

CONDUITE DE PROJET LOGICIEL

ESTIMATION DES COUTS
DE DEVELOPPEMENT

Estimation des coûts

- Charge × Taux
- + Achats / approvisionnements
- + Frais de déplacements
- + Coûts indirects (si non inclus dans le taux)
- + Provisions pour aléas

→ BUDGET DU PROJET



Estimation de l'effort

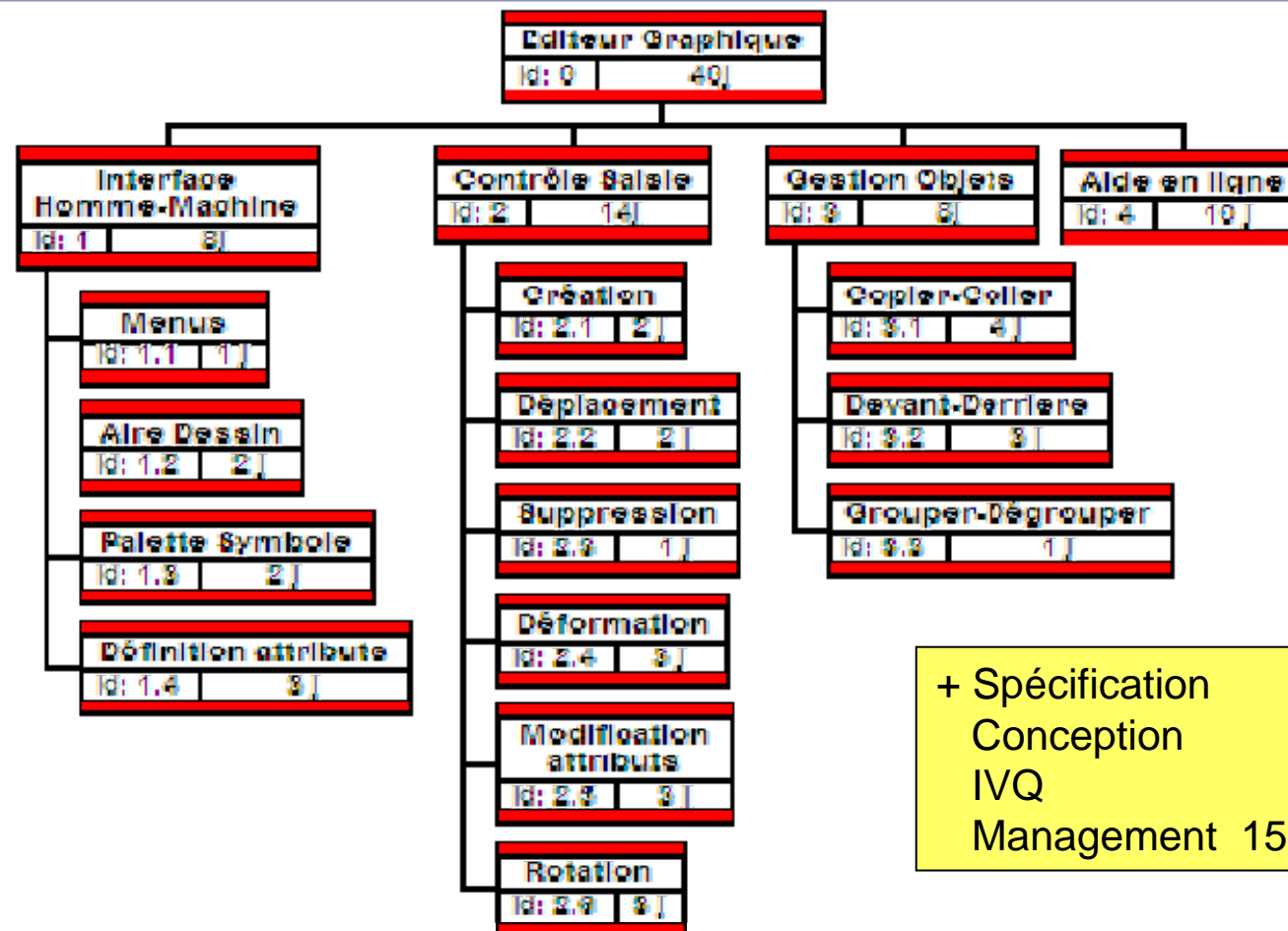
- L'estimation de l'effort (devis technique) est une activité critique dans un projet informatique .
- L'estimation est un préalable nécessaire à la planification.
- Le « jour×homme » (ou homme × jour) est l'unité de mesure de la charge de travail (ou effort) on trouve également «mois × homme » (HM) ou même « année × homme » pour les très grands projets
- Un projet estimé à 2000 jours*homme pourra être réalisé par :
 - 10 hommes en 1 an
 - 20 hommes en 6 mois
 - 5 hommes en 2 ans
 - mais certainement pas par 2000 hommes en 1 jour !!!



Certitudes

- Les programmes de grande taille ont un coût relatif supérieur à celui des petits programmes (loi des rendements décroissants)
 - La qualité et le coût de production d'un logiciel sont sujets à de grandes variances individuelles (facteurs humains)
 - Le coût et la récupération des erreurs représentent près de la moitié de l'effort total
 - Les problèmes de conception constituent la principale source d'erreurs
 - Dans les grands logiciels la majorité des erreurs est localisée dans un petit nombre de modules
 - La détection et la réparation précoces des erreurs
 - La prévention des fautes
 - L'utilisation des langages évolués
 - Les environnements intégrés de programmation
- } augmentent la productivité

Approche analytique



+ Spécification	10 %
Conception	10 %
IVQ	15 %
Management	15 %

Types de modèles

- **MODELES DE REGRESSION**

- Relation mathématique entre paramètres d'entrée et de sortie
- Calibrage pour un environnement donné (à partir de projets antérieurs)

Ex: Halstead, cocomo de Boehm

- **MODELES HEURISTIQUES**

- Découpage du projet en unités élémentaires pour lesquels le retour d'expérience fournit des ratios de productivité

Ex: FPA

- **MODELES PHENOMENOLOGIQUES**

- Description d'un phénomène général et application à des phénomènes particuliers (en l'occurrence le développement du logiciel)

Ex: Putnam, Norden-Rayleigh

Modèles de régression

Calcul de l'effort et du délai a partir d'une donnée unique

Taille du logiciel = Nombre de milliers de lignes de code à produire (KISL)

CALCUL DE L'EFFORT (en MOIS-HOMMES)

$$\text{EFFORT} = A \times (\text{Taille})^B$$

CALCUL DU DELAI (en MOIS)

$$\text{DELA I} = C \times (\text{Effort})^D$$

Modèles de régression: Equations de calcul d'effort

WATSON-FELIX 77.....	$HM = 5.2 \times (Taille)^{0.91}$
NELSON 78	$HM = 4.9 \times (Taille)^{0.98}$
FREBERGER-BASILI 79	$HM = 1.48 \times (Taille)^{1.02}$
BOEHM 81 (Modèle1)	$HM = 2.4 \times (Taille)^{1.05}$
BOEHM 81 (Modèle2)	$HM = 3.0 \times (Taille)^{1.12}$
FREDERIC 74.....	$HM = 2.43 \times (Taille)^{1.18}$
BOEHM 81 (Modèle3)	$HM = 3.6 \times (Taille)^{1.20}$
PHISTER 79.....	$HM = 0.99 \times (Taille)^{1.275}$
JONES 77	$HM = 1.0 \times (Taille)^{1.40}$
HALSTEAD 77	$HM = 0.70 \times (Taille)^{1.50}$

