mise à disposition de la version Costar 6.0 avec supports de COCOMO II.2000, MBASE, REVIC, et des contraintes de planification.</p> <p>2003 : Mise à disposition de la version Costar 7.0, supportant drag & drop, wizards, XP, import from USC tool, toolbars, and filters.</p> <p>Costar est le seul outil d'estimation qui supporte tous les modèles estimation COCOMO !</p>
BTD/GL/GP	3

CENTRALE L Y O N	<h2>Vue globale de COCOMO</h2>
Bertrand DAVID : Génie Logiciel	<ul style="list-style-type: none">• Le modèle d'estimation de coût COCOMO est utilisé par des milliers de chefs de projets informatiques et est basé sur l'étude de centaine de projets logiciels.• A la différence des autres modèle d'estimation, COCOMO est un modèle ouvert, Tous les détails sont publiés, y compris :<ul style="list-style-type: none">→ Les équations d'estimation de coût sous-jacentes,→ Toute hypothèse faite dans le modèle,→ Toute définition (comme la définition précise de chaque phase du projet),→ Les coûts se trouvant dans les estimations sont clairement décrits (gestion du projet est comptabilisé, pas la secrétaire).• Parce que COCOMO est bien défini, et parce que les algorithmes d'estimation ne sont pas propriétaires, les avantages de Costar sont les suivants :<ul style="list-style-type: none">→ Estimations COCOMO sont plus objectives et répétables que celles basées sur les méthodes et modèles propriétaires.→ COCOMO peut être calibré pour refléter les environnements de développements concrets et permet ainsi des estimations plus précises
BTD/GL/GP	4

CENTRALE L Y O N	<h1>COSTAR</h1>
Bertrand DAVID : Génie Logiciel	<ul style="list-style-type: none"> • Costar est une implémentation fidèle du modèle COCOMO qui est simple à utiliser dans les petits projets et assez puissant pour planifier et gérer des grands projets. • Typiquement on peut commencer avec une description très sommaire du système logiciel à développer et on peut utiliser Costar pour obtenir des estimations sommaires concernant planning et des effectifs, • Quand on raffine la connaissance du problème et sur le système à concevoir, il est possible d'utiliser Costar pour produire des estimations plus précises. • Costar permet de définir la structure du logiciel qui correspond aux besoins. Par exemple, l'estimation initiale peut être de 3000 lignes. La seconde peut être plus précise, tenant compte de découpage en deux sous-systèmes, car on a une meilleure idée des fonctionnalités et donc une vision plus précise du nombre de lignes de chaque sous-système. • La prochaine estimation peut prendre en compte les composants de chaque sous-système. Costar permet de continuer ce processus jusqu'à arriver au niveau de détails souhaité. • <i>Mot d'avertissement</i> : Il est simple d'utiliser Costar pour faire des estimations de coût, mais il est possible de mal les utiliser. Tout utilisateur de Costar doit consacrer un temps approprié pour apprendre les principes sous-jacents de COCOMO et les hypothèses et définition issues de <i>Software Engineering Economics</i> et de <i>Software Cost Estimation with COCOMO II</i>.
BTD/GL/GP	5

CENTRALE L Y O N	<h1>Introduction au modèle COCOMO</h1>
Bertrand DAVID : Génie Logiciel	<ul style="list-style-type: none"> • Le calcul fondamental du modèle COCOMO utilise l'Equation d'Effort pour estimer le nombre de Homme-Mois nécessaires pour développer le projet. Principaux autres résultats de COCOMO concernent les estimations d'identification des besoins et de Maintenance, qui sont dérivées de la qualité requise. • Lignes de code source (Source Lines of Code - SLOC) • Les calculs de COCOMO sont basés sur l'estimation de la taille du projet en lignes de code source (SLOC). SLOC est défini de la manière suivante : • Seulement les lignes de code source qui sont livrées dans le produit sont prises en compte – les moteurs de test et autres supports sont exclus • Les lignes de code sources sont produites par le personnel, les lignes créées par les générateurs sont exclus • Une SLOC est une ligne logique de code • Les déclarations sont comptabilisées comme SLOC • Les commentaires ne sont pas comptabilisés comme SLOC • Le modèle original COCOMO 81 a défini le terme de d'Instructions Sources Livrées (Delivered Source Instructions - DSI), qui est similaire de SLOC. La différence majeure entre DSI et SLOC est qu'une ligne de code source peut correspondre à plusieurs lignes physiques. Par exemple, un instruction "if-then-else" sera comptabilisée comme une SLOC, mais pourrait être comptabilisée comme plusieurs DSI.
BTD/GL/GP	6

