

Le cycle de vie d'un logiciel : Planification

Planification et estimation des coûts

Le cycle de vie d'un logiciel : Planification

- Estimation des coûts et délais par la méthode COCOMO

COCOMO est un acronyme pour **CO**nstructive **CO**st **M**odel. C'est une méthode pour estimer le coût d'un projet logiciel dans le but d'éviter les erreurs de budget et les retards de livraison, qui sont malheureusement habituels dans l'industrie de développement logiciel.

Le cycle de vie d'un logiciel : Planification

- Estimation des coûts et délais par la méthode COCOMO

Le premier modèle **COCOMO** date de **1981**, et a été développé par Dr. Barry Boehm pour estimer le coût , **en nombre de mois-homme**, et le temps de développement d'un produit logiciel.

A l'origine il a été construit sur une étude de 63 projets logiciels de 2000 à 100.000 lignes de code dans l'entreprise TRW Inc., **mais une seule entreprise est-elle assez représentative comme base de développement de COCOMO?** De plus, il reste très lié au nombre de lignes de code, surtout le modèle de base, mais **plus les programmeurs sont experts (et leur salaire élevé), moins ils écrivent de lignes de code pour un même projet!**

Le cycle de vie d'un logiciel : Planification

- Estimation des coûts et délais par la méthode COCOMO

Aujourd'hui, **COCOMO II** est un nouveau produit beaucoup plus adapté à l'aspect réutilisation des composants (modules existants).

La version 1998 a été calibrée sur 161 points de données, en utilisant l'approche statistique 'Bayesian' pour combiner les données empiriques avec les avis experts. De plus elle peut être re-calibrée sur les données de l'entreprise.

Le cycle de vie d'un logiciel : Planification

- Estimation des coûts et délais par la méthode COCOMO

Modèle de Base :

Le modèle de base est assez simpliste. Il estime **l'effort** (le nombre de mois-homme) en fonction du nombre de lignes de code, **la productivité** (le nombre de lignes de code par personne par mois) et un facteur **d'échelle qui dépend du type de projet**.

Les 3 types de projet identifiés sont :

- ☐ **organique** : organisation simple et petites équipes expérimentées. (ex: système de notes dans une école; Produit sans interaction avec matériel,etc.)
- ☐ **semi-detaché** : entre organique et imbriqué. (Produit avec peu d'interaction avec le matériel Exemple : compilateur; Forte interaction avec le matériel)
- ☐ **imbriqué** : techniques innovante, organisation complexe, couplage fort avec beaucoup d'interactions. (ex : système de contrôle aérospatial.)

TABLE 6-3 Distinguishing Features of Software Development Modes

Feature	Mode		
	Organic	Semidetached	Embedded
Organizational understanding of product objectives	Thorough	Considerable	General
Experience in working with related software systems	Extensive	Considerable	Moderate
Need for software conformance with pre-established requirements	Basic	Considerable	Full
Need for software conformance with external interface specifications	Basic	Considerable	Full
Concurrent development of associated new hardware and operational procedures	Some	Moderate	Extensive
Need for innovative data processing architectures, algorithms	Minimal	Some	Considerable
Premium on early completion	Low	Medium	High
Product size range	<50 KDSI	<300 KDSI	All sizes
Examples	Batch data reduction Scientific models Business models Familiar OS, compiler Simple inventory, production control	Most transaction processing systems New OS, DBMS Ambitious inventory, production control Simple command-control	Large, complex transaction processing systems Ambitious, very large OS Avionics Ambitious command-control

**KDSI : thousands of delivered line of code*

Le cycle de vie d'un logiciel : Planification

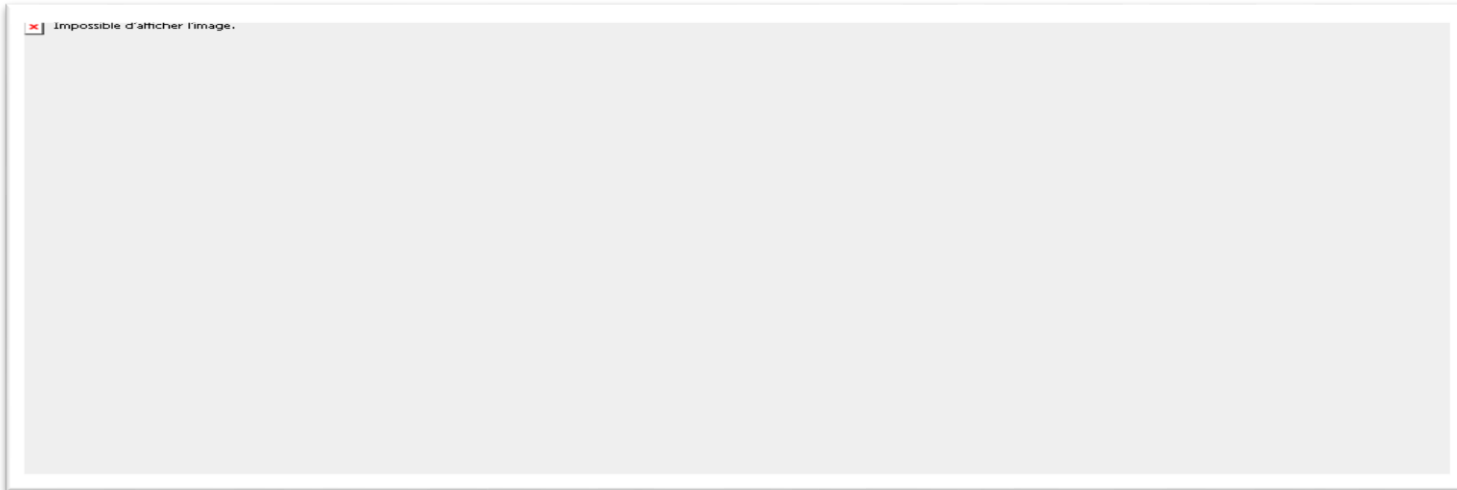
- Estimation des coûts et délais par la méthode COCOMO

Modèle de Base :

Effort : $PM = A \times KLOC^E$ (hommes-mois)

$TDEV = C \times PM^F$ (mois)

La taille de l'équipe = $PM/TDEV$ (hommes)



Le cycle de vie d'un logiciel : Planification

- Estimation des coûts et délais par la méthode COCOMO

Modèle de Base :

Exemple

- Une entreprise souhaite gérer les matières premières qu'elle utilise. Elle fait appel à ses informaticiens en interne, qui ont l'habitude de ce genre de projets.
- Étude initiale : environ 32000 instructions.

Le cycle de vie d'un logiciel : Planification

- Estimation des coûts et délais par la méthode COCOMO

Modèle de Base :