Modèles de régression: Equations de calcul de delai

WALSTON-FELIX 77..... $TDEV = 2.47 \times (\text{Effort})^{0.35}$

BOEHM 81 Modèle 1 $TDEV = 2.5 \times (\text{Effort})^{0.38}$

BOEHM 81 Modèle 2 $TDEV = 2.5 \times (\text{Effort})^{0.35}$

BOEHM 81 Modèle 3 $TDEV = 2.5 \times (\text{Effort})^{0.32}$

Le modèle COCOMO

COConstructive COst MOdel

- Modèle de régression pour calculer l'effort et les délais de développement
- Méthode heuristique de répartition des charges et délais par phase.
- **Première version en 1981 (64 projets)**
- **Mise à jour pour ADA (1987)**
- **Actualisation pour nouveaux cycles de vie (COCOMO 2.0/1998)**
- Activités couvertes:
 - Spécifications
 - Conception générale et détaillée
 - Réalisation
 - Tests
 - Management
 - Gestion des configurations
 - assurance qualité
 - Documentation
- Nécessité d'une base de référence



Barry BOEHM

COCOMO 81

3 MODELES:

● MODELE BASIC

- Calcul des charges et des délais
- Répartition par phases
- Seul paramètre = la taille du logiciel

● MODELE INTERMEDIAIRE

- Calcul des charges et des délais
- Répartition par phases
- 2 paramètres = taille du logiciel + facteurs d'ajustement

● MODELE DETAILLE

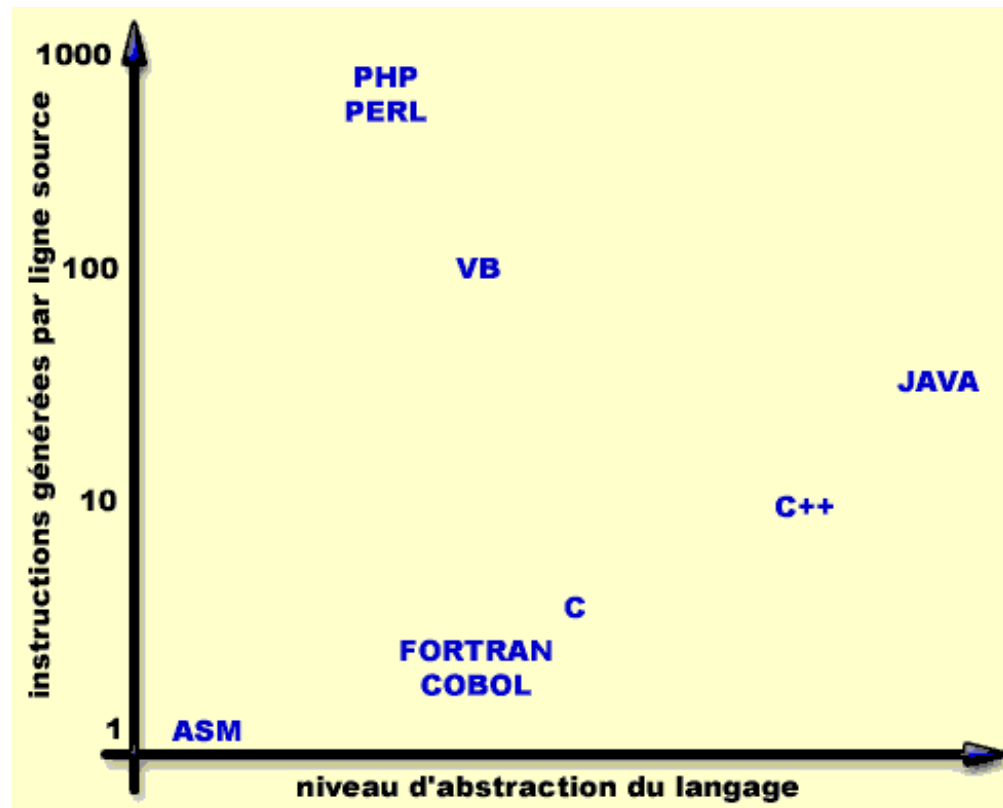
- Même procédé d'évaluation que le modèle intermédiaire avec application des facteurs d'ajustement pour chaque phase du projet

Hypothèses et définitions

- La spécification est validée
- 1 Mois-Homme = 152 heures (hors congés)
- Unité de mesure de la taille du logiciel =
Ligne d 'Instruction Source Livrable Produite (ISL ou DSI)
= ligne d 'instruction(s) figurant dans un programme créé par le personnel du projet, traduite en code exécutable selon un quelconque procédé de génération (générateur, pré-processor, compilateur, etc.) pour être intégré à l 'un des composants du produit livré.
Exemples : 1 ligne composée de 3 instructions = 1 ISL
 3 lignes de déclaration de données = 3 ISL
- KISL = Millier d 'instructions source Livrables

Influence du langage

Un programmeur écrit le même nombre de lignes de code par jour quel que soit le langage utilisé.



Comptage des lignes d'instructions

comptabilisé

➤ Type Instruction

- instruction exécutable
- déclaration
- directives de compilation

➤ Procédé de production

- programmeur
- traducteur automatique
- copie ou réutilisation sans modif.
- modification

➤ Origine

- nouveau développement
- version précédente modifiée
- composant ou librairie réutilisable

non comptabilisé

- commentaire

- générateur de code source

- suppression

- composant sur étagère (COTS)
- composant fourni (GFS)
- librairie ou produit du commerce
- librairie ou produit sous-traité

COCOMO 81

3 TYPES DE PROJETS

● ORGANIQUE

- Equipes restreintes dans un environnement familier et stable
- Domaine d 'application et technologies connues
- Développement autonome ou peu dépendant
- Taille relativement petite

● SEMI-DETACHE

- Niveau intermédiaire

● IMBRIQUE

- Fortes contraintes d 'environnement
- Logiciel intégré dans un système complexe
- Exigences mal connues et/ou évolutives
- Montage industriel complexe
- Modules critiques
- Montée en charge brutale

Basic COCOMO: Calcul de l'effort

Charge exprimée en Homme_Mois (HM) (*)

• **Mode Organique**

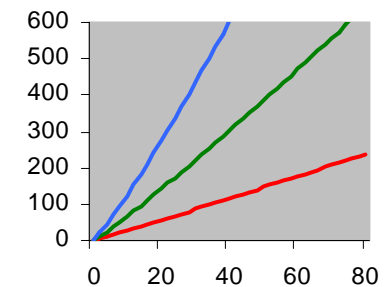
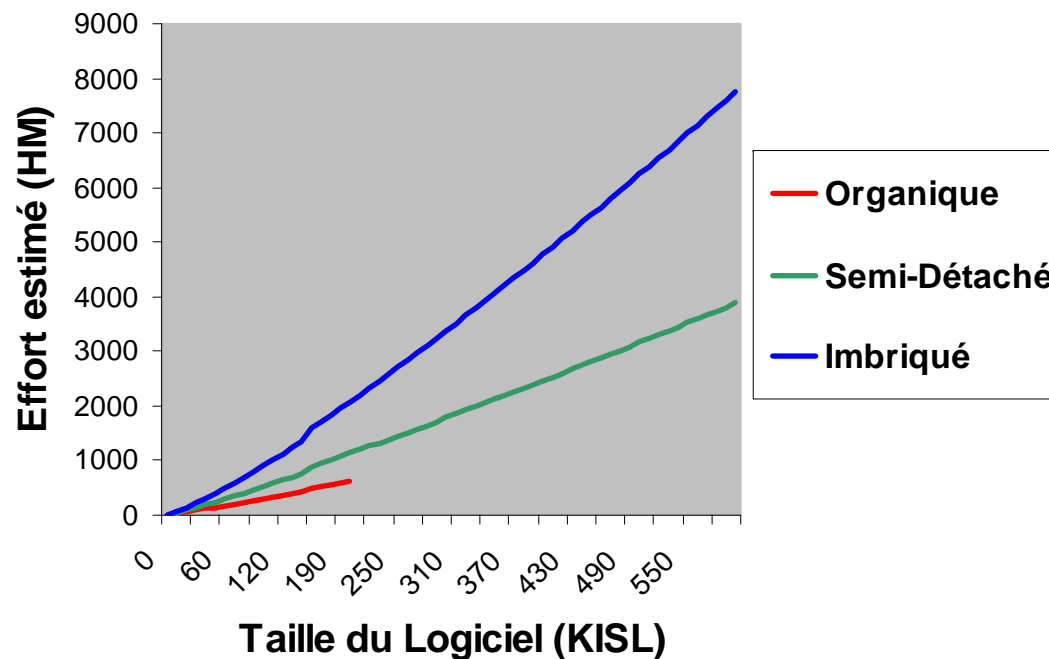
$$HM = 2,4 \times KISL^{1,05}$$