CENTRALE L Y O N	<h2>Pilotes d'Echelle (The Scale Drivers)</h2>
Bertrand DAVID : Génie Logiciel	<ul style="list-style-type: none"> • En modèle COCOMO II plusieurs facteurs importants qui contribuent à l'estimation de la durée du projet et son coût sont les Pilotes d'échelle (Scale Drivers). On peut positionner chaque Pilote d'Echelle (Scale Drivers) pour déterminer l'exposant dans l'Equation d'Effort. • Les 5 Pilotes d'Echelle (Scale Drivers) sont : <ul style="list-style-type: none"> → Precedentedness (Caractère de nouveauté) → Development Flexibility (Flexibilité de développement) → Architecture / Risk Resolution (Risques architecturaux) → Team Cohesion (Cohésion de l'équipe) → Process Maturity (Maturité du processus) • Il faut noter que les Pilotes d'Echelle ont remplacé les Modes de Développement de COCOMO 81. Les deux premiers Pilotes d'Echelle (Precedentedness and Development Flexibility) décrivent sensiblement les mêmes influences que les Modes de Développements de l'époque.
BTD/GL/GP	7

CENTRALE L Y O N	<h2>Pilotes de Coût (Cost Drivers)</h2>
Bertrand DAVID : Génie Logiciel	<ul style="list-style-type: none"> • COCOMO II propose 17 pilotes de coût (Cost Drivers) – il faut estimer le projet, l'environnement de développement et l'équipe pour positionner chaque pilote de coût. • Les pilotes de coûts sont des facteurs multiplicateurs qui déterminent l'effort nécessaire pour terminer le projet logiciel. • Par exemple, si le projet concerne le développement d'un logiciel de contrôle de vol d'avion, il faut positionner le pilote de coût de la fiabilité requise (Required Software Reliability - RELY) à très important (Very High). Le taux correspondant à l'effort est multiplié par 1.26, exprimant que le projet aura besoin 26% de plus d'effort qu'un projet logiciel typique. • COCOMO II définit chaque Pilote de Coût et le Multiplicateur d'Effort associé à chaque indice. On peut consulter l'aide de Costar pour voir les détails des définitions et comment positionner les pilotes de coût.
BTD/GL/GP	8

CENTRALE L Y O N	<h2>Equation d'Effort de COCOMO II</h2>
Bertrand DAVID : Génie Logiciel	<p>Le modèle COCOMO II fournit les estimations d'Effort (mesurées en Homme-Mois HM) basé principalement sur l'estimation de la taille du projet logiciel (mesuré en milliers de SLOC, KSLOC):</p> $\text{Effort} = 2.94 * \text{EAF} * (\text{KSLOC})^E$ <p>Où</p> <p>EAF est le facteur d'ajustement d'effort (Effort Adjustment Factor) issu de Pilotes de Coût (Cost Drivers)</p> <p>E est un exposant dérivé de 5 Pilotes d'échelle (Scale Drivers)</p> <p>Par exemple : Un projet avec les Pilotes de Coût et les Pilotes d'échelle à la valeur nominale devrait avoir EAF à 1 et exposant E à 1.0997. Si on suppose que le projet va consister en 8 000 SLC, COCOMO II va estimer l'effort en HM à 28.9 avec la valuation suivante :</p> $\text{Effort} = 2.94 * (1.0) * (8)^{1.0997} = 28.9 \text{ Homme-Mois}$
BTD/GL/GP	9

CENTRALE L Y O N	<h2>Facteur d'Ajustement de l'Effort Effort Adjustment Factor</h2>
Bertrand DAVID : Génie Logiciel	<ul style="list-style-type: none"> Le Facteur d'Ajustement d'Effort (Effort Adjustment Factor) dans l'équation d'effort est simplement un produit de multiplicateur d'effort correspondant à tous les pilotes de coût du projet. Par exemple, si le projet est estimé de très grande complexité (l'effort multiplicateur de 1.34), et bas pour Langage et Expérience d'outils (l'effort multiplicateur de 1.09), et tous les autres Pilotes de Coût sont estimés à Nominal (l'effort multiplicateur de 1.00), le Facteur d'Ajustement d'Effort (EAF) du produit sera de $1.34 * 1.09$. $\text{Effort Adjustment Factor} = \text{EAF} = 1.34 * 1.09 = 1.46$ $\text{Effort} = 2.94 * (1.46) * (8)^{1.0997} = 42.3 \text{ Homme-Mois}$
BTD/GL/GP	10