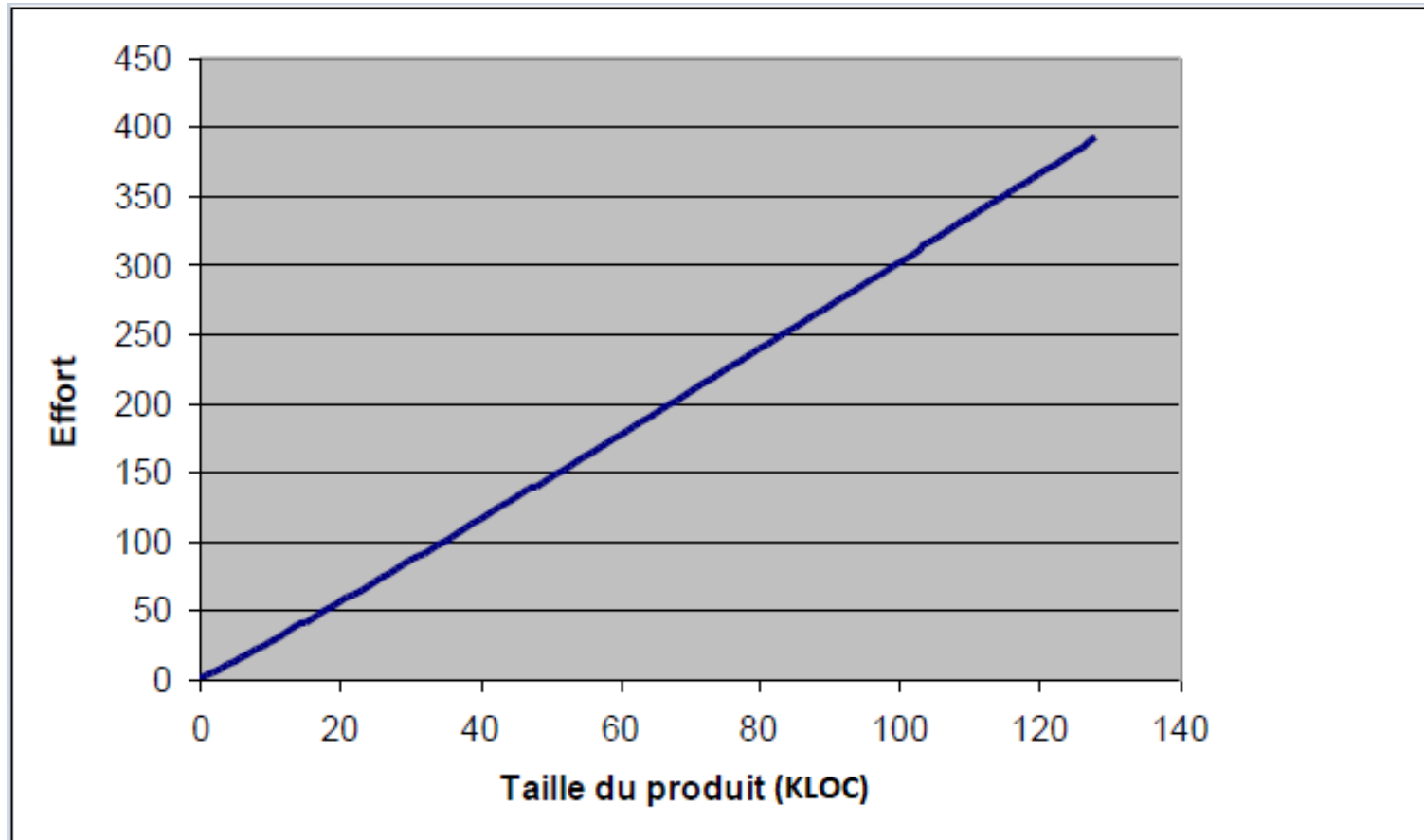
Exemple

- $PM = 2.4 \times 32^{1.05} = 91$ homme-mois
- $TDEV = 2.5 \times 91^{0.38} = 14$ mois
- Taille équipe = $91\text{h-m} / 14 \text{ mois} = 6.5$ personnes à temps plein
- Productivité = Taille du projet / PM = 352 lignes/homme-mois

Le cycle de vie d'un logiciel : Planification

- Estimation des coûts et délais par la méthode COCOMO

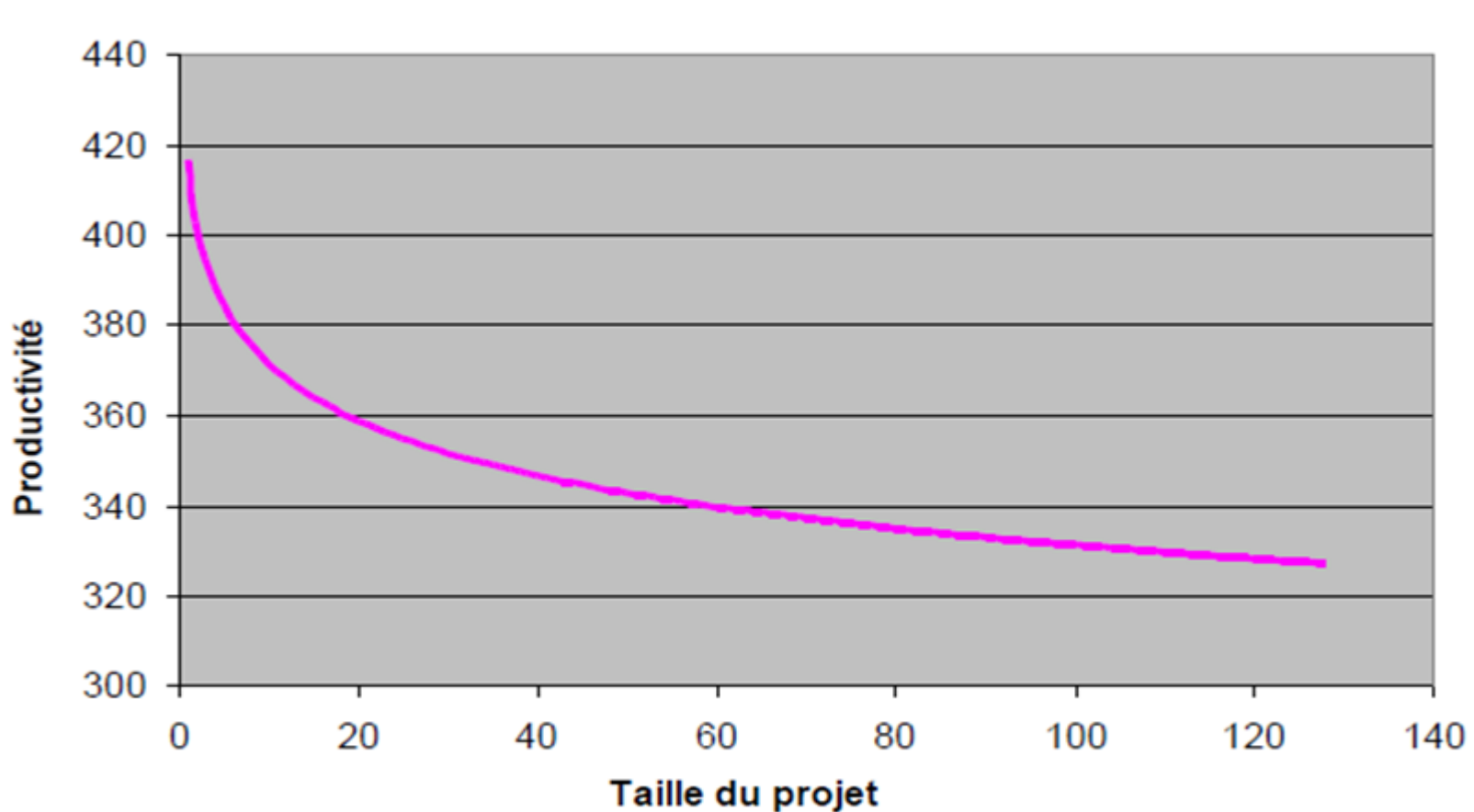
La relation entre effort (PM) et Taille du produit en KLOC