• **Mode Semi-détaché**

$$HM = 3,0 \times KISL^{1,12}$$

• **Mode Imbriqué**

$$HM = 3,6 \times KISL^{1,20}$$



(*) Charge calculée sur la base de 152 heures /mois (modele americain) modele français => HM x 152/146

Basic COCOMO: Calcul du délai

Temps de développement (TDEV) exprimé en Mois

• **Mode Organique**

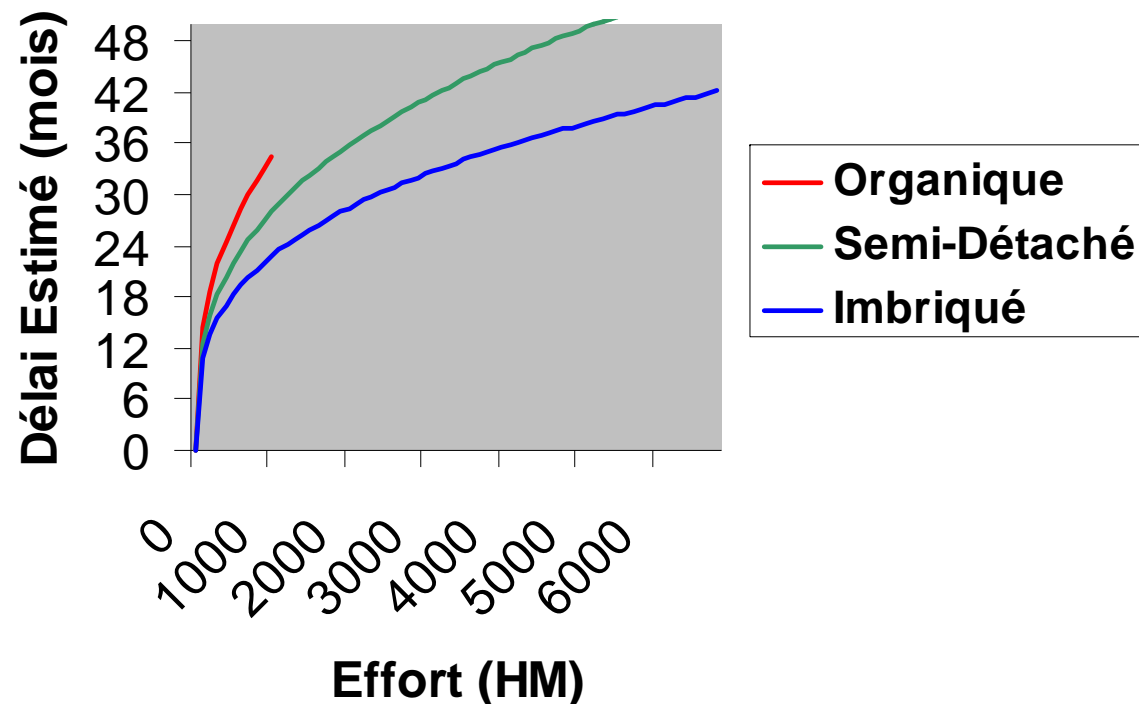
$$\text{TDEV} = 2,5 \times \text{HM} \times 0,38$$

• **Mode Semi-détaché**

$$\text{TDEV} = 2,5 \times \text{HM} \times 0,35$$

• **Mode Imbriqué**

$$\text{TDEV} = 2,5 \times \text{HM} \times 0,32$$



Basic COCOMO:

Répartition de l'effort par phase

Répartition de l'effort		Taille				
Mode	Phase	Petit 2KISL	Intermédiaire 8KISL	Moyen 32KISL	Grand 128 KISL	Très Grand 512 KISL
<u>Organique</u>	Plans & Spécifications	+6%	+6%	+6%	+6%	
	Conception Préliminaire	16%	16%	16%	16%	
	Réalisation	68%	65%	62%	59%	
	Conception Détaillée	26%	25%	24%	23%	
	Codage et tests unitaires	42%	40%	38%	36%	
	Intégration & Tests	16%	19%	22%	25%	
<u>Semi-détaché</u>	Plans & Spécifications	+7%	+7%	+7%	+7%	+7%
	Conception Préliminaire	17%	17%	17%	17%	17%
	Réalisation	64%	61%	58%	55%	52%
	Conception Détaillée	27%	26%	25%	24%	23%
	Codage et tests unitaires	37%	35%	33%	31%	29%
	Intégration & Tests	19%	22%	25%	28%	31%
<u>Imbriqué</u>	Plans & Spécifications	+8%	+8%	+8%	+8%	+8%
	Conception Préliminaire	18%	18%	18%	18%	18%
	Réalisation	60%	57%	54%	51%	48%
	Conception Détaillée	28%	27%	26%	25%	24%
	Codage et tests unitaires	32%	30%	28%	26%	24%
	Intégration & Tests	22%	25%	28%	31%	34%

Basic COCOMO:

Répartition des délais par phase

Répartition des délais		Taille				
Mode	Phase	Petit 2KISL	Intermédiaire 8KISL	Moyen 32KISL	Grand 128 KISL	Très Grand 512 KISL
<u>Organique</u>	Plans & Spécifications	+10%	+11%	+12%	+13%	
	Conception Préliminaire	19%	19%	19%	19%	
	Réalisation	63%	59%	55%	51%	
	Intégration & Tests	18%	22%	28%	30%	
<u>Semi-détaché</u>	Plans & Spécifications	+16%	+18%	+20%	+22%	+24%
	Conception Préliminaire	24%	25%	26%	27%	28%
	Réalisation	56%	52%	48%	44%	40%
	Intégration & Tests	20%	23%	26%	29%	32%
<u>Imbriqué</u>	Plans & Spécifications	+24%	+28%	+32%	+36%	+40%
	Conception Préliminaire	30%	32%	34%	36%	38%
	Réalisation	48%	44%	40%	36%	32%
	Intégration & Tests	22%	24%	26%	28%	30%

Prise en compte de la réutilisation

Le logiciel réutilisé ne fait pas partie du volume de code à produire et ses lignes d'instructions ne doivent pas être comptabilisées dans l'évaluation de la taille du produit à développer.