CENTRALE L Y O N	Equation de durée de COCOMO II (Schedule Equation)
	<ul style="list-style-type: none"> L'équation de durée de COCOMO II (Schedule Equation) estime le nombre de mois nécessaire pour terminer le projet. La durée du projet est basée sur l'effort prévisionnel obtenu par l'équation de l'effort : $\text{Durée} = 3.67 * (\text{Effort})^{\text{SE}}$ <p>Où: Effort correspond à l'effort issu de l'équation d'effort de COCOMO II SE est l'exposant de l'équation de durée dérivé de cinq Pilotes d'Echelle</p> <p>En continuant l'exemple, et en substituant l'exposant 0.3179 qui est calculé à partir de Pilotes d'Echelle, on obtient l'estimation d'un an avec la moyenne du personnel entre 3 et 4 personnes :</p> <ul style="list-style-type: none"> $\text{Durée} = 3.67 * (42.3)^{0.3179} = 12.1 \text{ mois}$ $\text{Moyenne du personnel} = (42.3 \text{ HM}) / (12.1 \text{ Mois}) = 3.5 \text{ personnes}$
BTDAVID : Génie Logiciel	
BTD/GL/GP	11

CENTRALE L Y O N	Le Pilote de Coût (SCED)
	<ul style="list-style-type: none"> Le Pilote de Coût de COCOMO pour le Planning de Développement Nécessaire (Required Development Schedule SCED) est unique et nécessite les explications suivantes. <p>Le Pilote de Coût SCED est utilisé pour exprimer l'observation qu'un projet développé selon un planning plus rapide aura besoin de plus d'effort qu'un projet développé selon le planning optimal. Le coefficient SCED Très Bas correspond à un Multiplicateur d'Effort (Effort Multiplier) de 1.43 (en modèle COCOMO II.2000) qui exprime que si l'on envisage de terminer le projet en 75% du planning optimal (comme déterminé par le calcul COCOMO précédent) il faut augmenter les efforts à fournir. En continuant l'exemple précédent avec le Pilote de Coût (SCED) à Très Bas, COCOMO produit les estimations suivantes :</p> $\text{Durée} = 75\% * 12.1 \text{ Mois} = 9.1 \text{ Mois}$ $\text{Facteur d'ajustement d'Effort (Effort Adjustment Factor)} = \text{EAF} = 1.34 * 1.09 * 1.43 = 2.09$ $\text{Effort} = 2.94 * (2.09) * (8)1.0997 = 60.4 \text{ Homme-Mois}$ $\text{Moyenne du Personnel} = (60.4 \text{ HM}) / (9.1 \text{ Mois}) = 6.7 \text{ personnes}$ <p>A noter que le calcul de la durée n'est pas basé directement sur l'Effort (en HM) – à la place il est basé sur le planning qui serait nécessaire pour le projet effectué dans le planning nominal.</p> <ul style="list-style-type: none"> Ne pas oublier que Le Pilote de Coût SCED exprime "accélération par rapport au planning nominal". La commande Costar <i>Constraints Constrain Project</i> affiche une boîte de dialogue qui permet d'ajuster la durée par rapport à l'effort (SCED est alors positionné automatiquement). Il est possible d'utiliser cette boîte de dialogue pour contraindre le projet en fixant soit la durée, soit le coût (effort).
BTDAVID : Génie Logiciel	
BTD/GL/GP	12

CENTRALE
L Y O N

Bertrand DAVID : Génie Logiciel

BTD/GL/GP

Ada COCOMO (1/2)