Le cycle de vie d'un logiciel : Planification

- Estimation des coûts et délais par la méthode COCOMO

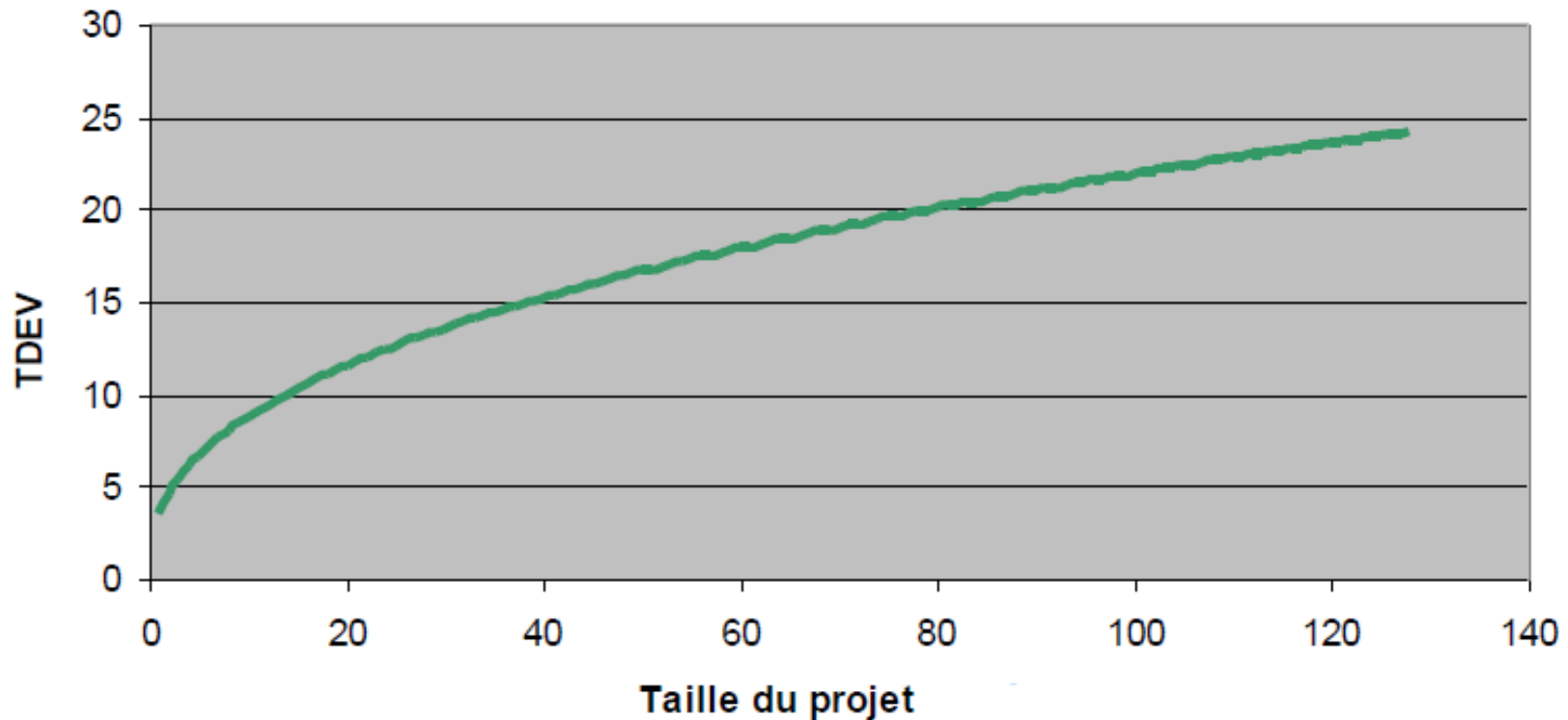
La relation entre productivité (lignes/h-mois)) et Taille du produit en KLOC



Le cycle de vie d'un logiciel : Planification

- Estimation des coûts et délais par la méthode COCOMO

La relation entre TDEV (mois) et Taille du produit en KLOC



Le cycle de vie d'un logiciel : Planification

Distribution par phase du temps de développement *En Pourcentage*

Complexité	Phase	Taille de 2 KLS	Taille de 8 KLS	Taille de 32 KLS	Taille de 128 KLS	Taille de 512 KLS
S	<i>Expression des besoins et planification</i>	10	11	12	13	
	<i>Conception générale</i>	19	19	19	19	
	<i>Programmation</i>	63	59	55	51	
	<i>Tests et intégration</i>	18	22	26	30	
P	<i>Expression des besoins et planification</i>	16	18	20	22	24
	<i>Conception générale</i>	24	25	26	27	28
	<i>Programmation</i>	56	52	48	44	40
	<i>Tests et intégration</i>	20	23	26	29	32
E	<i>Expression des besoins et planification</i>	24	28	32	36	40
	<i>Conception générale</i>	30	32	34	36	38
	<i>Programmation</i>	48	44	40	36	32
	<i>Tests et intégration</i>	22	24	26	28	30

Le cycle de vie d'un logiciel : Planification

Le modèle COCOMO intermédiaire

Modèle de base + attributs

Attributs du produit

- RELY : sureté du produit
- DATA : taille de la base de données
- CPLX : complexité du produit

Attributs de la machine

TIME : contraintes de temps d'exécution

STOR : contraintes de mémoire principale

VIRT : changement dans l'ensemble soft+hard

TURN : temps de retour d'une tâche soumise à l'ordinateur.

Le cycle de vie d'un logiciel : Planification

Le modèle COCOMO intermédiaire

Attributs du personnel

- ACAP : aptitude de l'analyste
- AEXP : expérience de ces applications
- PCAP : aptitude du programmeur
- VEXP : expérience de l'ensemble soft+hard
- LEXP : expérience du langage de programmation

Attributs du projet

- MODP : pratique des techniques modernes de programmation
- TOOL : utilisation d'outils
- SCED : agenda imposé

TABLE 8-3 Software Cost Driver Ratings

Cost Driver	Ratings					
	Very Low	Low	Nominal	High	Very High	Extra High
Product attributes						
RELY	Effect: slight inconvenience	Low, easily recoverable losses	Moderate, recoverable losses	High financial loss	Risk to human life	
DATA		$\frac{\text{DB bytes}}{\text{Prog. DSI}} < 10$	$10 \leq \frac{D}{P} < 100$	$100 \leq \frac{D}{P} < 1000$	$\frac{D}{P} \geq 1000$	
CPLX	See Table 8-4					
Computer attributes						
TIME			$\leq 50\%$ use of available execution time	70%	85%	95%
STOR			$\leq 50\%$ use of available storage	70%	85%	95%
VIRT		Major change every 12 months Minor: 1 month	Major: 6 months Minor: 2 weeks	Major: 2 months Minor: 1 week	Major: 2 weeks Minor: 2 days	
TURN		Interactive	Average turnaround <4 hours	4-12 hours	>12 hours	
Personnel attributes						
ACAP	15th percentile ^a	35th percentile	55th percentile	75th percentile	90th percentile	
AEXP	≤ 4 months experience	1 year	3 years	6 years	12 years	
PCAP	15th percentile ^a	35th percentile	55th percentile	75th percentile	90th percentile	
VEXP	≤ 1 month experience	4 months	1 year	3 years		
LEXP	≤ 1 month experience	4 months	1 year	3 years		
Project attributes						
MODP	No use	Beginning use	Some use	General use	Routine use	
TOOL	Basic microprocessor tools	Basic mini tools	Basic midi/maxi tools	Strong maxi programming, test tools	Add requirements, design, management, documentation tools	
SCED	75% of nominal	85%	100%	130%	160%	

^a Team rating criteria: analysis (programming) ability, efficiency, ability to communicate and cooperate

Le cycle de vie d'un logiciel : Planification

Le modèle COCOMO intermédiaire

- Pour chaque attribut
 - ▣ On associe un facteur multiplicatif
 - ▣ On multiplie ces 15 facteurs
 - ▣ Puis équations :
 - Organique : $MM_{\text{nominal}} = 3.2 \times KDSI^{1.05}$
 - Semi détaché : $MM_{\text{nominal}} = 3.0 \times KDSI^{1.12}$
 - Embarqué : $MM_{\text{nominal}} = 2.8 \times KDSI^{1.20}$
 - ▣ Finalement :
 - $MM = \text{ProduitFacteurs} \times MM_{\text{nominal}}$

*MM == PM

*KDSI == KLOC

Le cycle de vie d'un logiciel : Planification

Le modèle COCOMO intermédiaire : Facteurs multiplicatifs

TABLE 8-2 Software Development Effort Multipliers

Cost Drivers	Ratings					
	Very Low	Low	Nominal	High	Very High	Extra High
Product Attributes						
RELY Required software reliability	.75	.88	1.00	1.15	1.40	
DATA Data base size		.94	1.00	1.08	1.16	
CPLX Product complexity	.70	.85	1.00	1.15	1.30	1.65
Computer Attributes						
TIME Execution time constraint			1.00	1.11	1.30	1.66
STOR Main storage constraint			1.00	1.06	1.21	1.56
VIRT Virtual machine volatility ^a		.87	1.00	1.15	1.30	
TURN Computer turnaround time		.87	1.00	1.07	1.15	
Personnel Attributes						
ACAP Analyst capability	1.46	1.19	1.00	.86	.71	
AEXP Applications experience	1.29	1.13	1.00	.91	.82	
PCAP Programmer capability	1.42	1.17	1.00	.86	.70	
VEXP Virtual machine experience ^a	1.21	1.10	1.00	.90		
LEXP Programming language experience	1.14	1.07	1.00	.95		
Project Attributes						
MODP Use of modern programming practices	1.24	1.10	1.00	.91	.82	
TOOL Use of software tools	1.24	1.10	1.00	.91	.83	
SCED Required development schedule	1.23	1.08	1.00	1.04	1.10	