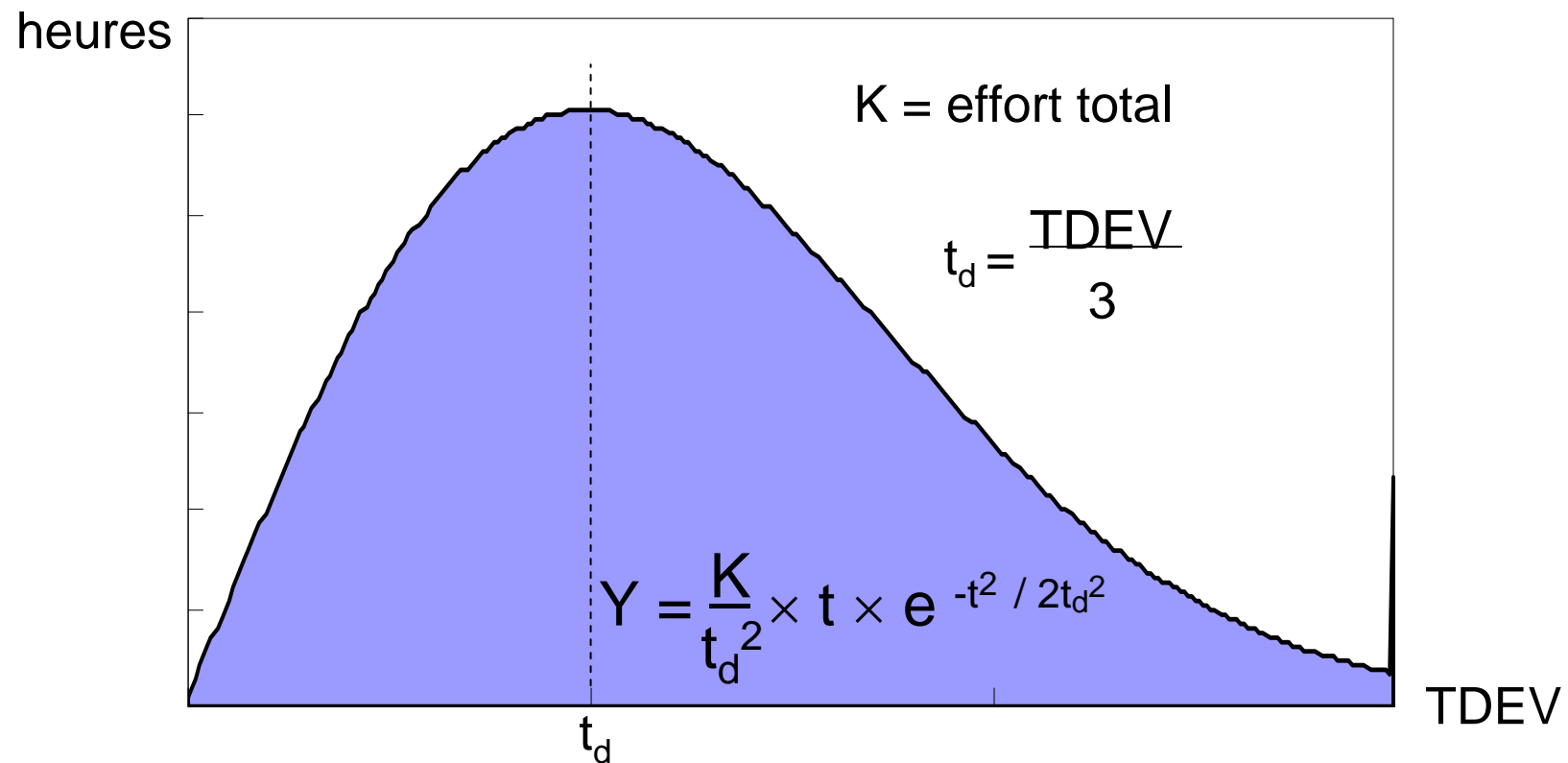
Mais ...

PEUT-ON CONSIDERER QUE LA REUTILISATION EST « GRATUITE » ?

- **Ne faudra-t-il pas modifier ou enrichir la conception pour atteindre les objectifs particuliers du produit à développer?**
- **Ne faudra-t-il pas reprendre certaines sections de code pour les adapter aux particularités de l'environnement cible?**
- **Ne faut-il pas intégrer et tester le module réutilisé avec les autres modules développés?**

En pratique, utilisation d'un coefficient de 0.3 à 0.7 dans les phases de conception, réalisation, intégration et tests.

Modèle de Putnam: Courbe d'effort instantané



Ajustement de l'effectif

- POURQUOI

- Pour réduire le délai (+ de personnes)
- Parce que l'effectif disponible est différent de l'effectif calculé (- de personnes)

Attention à la loi de Brook : doubler le nombre de programmeurs sur un projet en retard ne fait que doubler le retard.

- INCIDENCE

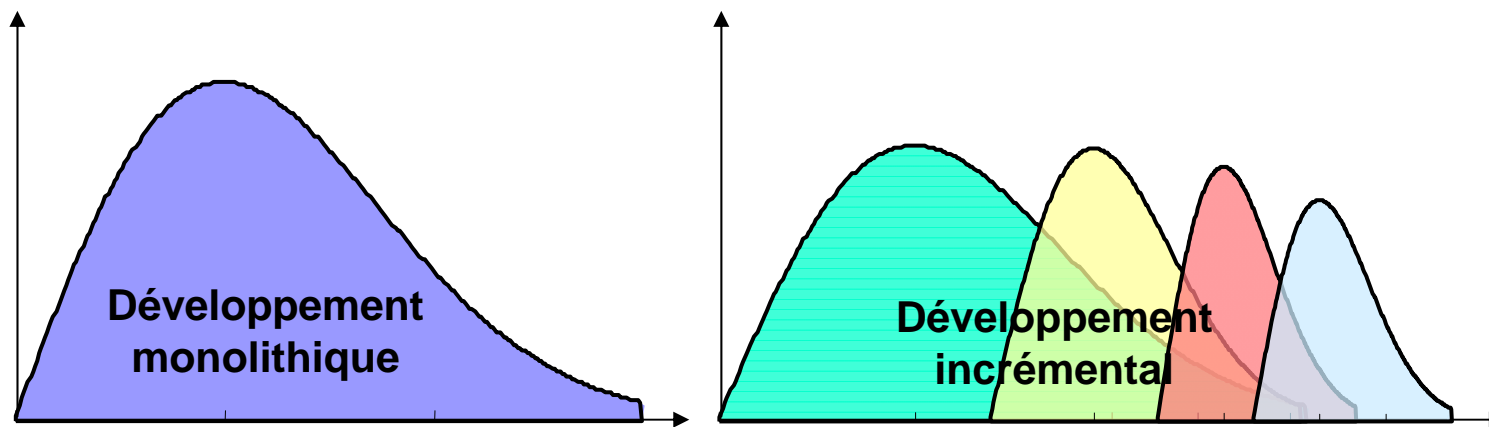
- \uparrow Effectif $\Rightarrow \uparrow$ Charge (Personnes x 2 \Rightarrow 25% de charge en +)
- \downarrow Effectif $\Rightarrow \downarrow$ Charge (Personnes / 2 \Rightarrow 20% de charge en -)

$$\text{Charge Réelle} = \text{Charge Estimée} \times \left(0.8^{\log_2 \frac{\text{Effectif calculé}}{\text{Effectif réel}}} \right)$$

$$\text{Délai} = \frac{\text{Charge Réelle}}{\text{Effectif Réel}}$$

Application au processus incrémental

LISSAGE DE LA CHARGE



Avantage:

- Optimisation de l'utilisation des ressources

Inconvénient:

- Facteur de cassure du logiciel [*Software Breakage Factor*]

Evaluation de chaque incrément en considérant la réutilisation du logiciel développé à l'occasion des précédentes itérations

Principes du modèle intermédiaire

- Même méthode de calcul que pour le modèle basic avec d'autres équations:
 - **Mode Organique** $HM = 3,2 \times KISL^{1,05}$
 - **Mode Semi-détaché** $HM = 3,0 \times KISL^{1,12}$
 - **Mode Imbriqué** $HM = 2,8 \times KISL^{1,20}$
- Pondération de la charge par des facteurs d'ajustement

Facteurs d'ajustement (COCOMO 81)