- COCOMO a évolué et a été continuellement amélioré. Costar supporte le modèle traditionnel de COCOMO ainsi que les modèles plus récents.
- Le document TRW IOC Ada COCOMO: Definition and Refinements (Barry Boehm & Walker Royce, 1987, 1988) définit le modèle "Ada COCOMO".
- Chaque modèle comporte des Pilotes de Coût, des équations, des tables de distribution de phases, etc., et est sauvegardé dans un fichier comme modèle Costar (avec extension ".mdl"). On peut modifier tous les modèles Costar pour les faire correspondre à l'environnement utilisé. Costar 7.0 propose 13 modèles.
- Deux Pilotes de Coût : SECU (Classified Security Application) et RUSE (Required Reusability) ont été ajoutés au modèle Ada COCOMO.
- Le modèle appelé "ADA_87" considère que le langage de programmation Ada est utilisé. Utiliser Ada permet de développer plus facilement les systèmes de fiabilité élevée et de gérer des applications complexes, c'est la raison de nouveaux Pilotes RELY et CPLX.

BTD/GL/GP

13

CENTRALE
L Y O N

Bertrand DAVID : Génie Logiciel

BTD/GL/GP

Ada COCOMO

- Le modèle APM_88 est basé sur un "modèle du processus" différent appelé Ada Process Model. Les hypothèses suivantes sont prises en compte :
 - On utilise Ada pour produire des packages de spécifications vérifiés par le compilateur et permettant de produire Product Design Review (PDR)
 - Les équipes de conception de petite taille sont utilisées
 - Plus d'effort est consacré à l'analyse des besoins et la conception
 - Moins d'effort est consacré au codage, l'intégration et le test
- Le modèle APM_88 incorpore les changements d'équations, plusieurs pilotes de coût et une variété d'autres tables.
- La table suivante synthétise des différents modèles COCOMO qui sont utilisés dans Costar. Il est également possible d'utiliser ces outils pour créer des nouvelles versions de modèles COCOMO et de tables, qui sont calibrés aux environnements de développements particuliers
- Table : ModelAda
 Language?EquationsDescriptionCOCOMO_85NoTraditionalSoftware
 Engineering Economics + TURN VL, TOOL XHCOCOMO_87NoTraditionalNew
 VMVH, VMVT, SECU, RUSE, SCEDADA_87YesTraditionalNew RELY, CPLX,
 LEXPAPM_88YesAda ProcessNew MODP, ACAP, PCAP, sigma, tables...

BTD/GL/GP

14

CENTRALE L Y O N	<h2>Modèle de COCOMO Incrémental</h2>
Bertrand DAVID : Génie Logiciel	<ul style="list-style-type: none"> • Modèle développement incrémental COCOMO a été défini en même temps que le modèle COCOMO pour Ada. Le modèle COCOMO incrémental est une alternative moderne Incremental COCOMO is a modern alternative to the traditional Waterfall model of the software development process. • Incremental Development COCOMO lets you model a variety of development processes. Instead of modeling your software development as if it were a single effort devoted to inventing a single product, Incremental Development COCOMO lets you model development as a series of concurrent software projects, each yielding an intermediate product. • This strategy reduces your risk, and permits you to deliver an initial product to your customer earlier. • We've extended and generalized the definition so that Costar performs the calculations for increments with multiple components. • You can use Costar's worksheets to define which development phases are included in each increment, and how each increment is synchronized with the other increments. You can add delays between phases or between increments to match your schedule. • You can assign any component to any increment. • Incremental Development can be used with any COCOMO model, whether it's Ada COCOMO, or standard COCOMO. It can be used with Organic, Semidetached, or Embedded projects. fini
BTD/GL/GP	15

CENTRALE L Y O N	<h2>Points de Fonction</h2>
Bertrand DAVID : Génie Logiciel	<ul style="list-style-type: none"> • La méthodologie Points de Fonction a été développée par Allan Albrecht chez IBM. • Cette méthodologie est basée sur le constat que la taille du projet logiciel peut être estimée très tôt, pendant l'identification des besoins, en s'appuyant sur les Entrées et Sorties du système. Cinq Five classes d'éléments sont comptabilisées : <ul style="list-style-type: none"> → External Inputs → External Outputs → Logical Internal Files → External Interface Files → External Inquiries • Basé sur le décompte de chacun de ces éléments, de facteurs de poids associés et de facteurs d'ajustement que propose Albrecht, on peut calculer le total du compte de Points de fonction. • Costar convertit le compte de points de Fonction en un nombre équivalent de SLOC et l'utilise dans les équations COCOMO pour effectuer les estimations.
BTD/GL/GP	16