Le cycle de vie d'un logiciel : Planification

Le modèle COCOMO intermédiaire : Facteurs multiplicatifs

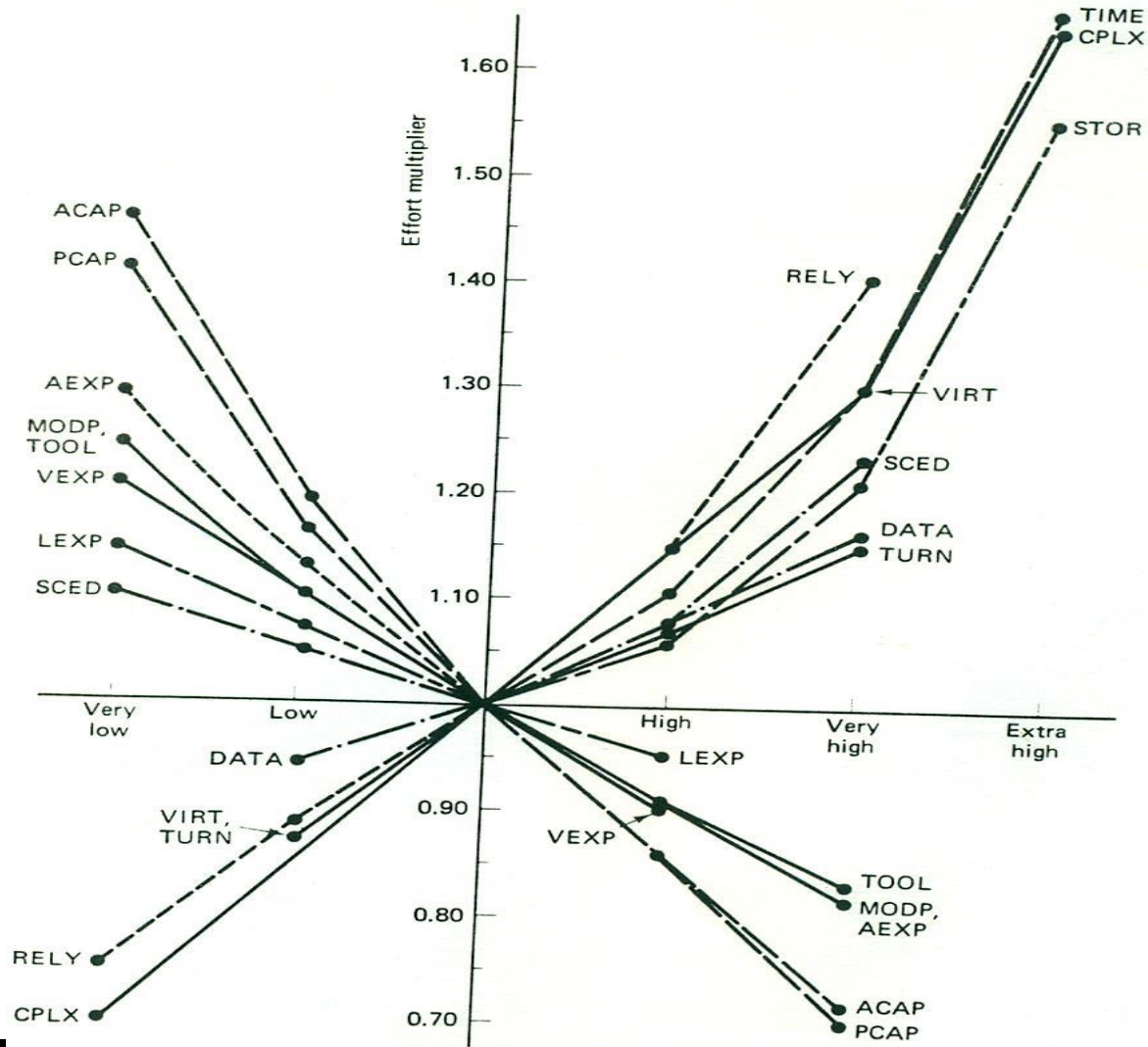


FIGURE 8-2 Intermediate COCOMO effort multipliers

Le cycle de vie d'un logiciel : Planification

Méthode d'estimation à trois points

La méthode d'estimation à 3 points, se base sur des dates estimées (temps moyen/temps optimiste/temps pessimiste) et sur l'écart type.

TM: temps moyen estimé (travail dans des conditions normales).

TO: temps optimiste (conditions idéales, pas d'obstacles, temps minimum pour accomplir la tâche).

TP: temps pessimiste (temps maximum pour accomplir la tâche dans les pires conditions).

- Pour une probabilité avec une distribution normale, la formule est :

$$P = TO + (TP - TO) / 2$$

- Pour une probabilité avec une distribution Beta (à double triangulaire) formule de répartition sera la suivante : $P = (TP + 4 TM + TO) / 6$

Le cycle de vie d'un logiciel : Planification

Méthode des Potentiels et antécédents Métra « M.P.M »

- La Méthode des Potentiels et antécédents Métra (MPM) est, comme le PERT, une technique d'ordonnancement basée sur la théorie des graphes, visant à optimiser la planification des tâches d'un projet.
- mise au point en 1958 par un chercheur français, Bernard Roy, au sein de la société de conseil Métra.

Le cycle de vie d'un logiciel : Planification

MÉTHODOLOGIE DE CONSTRUCTION D'UN RÉSEAU MPM

Étapes clés de la méthode M.P.M:

- Étape 1 : Énumérer les activités

Nom du projet	Tâche 1	Sous-tâche 1.1	Élément de travail 1.1.1
			Élément de travail 1.1.2
		Sous-tâche 1.2	Élément de travail 1.2.1
			Élément de travail 1.1.2
	Tâche 2		
		Sous-tâche 2.1	
			Élément de travail 2.1.1
			Élément de travail 1.1.1

Le cycle de vie d'un logiciel : Planification

MÉTHODOLOGIE DE CONSTRUCTION D'UN RÉSEAU MPM

Étapes clés de la méthode du MPM:

Étape 2 : Établir les dépendances (séquence d'activités)

- Quelle tâche doit être réalisée avant que cette tâche ne se réalise ?
- Quelles tâches doivent être achevées en même temps que celle-ci ?
- Quelles tâches doivent être réalisées immédiatement après celle-ci ?

Tâches	Durée	Antériorité
A	2	-
B	4	-
C	4	A
D	5	A,B
E	6	C,D

TABLEAU D'ANTÉRIORITÉ DU PROJET

Le cycle de vie d'un logiciel : Planification

MÉTHODOLOGIE DE CONSTRUCTION D'UN RÉSEAU MPM

Étapes clés de la méthode du MPM

Étape 3 : Dessiner le graphe MPM

La méthode des potentiels Métra permet de représenter l'ensemble de ces tâches sur un graphe orienté, à partir duquel il sera possible d'identifier leurs dates au plus et au plus tard et de calculer leurs marges.

Un graphe orienté est un réseau composé d'une entrée et d'une sortie, ainsi que de points (appelés "sommets") reliés entre eux par des flèches (appelées "arcs").