FACTEURS		Coefficient					
		Très faible	Faible	Nominal	Fort	Très Fort	Extra Fort
Caractéristiques du Produit							
RELY	Fiabilité requise	0.75	0.88	1.00	1.15	1.40	
DATA	Volume de données à gérer		0.94	1.00	1.08	1.16	
CPLX	Complexité du produit	0.70	0.85	1.00	1.15	1.30	1.65
Caractéristiques de la machine							
TIME	Contraintes de temps d'exécution			1.00	1.11	1.30	1.66
STOR	Contraintes d'occupation mémoire			1.00	1.06	1.21	1.56
VIRT	Stabilité de la plate-forme cible		0.87	1.00	1.15	1.30	
TURN	Disponibilité de la plate-forme de développement		0.87	1.00	1.07	1.15	
Caractéristiques du personnel							
ACAP	Compétence des analystes	1.46	1.19	1.00	0.86	0.71	
AEXP	Expérience du domaine d'application	1.29	1.13	1.00	0.91	0.82	
PCAP	Compétence des programmeurs	1.42	1.17	1.00	0.86	0.70	
VEXP	Expérience de la machine virtuelle	1.21	1.10	1.00	0.90		
LEXP	Expérience du langage de programmation	1.14	1.07	1.00	0.95		
Caractéristiques du projet							
MODP	Utilisation de techniques modernes	1.24	1.10	1.00	0.91	0.82	
TOOL	Utilisation d'outils de développement	1.24	1.10	1.00	0.91	0.83	
SCED	Exigences du planning de développement	1.23	1.08	1.00	1.04	1.10	

Intermediate COCOMO: méthode

- **Calcul du facteur global d'ajustement**

- Facteur global d'ajustement = produit des 15 coefficients de facteurs élémentaires

- **Calcul de la charge**

- Charge ajustée = charge estimée x facteur global d'ajustement

- **Calcul des délais**

- Mêmes formules que pour le modèle basic

- **Répartition des charges et délais par phases**

- Mêmes tables que pour le modèle basic

Detailed COCOMO

- Certains facteurs tels que la fiabilité, l'expérience du domaine d'application ou du langage de programmation sont plus sensibles sur certaines phases du développement.

☞ **Le modèle détaillé fournit, pour chaque facteur d'ajustement, un ensemble de coefficients différents en fonction de la phase**

- Dans le modèle intermédiaire, les facteurs d'ajustement peuvent être appliqués soit globalement pour l'ensemble du logiciel, soit composant par composant.

☞ **Le modèle détaillé permet de prendre en compte les facteurs au niveau approprié en proposant une hiérarchie à 3 niveaux (système/ sous-système/composant élémentaire) avec possibilité «d'ajuster» à chacun des niveaux.**

☞ **Les équations sont identiques à celle du modèle intermédiaire.**

COCOMO 2: Objectifs

Tenir compte ...

- des nouveaux marchés et des avancées du génie logiciel (Génération d'applications, intégration de COTS, méthodes agiles...)
- de l'information rendue disponible au fur et à mesure de l'évolution du processus de développement (complexité, taille du logiciel, risques, etc.)
- de la stratégie particulière choisie pour conduire le processus de développement d'un projet .

Les 3 modèles d'estimation de COCOMO II

- Le modèle « Elaboration d'application »
(Premières phases du cycle, Prototypage,...)
- Le modèle « Conception primitive »
(Ebauche d'architecture, Premiers incréments, ...)
- Le modèle « Post-Architecture »
(Architecture définie, Début de développement,...)

Principes de base

$$\text{Effort} = A \times (\text{Taille})^B$$

A = Paramètre de productivité (constante représentative de l'effort requis pour chaque ligne produite)

B = Paramètre d'échelle (permet de traduire l'économie d'échelle / la croissance exponentielle de l'effort en fonction de la taille)

$$B = 0.91 + 0.01 \sum W_i$$

(W_i = facteurs d'échelle)

Facteurs d'échelle



Facteurs d'échelle (W _i)	Très Bas (5)	Bas (4)	Nominal (3)	Haut (2)	Très Haut (1)	Extra Haut (0)
Expérience du domaine et des technologies PREC	totalemt nouveau	en grande partie nouveau	quelques originalités	plutôt familier	bien connu	parfaitement connu
Flexibilité par rapport au modèle de développ ^t FLEX	rigoureux	libertés occasionnelles	quelques libertés	Globalement conforme	A peu près conforme	objectifs généraux
Résolution des risques identifiés en conception RESL	rarement (20%)	parfois (40%)	souvent (60%)	généralement (75%)	la plupart du temps (90%)	toujours (100%)
Cohésion de l'équipe / relations humaines TEAM	relations très difficiles	quelques relations difficiles	relations plutôt coopératives	relations coopératives	relations fortement coopératives	comme si il n'y avait pas de relation
Maturité du processus (mesure CMM) PMAT	Moyenne pondérée des " oui " aux réponses du questionnaire de maturité CMM					

Facteurs d'échelle: valeurs

Scalefactor Paramters						
	VLO	LO	NOM	HI	VHI	XHI
PREC	6.20	4.96	3.72	2.48	1.24	0.00
FLEX	5.07	4.05	3.04	2.03	1.01	0.00
RESL	7.07	5.65	4.24	2.83	1.41	0.00
TEAM	5.48	4.38	3.29	2.19	1.10	0.00
PMAT	7.80	6.24	4.68	3.12	1.56	0.00

Facteurs d'ajustement avec COCOMO II

Caractéristiques du Produit

	RELY	Fiabilité requise	}RCPX	Fiabilité et complexité du produit
	DATA	Volume de données à gérer			
	CPLX	Complexité du produit			
	DOCU	Adéquation de la doc avec le besoin			
	RUSE	Exigences de réutilisation	RUSE	

Caractéristiques de la machine

	TIME	Contraintes de temps d'exécution	}PDIF	Difficultés de la plate-forme
	STOR	Contraintes d'occupation mémoire			
	VIRT	Stabilité de la plate-forme cible			
	TURN	Disponibilité de la plate-forme de devpt			
	PVOL	Volatilité de la plate-forme			

Caractéristiques du personnel

	ACAP	Compétence des analystes	}PERS	Aptitudes du personnel
	PCON	Renouvellement du personnel			
	PCAP	Compétence des programmeurs	}PEXP	Expérience du personnel
	AEXP	Expérience du domaine d'application			
	VEXP	Expérience de la machine virtuelle			
	LEXP	Expérience du langage de programmation			

Caractéristiques du projet

	MODP	Utilisation de techniques modernes	}PEXP	Expérience du personnel
	TOOL	Utilisation d'outils de développement			
	SITE	Développement sur plusieurs sites			
	SCED	Exigences du planning de développement	SCED	

Facteurs d'ajustement du modèle «conception primitive»

Early Design Paramters							
	XLO	VLO	LO	NOM	HI	VHI	XHI
RCPX	0.73	0.81	0.98	1.00	1.30	1.74	2.38
RUSE	XXXX	XXXX	0.95	1.00	1.07	1.15	1.24
PDIF	XXXX	XXXX	0.87	1.00	1.29	1.81	2.61
PERS	2.12	1.62	1.26	1.00	0.83	0.63	0.50
PREX	1.59	1.33	1.12	1.00	0.87	0.71	0.62
FCIL	1.43	1.30	1.10	1.00	0.87	0.73	0.62
SCED	XXXX	1.43	1.14	1.00	1.00	1.00	XXXX

Facteurs d'ajustement du modèle « post-architecture »

Product Parameters						
	VLO	LO	NOM	HI	VHI	XHI
RELY	0.82	0.92	1.00	1.10	1.26	XXXX
DATA	XXXX	0.90	1.00	1.14	1.28	XXXX
DOCU	0.81	0.91	1.00	1.11	1.23	XXXX
CPLX	0.73	0.87	1.00	1.17	1.34	1.74
RUSE	XXXX	0.95	1.00	1.07	1.15	1.24

Personnel Parameters						
	VLO	LO	NOM	HI	VHI	XHI
ACAP	1.42	1.19	1.00	0.85	0.71	XXXX
AEXP	1.22	1.10	1.00	0.88	0.81	XXXX
PCAP	1.34	1.15	1.00	0.88	0.76	XXXX
PEXP	1.19	1.09	1.00	0.91	0.85	XXXX
LTEX	1.20	1.09	1.00	0.91	0.84	XXXX
PCON	1.29	1.12	1.00	0.90	0.81	XXXX

Platform Parameters						
	VLO	LO	NOM	HI	VHI	XHI
TIME	XXXX	XXXX	1.00	1.11	1.29	1.63
STOR	XXXX	XXXX	1.00	1.05	1.17	1.46
PVOL	XXXX	0.87	1.00	1.15	1.30	XXXX

Project Parameters						
	VLO	LO	NOM	HI	VHI	XHI
TOOL	1.17	1.09	1.00	0.90	0.78	XXXX
SCED	1.43	1.14	1.00	1.00	1.00	XXXX
SITE	1.22	1.09	1.00	0.93	0.86	0.80

Exigences de délai

	Très serré	Serré	Nominal	Lâche	Très lâche
SCED	75% of nominal	85%	100%	130%	160%
	1.43	1.14	1.00	1.00	1.00

$$\text{TDEV} = [3.0 \times (\text{Effort})^{(0.28 + 0.2 \times (B - 0.91))}] \times \frac{\% \text{ SCED}}{100}$$

$$\begin{aligned} B &= \text{Paramètre d'échelle} \\ &= 0.91 + 0.01 \sum W_i \end{aligned}$$

Méthodes d'évaluation de taille

- Les points d'objets
- Les points de fonctions
- Le nombre de lignes de code

Evaluation par points d 'objets

OBJET := ECRAN | RAPPORT | COMPOSANT L3G

Avantages:

- Simple, Rapide et Facile à mettre en œuvre
- Orienté « utilisateur final »
- Ne nécessite ni référentiel ni calibrage (pas de lignes de code)
- Utilisable en tout début de projet
- Prend en compte globalement l 'environnement de développement (utilisation d 'outils et d 'ateliers de Génie Logiciel)
- Assez bien adapté aux développement RAD

Inconvénients:

- Evaluation « grossière »
- Très forte incertitude relative (Optimiste= 50%, Pessimiste = 200%)

Points d 'objets: évaluer la difficulté du logiciel

- 1) Dénombrer les objets
- 2) Evaluer la complexité des objets

Ecrans				Rapports			
Nombre de vues incluses	nombre et origine des données			Nombre de sections incluses	nombre et origine des données		
	Total < 4 (< 2 srv. < 3 clnt)	Total < 8 (< 2-3 srv < 3-5 clnt)	Total >=8 (>3 srv >5 clnt)		Total < 4 (< 2 srv. < 3 clnt)	Total < 8 (< 2-3 srv < 3-5 clnt)	Total >=8 (>3 srv >5 clnt)
<3	simple	simple	moyen	0 ou 1	simple	simple	moyen
entre 3 et 7	simple	moyen	difficile	2 ou 3	simple	moyen	difficile
> 8	moyen	difficile	difficile	> 3	moyen	difficile	difficile

- 3) Pondérer les nombres d'objets en tenant compte de leur complexité

Type d'objet	Pondération de complexité		
	Simple	Moyen	Difficile
Ecran	1	2	3
Rapport	2	5	8
Composant L3G			10

Points d 'objets: convertir en effort

- 4) Totaliser les points d 'objets
- 5) Estimer le pourcentage de réutilisation et calculer les points d 'objets nouveaux (PON)

$$\text{PON} = (\text{Points d 'Objets}) \times (100 - \% \text{ réutilisation}) / 100$$

- 6) Déterminer le taux de productivité (PROD)

Expérience et aptitudes des développeurs	Très bas	Bas	Nominal	Haut	Très haut
Maturité et efficacité de l'environnement de développement	Très bas	Bas	Nominal	Haut	Très haut
PROD	4	7	13	25	50

- 7) Calculer l 'effort

$$\text{HM} = \text{PON} / \text{PROD}$$

Evaluation par points de fonction

- Détermination d'un nombre de points caractéristiques de la taille et de la complexité du logiciel
- Valorisation des points en lignes de code par application d'un ratio de productivité dépendant du langage de programmation

Avantages:

- Facile et simple
- Permet d'estimer la taille du logiciel en fonction des caractéristiques externes du logiciel (plus facile)
- Permet d'appliquer les modèles de régression, les facteurs d'ajustement et le modèle de réutilisation

Inconvénients:

- Evaluation heuristique de la taille
- Incertitude relative importante (Optimiste= 67%, Pessimiste = 150%)

Points de fonction: caractériser le système logiciel

Quantifier les fonctions de traitement de l'information

Entrée Externe (Inputs)	Comptabiliser tous les types de données ou de commandes fournis par un utilisateur (i) dont l'origine est au delà des frontières du système logiciel à évaluer et (ii) qui ajoute ou modifie des données dans un fichier logique interne du système.
Sortie Externe (Outputs)	Comptabiliser tous les types de données ou de contrôles destinés à un utilisateur et issus de l'intérieur du système logiciel devant être mesuré.
Fichier Logique Interne (Files)	Comptabiliser tous les principaux groupes logiques de données ou d'information de contrôle du système. Prendre en compte chaque fichier logique (par exemple, chaque groupe logique de données) qui est produit, utilisé, ou mis à jour par le système.
Fichiers d'interface Externe (Interfaces)	Comptabiliser les fichiers échangés ou partagés avec d'autres systèmes logiciels.
Requête Externe (Queries)	Comptabiliser toutes les combinaisons d'entrée-sortie, où une donnée d'entrée provoque la fourniture immédiate d'un résultat.

Points de fonction: évaluer la complexité

Pour Fichiers internes et interfaces				Pour sorties et requêtes				Pour entrées			
# types	# éléments d'Informations			# types	# éléments d'Informations			# types	# éléments d'Informations		
	1 - 19	20 - 50	51+		1 - 5	6 - 19	20+		1 - 4	5 - 15	16+
1	Bas	Bas	Moyen	0 ou 1	Bas	Bas	Moyen	0 ou 1	Bas	Bas	Moyen
2 - 5	Bas	Moyen	Haut	2 - 3	Bas	Moyen	Haut	2 - 3	Bas	Moyen	Haut
6+	Moyen	Haut	Haut	4+	Moyen	Haut	Haut	3+	Moyen	Haut	Haut

Type de fonction	Pondération de complexité		
	Bas	Moyenne	Haut
Fichiers Logiques Internes	7	10	15
Fichiers Interfaces Externes	5	7	10
Entrées Externes	3	4	6
Sorties Externes	4	5	7
Requêtes Externes	3	4	6

Points de fonction: convertir les points en lignes

Langage	ISL / Points
Ada	71
AI Shell	49
APL	32
Assembleur	320
Assembleur (Macro)	213
ANSI/Quick/Turbo Basic	64
Basic- Compilé	91
Basic - Interprété	128
C	128
C++	29
ANSI COBOL 85	91
Fortran 77	105
Forth	64
Jovial	105
Lisp	64
Modula 2	80
Pascal	91
Prolog	64
Générateur de rapport	80
Tableur / L4G	6