Le cycle de vie d'un logiciel : Planification

MÉTHODOLOGIE DE CONSTRUCTION D'UN RÉSEAU MPM

Étapes clés de la méthode du MPM

Étape 3 : Dessiner le graphe MPM

Les principales conventions d'un réseau MPM sont les suivantes :

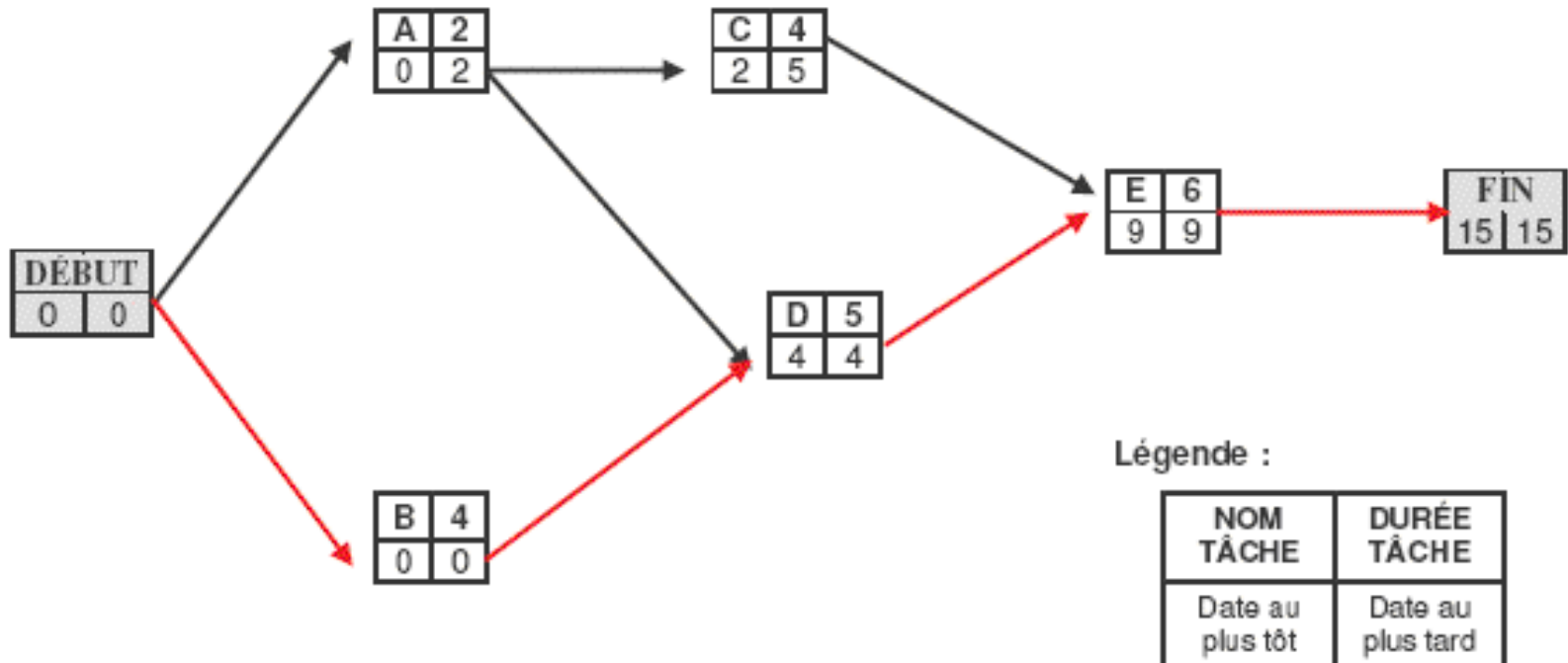
- chaque tâche est représentée par un sommet
- les contraintes de succession sont symbolisées par les arcs
- chaque tâche est renseignée sur sa durée ainsi que sur la date à laquelle elle peut commencer au plus tôt ("date au plus tôt") et au plus tard ("date au plus tard") pour respecter le délai optimal de réalisation du projet.
- le graphe commence et termine sur 2 sommets, respectivement appelés "Début" et "Fin" symbolisant les début et fin des opérations (mais ne correspondant pas une tâche).

Le cycle de vie d'un logiciel : Planification

MÉTHODOLOGIE DE CONSTRUCTION D'UN RÉSEAU MPM

Étapes clés de la méthode du MPM

Étape 3 : Dessiner le graphe MPM



Le cycle de vie d'un logiciel : Planification

MÉTHODOLOGIE DE CONSTRUCTION D'UN RÉSEAU MPM

Étapes clés de la méthode du MPM

Étape 4 :détermination des dates "au plus tôt" et "au plus tard" dans un réseau MPM

La date au plus tôt d'un réseau MPM correspond à la date à laquelle une tâche peut commencer au plus tôt.

- Elle s'obtient très simplement en ajoutant à la date au plus tôt de la tâche précédente la durée de la tâche en question :

$$\text{Date au plus tôt tâche T} = \text{Date au plus tôt tâche S} + \text{Durée tâche S}$$

Le cycle de vie d'un logiciel : Planification

MÉTHODOLOGIE DE CONSTRUCTION D'UN RÉSEAU MPM

Étapes clés de la méthode du MPM

Étape 4 :détermination des dates "au plus tôt" et "au plus tard" dans un réseau MPM

Lorsque plusieurs arcs arrivent à un même sommet (c'est-à-dire que plusieurs tâches sont immédiatement antérieures à la tâche considérée), il convient, d'effectuer ce calcul pour toutes les tâches précédant la tâche en question et de retenir comme "date au plus tôt" de cette dernière le maximum des valeurs ainsi trouvée (en effet, cette tâche ne pourra vraiment débuter que lorsque toutes les tâches qui lui sont immédiatement antérieures auront été terminées).

La formule précédente devient donc :

$$\text{Date au plus tôt tâche T} = \text{Max. (Date plus tôt tâches S + Durée tâches S)}$$

*Dans cette formule, "S" représente l'ensemble des tâches immédiatement antérieures à "T"

Le cycle de vie d'un logiciel : Planification

MÉTHODOLOGIE DE CONSTRUCTION D'UN RÉSEAU MPM

Étapes clés de la méthode du MPM

Étape 4 :détermination des dates "au plus tôt" et "au plus tard" dans un réseau MPM

La détermination des dates au plus tard des différentes tâches se fait à rebours du graphe, par calculs successifs, en partant du sommet "Fin" (pour lequel, par convention, on considère que la date au plus tard est égale à sa date au plus tôt).

Date au plus tard tâche S = Min. (date au plus tard tâches T - durée tâche S)

*Dans cette formule, "T" représente l'ensemble des tâches immédiatement postérieures à "S"

Le cycle de vie d'un logiciel : Planification

MÉTHODOLOGIE DE CONSTRUCTION D'UN RÉSEAU MPM

Étapes clés de la méthode du MPM

Étape 4 :calcul des différentes marges d'une tâche dans un réseau MPM

On appelle "**marge**" d'une tâche le retard qu'il est possible de tolérer dans la réalisation de celle-ci, sans que la durée optimale prévue du projet global en soit affectée.

Il est possible de calculer trois types de marges : **la marge totale, la marge certaine et la marge libre.**

Le cycle de vie d'un logiciel : Planification

MÉTHODOLOGIE DE CONSTRUCTION D'UN RÉSEAU MPM

Étapes clés de la méthode du MPM

Étape 4 : calcul des différentes marges d'une tâche dans un réseau MPM

La marge totale d'une tâche indique **le retard maximal** que l'on peut admettre dans sa réalisation (sous réserve qu'elle ait commencé à sa date au plus tôt) **sans allonger la durée optimale du projet.**

$$\text{Marge totale tâche S} = \text{Date plus tard tâche S} - \text{Date plus tôt tâche S}$$

***cas particulier**, un retard correspondant à la marge totale d'une tâche se traduit par une modification des dates au plus tôt des tâches qui lui succèdent et entraîne, généralement, l'apparition d'un 2° chemin critique. Il n'est donc pas possible de cumuler des retards correspondant à leur marge totale sur plusieurs tâches successives, sans remettre en cause la durée optimale prévue pour le projet.

Le cycle de vie d'un logiciel : Planification

MÉTHODOLOGIE DE CONSTRUCTION D'UN RÉSEAU MPM

Étapes clés de la méthode du MPM

Étape 4 : calcul des différentes marges d'une tâche dans un réseau MPM

La marge libre d'une tâche indique **le retard que l'on peut admettre dans sa réalisation** (sous réserve qu'elle ait commencé à sa date au plus tôt) **sans modifier les date au plus tôt des tâches suivantes et sans allonger la durée optimale du projet.**

Elle se calcule en retirant la durée de la tâche en question à l'écart existant entre sa date au plus tôt de la date au plus tôt de la tâche suivante :

$$\text{Marge libre tâche S} = \text{Date plus tôt tâche T} - \text{Date plus tôt tâche S} - \text{Durée tâche S}$$

*T : Tâche suivante à S.

Le cycle de vie d'un logiciel : Planification

MÉTHODOLOGIE DE CONSTRUCTION D'UN RÉSEAU MPM

Étapes clés de la méthode du MPM

Étape 4 : calcul des différentes marges d'une tâche dans un réseau MPM

Lorsque plusieurs arcs partent d'un même sommet, il convient de faire ce calcul pour toutes les tâches succédant à la tâche en question et de retenir comme "marge libre" de la tâche en question la valeur minimale des marges ainsi déterminées :

Marge libre tâche S = Min (Date plus tôt tâches T - Date plus tôt tâche S - Durée tâche S)

- Dans cette formule T représente l'ensemble des tâches succédant immédiatement à S
- Un retard correspondant à la marge libre d'une tâche reste sans conséquence sur les marges des tâches qui lui succèdent. Il est donc **possible de cumuler des retards**, s'inscrivant dans leur marge libre, **pour plusieurs tâches successives**, sans remettre en cause la durée optimale prévue pour le projet.

Le cycle de vie d'un logiciel : Planification

MÉTHODOLOGIE DE CONSTRUCTION D'UN RÉSEAU MPM

Étapes clés de la méthode du MPM

Étape 4 : calcul des différentes marges d'une tâche dans un réseau MPM

La marge certaine d'une tâche indique le retard que l'on peut admettre dans sa réalisation (quelle que soit sa date de début) sans allonger la durée optimale du projet.

Elle se calcule en retirant la durée de la tâche en question à l'écart qu'il peut y avoir entre sa date au plus tard de début et sa date au plus tôt de fin :

$$\text{Marge certaine tâche S} = \text{Max} [0 , \text{Min} (\text{Date au plus tôt tâche T} - \text{Date au plus tard tâche S} - \text{Durée tâche S})]$$

* D'après cette formule, la marge certaine est considérée comme nulle lorsque son calcul donne un nombre négatif

Le cycle de vie d'un logiciel : Planification

MÉTHODOLOGIE DE CONSTRUCTION D'UN RÉSEAU MPM

Étapes clés de la méthode du MPM

Étape 4 : calcul des différentes marges d'une tâche dans un réseau MPM

Lorsque plusieurs arcs partent d'un même sommet , il convient de faire ce calcul pour toutes les tâches succédant à la tâche en question et de retenir comme "marge certaine" de cette dernière la valeur minimale des marges ainsi déterminées :

Marge certaine tâche S = Max [0, Min (Date au plus tôt tâches T - Date au plus tard tâche S - Durée tâche S)

*Dans cette formule T représente l'ensemble des tâches succédant immédiatement à S

- Un retard correspondant à **la marge certaine d'une tâche** reste sans conséquence sur les marges des tâches qui lui succèdent, même si elle a commencé à sa date au plus tard.
- Il est donc **possible de cumuler des retards**, s'inscrivant dans leur marge certaine, pour plusieurs tâches successives, même si elles commencent à leur date au plus tard, sans remettre en cause la durée optimale prévue pour le projet.

Le cycle de vie d'un logiciel : Planification

MÉTHODOLOGIE DE CONSTRUCTION D'UN RÉSEAU MPM

Étapes clés de la méthode du MPM

Étape 4 : calcul des différentes marges d'une tâche dans un réseau MPM

Les marges des tâches composant le **chemin critique** sont nécessairement **nulles**, puisqu'il s'agit de tâches pour lesquels, par définition, **aucun retard n'est possible** sans remettre en cause la durée optimale prévue pour le projet.

Le cycle de vie d'un logiciel : Planification

MÉTHODOLOGIE DE CONSTRUCTION D'UN RÉSEAU MPM

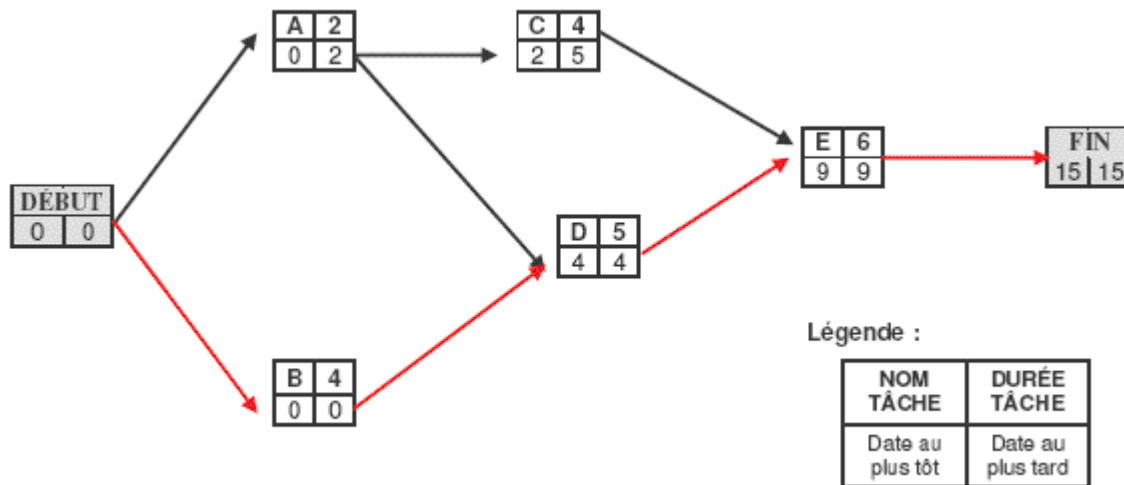
- Exemple : calculer les dates débuts au plus tôt et au plus tard, les marges et déterminer le chemin critique du projet suivant:

Tâches	Durée	Antériorité
A	2	-
B	4	-
C	4	A
D	5	A,B
E	6	C,D

Le cycle de vie d'un logiciel : Planification

MÉTHODOLOGIE DE CONSTRUCTION D'UN RÉSEAU MPM

- Graphe MPM



Le cycle de vie d'un logiciel : Planification

MÉTHODOLOGIE DE CONSTRUCTION D'UN RÉSEAU MPM

- Les marges

Tâches	Marge Totale	Marge libre	Marge certaine
A	2	0	0
B	0	0	0
C	3	3	0
D	0	0	0
E	0	0	0

Le cycle de vie d'un logiciel : Planification

MÉTHODOLOGIE DE CONSTRUCTION D'UN RÉSEAU MPM

- Exercice :

Un projet informatique « P » consiste à développer un site web:

Tâche	Durée (jours)	Antécédent
a. Planification	2	--
b. Conception	5	a
c. Développement de la base de données	7	b
d. Développement du front-end	14	a
e. Développement du back-end	14	c
f. Intégration du front-end et du back-end	5	d,e
g. Tests	4	d,e
h. mise en ligne et déploiement	2	f,g

Le cycle de vie d'un logiciel : Planification

Le diagramme de PERT « Programm of Evaluation and Review Technic »

- Le **PERT** (Programm of Evaluation and Review Technic) est, comme la MPM, une technique d'ordonnancement basée sur la théorie des graphes, visant à optimiser la planification des tâches d'un projet.
- Cette technique aurait été conçue sur la base de la méthode **CPM (Critical Method Path)** par la marine américaine, en 1958, pour coordonner les tâches des milliers d'entreprises impliquées dans son projet "Polaris" (programme de développement de missiles à ogive nucléaire).
- Elle aurait permis de réduire de 14 à 7 ans la durée globale de réalisation du projet Polaris) elle s'est rapidement imposée dans les organisations, gouvernementales ou non, ayant à gérer des projets importants (programme Apollo de la NASA, construction d'autoroute, etc.) au détriment du diagramme de Gantt.