1. Pondérer les fonctions par rapport à leur complexité.

$$P(\text{type}) = \text{poids} \times \text{nombre d'unités du type}$$

2. Calculer le nombre brut de points de fonctions.

$$N = \sum P(\text{type})$$

3. Convertir les points de fonctions en nombre de lignes de code source en tenant compte du langage de programmation

Evaluation de taille à partir d 'un modèle UML

1) DIMENSIONNEMENT DU SYSTEME par rapport aux

- **Applications** *Sous-système logiciel dédié à la réalisation d 'une activité du domaine « métier » (collections de classes qui collaborent pour la réalisation d 'un ensemble de cas d 'utilisation associés)*
- **Classes** *Abstractions fondamentales propres au domaine « métier » (encapsulation des données « métier »)*
- **Use Cases** *Caractéristiques du système perceptibles pour l 'utilisateur final et décrivant ses fonctionnalités et son comportement.*
- **Packages** *Elément structurel auquel est associé une responsabilité clairement définie dans l 'architecture du logiciel.*
- **Composants** *Abstractions fondamentales assurant la fourniture des services spécialisés.*
- **Services** *Caractéristiques fonctionnelles du système communes à plusieurs applications.*

2) QUANTIFICATION (nombre de lignes = f(dimension, langage))

Outil (1)

Qualifier Summary

Complexity | Genericity | Reuse

☐ Project ☒ Manual

	Trivial		Simple		Medium		Difficult		Complex		
Business											
Applications	0	0	0	0	100	100	0	0	0	0	= 100%
Classes	0	10	0	20	100	50	0	10	0	10	= 100%
Use Cases	0	0	0	40	100	40	0	10	0	10	= 100%
Infrastructure											
Packages	0	0	0	0	100	100	0	0	0	0	= 100%
Components	0	0	0	0	100	100	0	0	0	0	= 100%
Services	0	0	0	0	100	100	0	0	0	0	= 100%

OK Cancel Apply

Outil (2)

The screenshot shows the 'SELECT Estimator - [Estimator1*]' window. It has a menu bar with 'File', 'Data', 'Estimate', 'Tools', and 'Help'. The interface is divided into two main sections: 'Data Summary' on the left and 'Estimate' on the right.

Data Summary Section:

- Scope:** Radio buttons for 'Project' and 'Manual' (selected).

Applications:	2	Packages:	0
Classes:	18	Components:	2
Use Cases:	20	Services:	0
- Qualifiers:** Radio buttons for 'Project' and 'Manual' (selected).

Complexity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Genericity	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Reuse	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

% Complexity	T	S	M	D	C
Applications	0	0	100	0	0
Classes	10	20	50	10	10
Use Cases	0	40	40	10	10
Packages	0	0	100	0	0
- Technology:** Radio buttons for 'Project' and 'Manual' (selected).

% Language	Classes	Components
Unspecified	0	0
C++	75	100

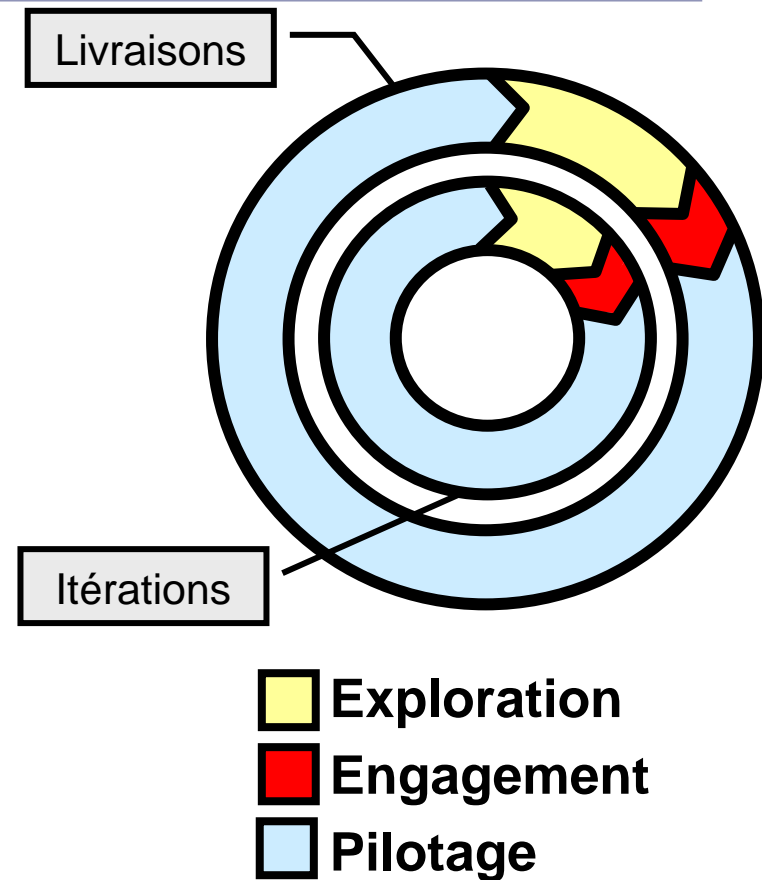
Estimate Section:

- Activity Profile:** A table showing effort for various activities.

Activity Profile	Effort
Planning	51
Analysis	92
Design	77
Programming	93
Testing	72
Integration	50
Review	25
Total person days effort	460
- Team:** A button.
- Schedule:** 'Duration (months)' with a text box containing '26'.
- Budget:** 'Cost' with an empty text box.

XP: Cadre méthodologique

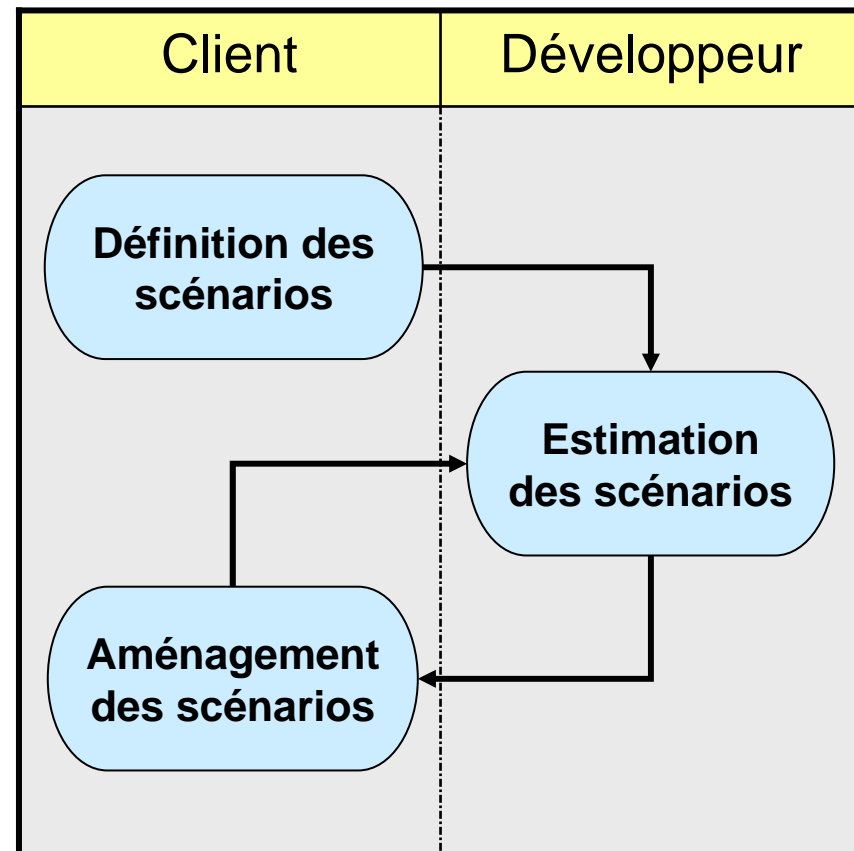
- Exploration
 - Identifier et évaluer le travail
- Engagement
 - Sélectionner et planifier le travail
- Pilotage
 - Contrôler le travail