Le cycle de vie d'un logiciel : Planification

Le diagramme de PERT « Program of Evaluation and Review Technic »

- **L'utilisation du PERT permet**, de déterminer la durée minimum nécessaire pour mener à bien un projet et les dates auxquelles peuvent ou doivent débiter les différentes tâches nécessaires à sa réalisation pour que cette durée minimum soit respectée.

Le cycle de vie d'un logiciel : Planification

Le diagramme de PERT « Program of Evaluation and Review Technic »

- Méthodologie de construction d'un réseau PERT

1- identifiées les différentes tâches nécessaires à la réalisation d'un projet, leur durée et leurs relations d'antériorité .

Tâches	Durée	Antériorité(s)
A	2	-
B	4	-
C	4	A
D	5	A,B
E	6	C,D

Le cycle de vie d'un logiciel : Planification

Le diagramme de PERT « Programm of Evaluation and Review Technic »

- **Méthodologie de construction d'un réseau PERT**

Les principales conventions d'un réseau PERT sont les suivantes :

- chaque **tâche est symbolisée par un arc**, auquel est associé une valeur numérique correspondant à sa durée.
- Les **sommets** auxquels aboutissent les arcs correspondent donc à **des étapes**, qui marquent l'aboutissement d'une ou plusieurs tâches.
- **Chaque étape** est identifiée par un numéro d'ordre et renseignée sur la date à laquelle elle peut être atteinte au plus tôt ("**date au plus tôt**") et au plus tard ("**date au plus tard**") pour respecter le délai optimal de réalisation du projet.
- Le graphe possède une entrée (sommet sans antécédent) et une sortie (sommet sans descendant) qui correspondent respectivement aux étapes "Début des opérations" et "Fin des opérations".
- il est parfois nécessaire d'introduire des "**tâches fictives**" pour **traduire correctement sur un graphe les relations d'antériorité** de certaines tâches, notamment lorsque celles-ci partagent avec d'autres une partie de leurs antécédents

Le cycle de vie d'un logiciel : Planification

Le diagramme de PERT « Programm of Evaluation and Review Technic »

- Méthodologie de construction d'un réseau PERT

2-Construction d'un graphe PERT

La démarche la plus appropriée consiste à procéder par "niveau" :

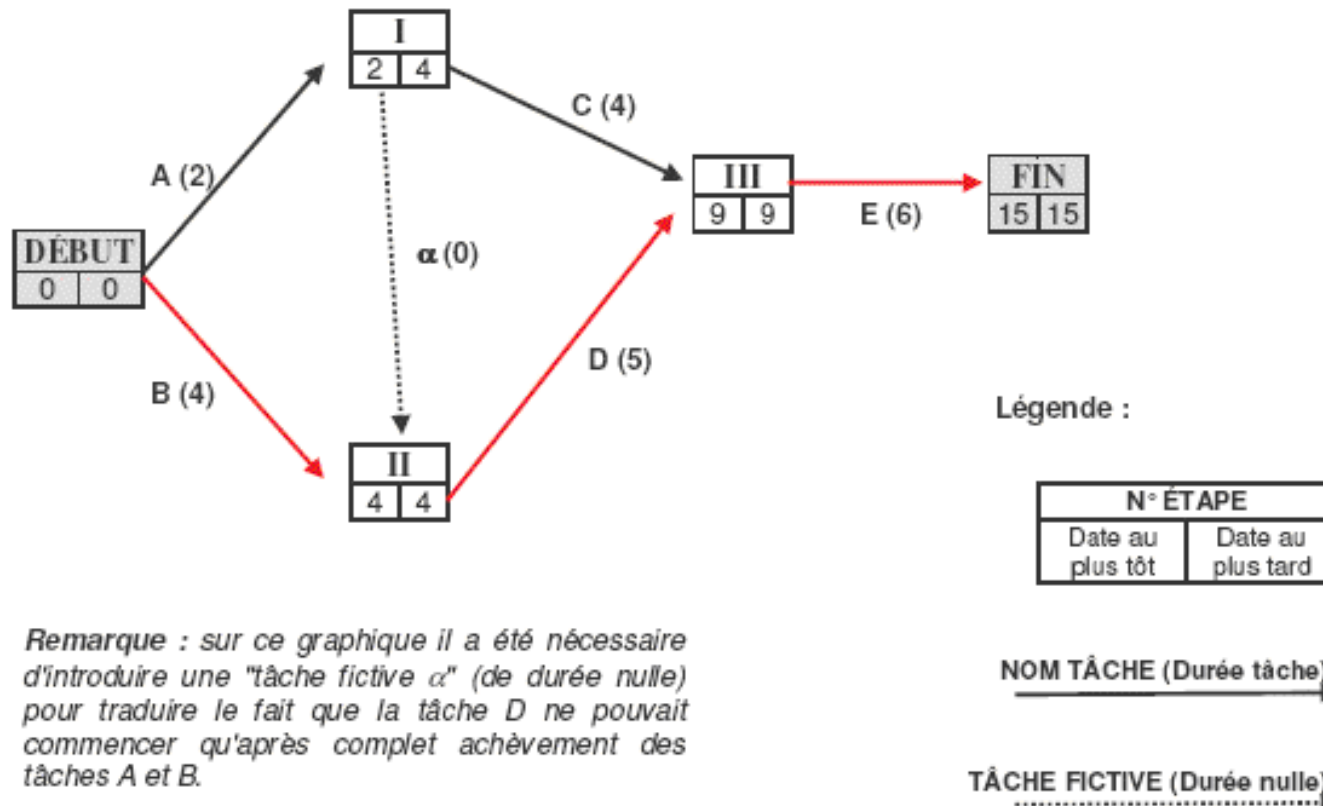
- Déterminer les tâches sans antécédent (tâches de niveau 1) et les relier à l'étape de "Début" .
- Identifier ensuite les tâches de niveau 2, c'est-à-dire celles dont les antécédents sont exclusivement du niveau 1 et les positionner sur le graphique en fonction de des derniers,
- continuer ainsi, jusqu'à ce que toutes les tâches aient pu être positionnées entre elles et relier celles n'ayant pas de descendant à l'étape de "Fin".

Le cycle de vie d'un logiciel : Planification

Le diagramme de PERT « Programm of Evaluation and Review Technic »

- Méthodologie de construction d'un réseau PERT

2-Construction d'un graphe PERT



Le cycle de vie d'un logiciel : Planification

Le diagramme de PERT « Programm of Evaluation and Review Technic »

- Méthodologie de construction d'un réseau PERT

2-Détermination des dates "au plus tôt" et "au plus tard" dans un réseau PERT

Date au plus tôt "étape j" = Date au plus tôt "étape i" + Durée tâche "ij"

Date au plus tôt "étape j" = Max. (Date plus tôt "étapes i" + Durée tâches "ij")

Le cycle de vie d'un logiciel : Planification

Le diagramme de PERT « Programm of Evaluation and Review Technic »

- Méthodologie de construction d'un réseau PERT

2-Détermination des dates "au plus tôt" et "au plus tard" dans un réseau PERT

Date au plus tard "étape i" = Date plus tard "étape j" - Durée tâche "ij"

Date au plus tard "étape i" = Min. (date au plus tard "étapes j" - Durée tâches "ij")

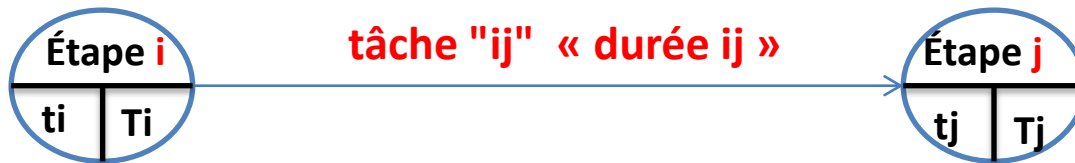
Le cycle de vie d'un logiciel : Planification

Le diagramme de PERT « Programm of Evaluation and Review Technic »

- Méthodologie de construction d'un réseau PERT

3-Détermination des Marges:

Marge totale tâche "ij" = Date au plus tard "étape j" - Date au plus tôt "étape i" - Durée tâche "ij"



Marge totale tâche "ij" = $T_j - t_i$ - Durée tâche "ij"