XP: Phase d'exploration

Au début de chaque cycle

- 1) Description des fonctionnalités sous forme de scénarios (user stories)
- 2) Attribution d'une priorité et d'un risque à chaque scénario
- 3) Estimation du coût (en points) de chaque scénario



XP: « User stories »

- Décrits sur des fiches A5
- Identifiés et recensés par le client en début de projet
- Complétés au début de chaque itération
- Évalués et estimés par les développeurs
- Mis à jour à chaque changement

Priorité fonctionnelle:
1: Indispensable
2: Essentiel
3: Utile

Risque technique:
1: Fort
2: Moyen
3: Faible

Estimation en points

Projet : Éditeur XML Schéma

Titre : Copier ou Couper / Coller

Priorité : 2

Risque : 3

Estimation: 3

Description:

1. L'utilisateur sélectionne un symbole avec la souris
2. Des « poignées » apparaissent sur le symbole
3. L'utilisateur sélectionne l'action « copier » ou « couper » dans le menu « Edition »
4. Si « couper » est sélectionné, le symbole marqué est effacé.
5. L'utilisateur sélectionne l'action « coller » dans le menu « Edition ».
6. Une copie du symbole apparaît à l'écran.

Tâches à réaliser:

- Ajout des options dans le menu « Edition »
- Gestion des clics souris et poignées de sélection
- Voir pour les raccourcis clavier et le drag & drop

XP: Estimation de coût

- Chaque scénario est estimé
 - Par les développeurs aidés du client
 - En intégrant l'effort de test et de recette
 - Suivant une unité « abstraite » de points (ou en « temps idéal »)
 - En découpant les scénarios de granularité trop importante
 - En prototypant si nécessaire pour lever les inconnues techniques
- Vitesse = Nombre de points de scénarios client que l'équipe peut traiter en une itération

XP: Estimation et planification

Pour chaque itération ...

- Le client **écrit** les user stories
- Les développeurs **estime** les histoires
 - + de 3 semaines: scinder la fonctionnalités
 - - d'une semaine: fusionner des fonctionnalités.
- L'équipe **planifie** l'itération
- Pour chaque histoire de l'itération
 - Les développeurs **implémente** les histoires
 - Le client **exécute** les tests de recette

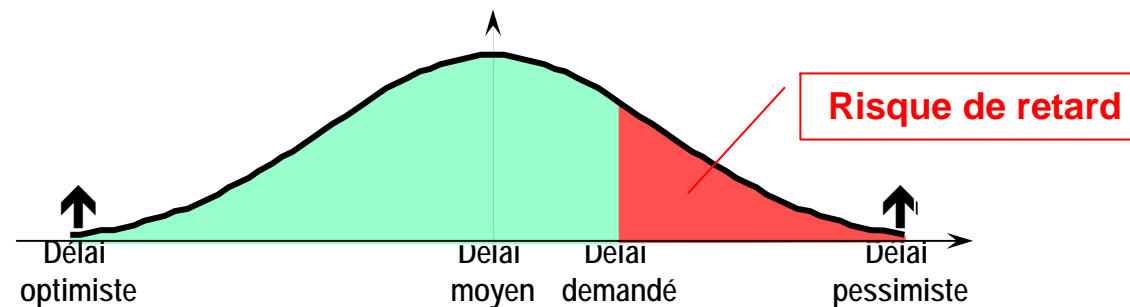


Scrum planning poker

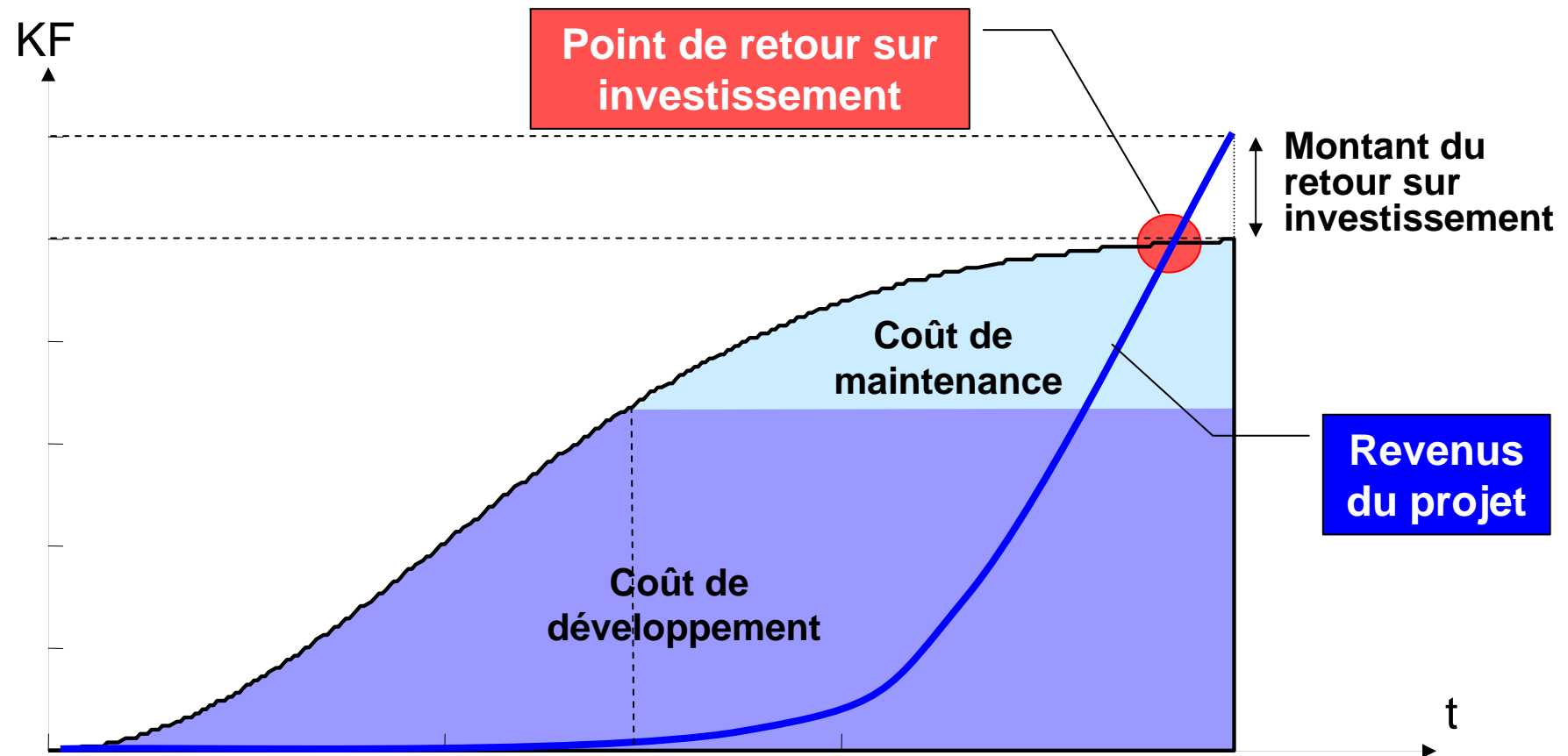


Recommandations

- Ne jamais se limiter a une seule estimation.
- Confronter les résultats obtenus avec différentes méthodes et/ou différents individus
- Faire la moyenne des différentes estimations en pondérant éventuellement
- Construire puis utiliser un référentiel de base pour calibrer le modèle
 - Pour comptabiliser les lignes ou les points
 - Pour déterminer les coefficients relatifs aux facteurs d'ajustement
 - Pour adapter les équations
 - Pour distribuer l'effort correspondant a chaque phase
- La probabilité pour que le projet se termine a une date est régie par la loi de Gauss



Retour sur investissement



Jamais content !!!