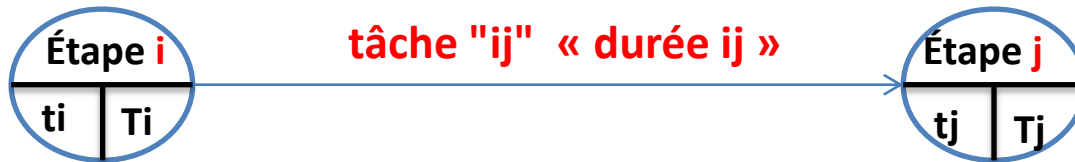
Le cycle de vie d'un logiciel : Planification

Le diagramme de PERT « Programm of Evaluation and Review Technic »

- Méthodologie de construction d'un réseau PERT

3-Détermination des Marges:

Marge libre tâche "ij" = Date au plus tôt "étape j" - Date au plus tôt "étape i" - Durée tâche "ij"



Marge libre tâche "ij" = $t_j - t_i - \text{durée « ij »}$

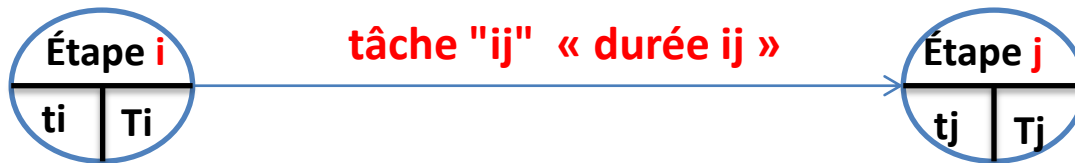
Le cycle de vie d'un logiciel : Planification

Le diagramme de PERT « Programm of Evaluation and Review Technic »

- Méthodologie de construction d'un réseau PERT

3-Détermination des Marges:

Marge certaine tâche "ij" = $\text{Max} [0 , (\text{Date au plus tôt "étape j" - Date au plus tard "étape i" - Durée tâche "ij")]$



Marge certaine tâche "ij" = $\text{Max} [0 , (t_j - T_i - \text{Durée "ij"})]$

Le cycle de vie d'un logiciel : Planification

Le diagramme de PERT « Programm of Evaluation and Review Technic »

- Méthodologie de construction d'un réseau PERT

3-Détermination le chemin critique :

On remarque que l'ensemble des marges des tâches composant le **chemin critique** sont nécessairement nulles, puisqu'il s'agit de tâches pour lesquels, par définition, aucun retard n'est possible sans remettre en cause la durée optimale prévue pour le projet.

Le cycle de vie d'un logiciel : Planification

Diagramme de PERT

- Exercice :

Un projet informatique « P » consiste à développer un site web:

Tâche	Durée (jours)	Antécédent
a. Planification	2	--
b. Conception	5	a
c. Développement de la base de données	7	b
d. Développement du front-end	14	a
e. Développement du back-end	14	c
f. Intégration du front-end et du back-end	5	d,e
g. Tests	4	d,e
h. mise en ligne et déploiement	2	f,g

Le cycle de vie d'un logiciel : Planification

Diagramme de Gantt

- Le **Diagramme de Gantt** a été conçu en 1917, par Henry L. Gantt.
- Le Diagramme de Gantt se présente **sous la forme d'un planning** présentant en ligne **les tâches élémentaires** d'un projet et en colonne **l'échelle de temps** retenue (jours, semaine, etc.).

Le cycle de vie d'un logiciel : Planification

MÉTHODOLOGIE DE CONSTRUCTION D'UN DIAGRAMME DE GANTT

1. Déterminer et structurer la liste des tâches.
2. Estimer les durées et les ressources des tâches identifiées.
3. Réaliser le "réseau logique" : les relations d'antériorité des tâches définies.

Le cycle de vie d'un logiciel : Planification

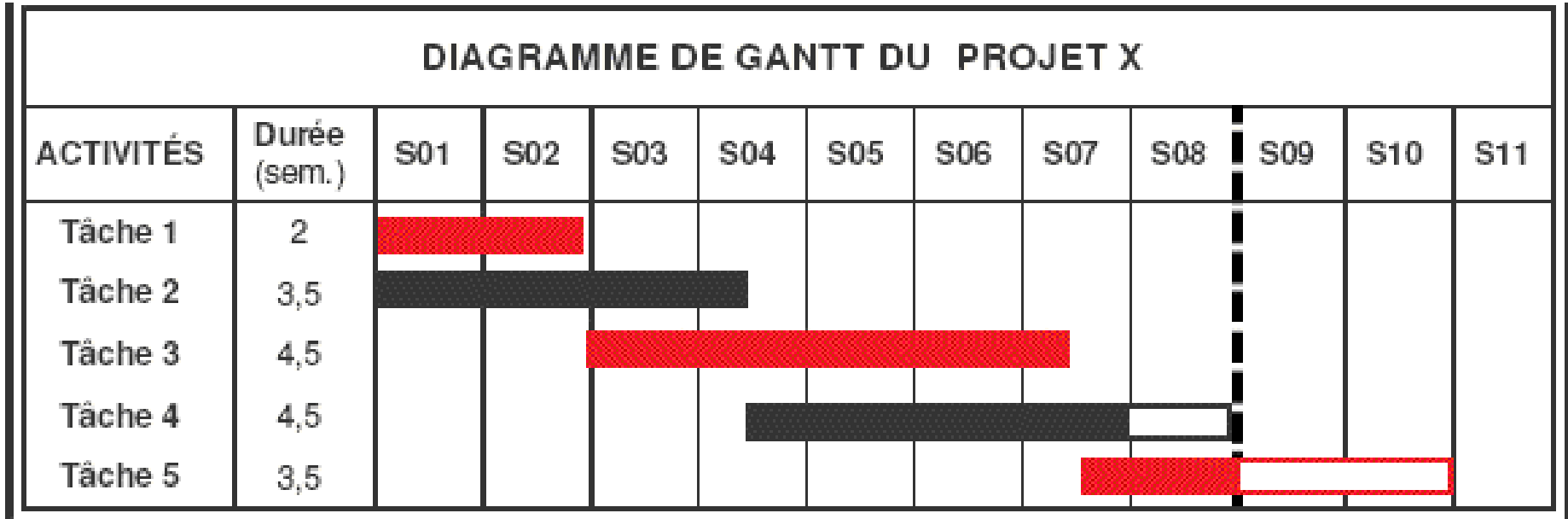
MÉTHODOLOGIE DE CONSTRUCTION D'UN DIAGRAMME DE GANTT

4) Tracer le diagramme de GANTT :

- Apparaître en ordonnée la liste des "lots de tâches" (*workpackages*) et en abscisse l'échelle de temps adopté (jours, semaines ou mois).
- Les tâches sont figurées sous la forme de traits ou de rectangles d'une longueur proportionnelle à leur durée, leur position horizontale reflétant leur ordre logique d'exécution.
- La méthode la plus simple consiste à commencer par représenter les tâches n'ayant aucune antériorité, puis celles qui peuvent immédiatement leur succéder et ainsi de suite jusqu'à ce que toutes les tâches aient été positionnées.
- Éventuellement plusieurs tâches peuvent être réalisées simultanément (sous réserve d'une disponibilité des ressources nécessaires) ce qui permet de diminuer la durée totale d'exécution du projet et, donc, son coût.

Le cycle de vie d'un logiciel : Planification

MÉTHODOLOGIE DE CONSTRUCTION D'UN DIAGRAMME DE GANTT



Le cycle de vie d'un logiciel : Planification

DIAGRAMME DE GANTT : EXERCICES

Tracer le diagramme de gantt du projet suivant :

<i>Code de la tâche</i>	<i>Désignation de la tâche</i>	<i>Durée en jours</i>	<i>Tâches antérieures</i>
A	Définition des contraintes	2	B E
B	Mise en place du projet	6	-
C	Mise à jour des droits d'accès	2	F
D	Achat des composants matériels	8	J
E	Définition du budget	3	-
F	Mise à jour des gr. utilisateurs	2	K
G	Formation de l'administrateur réseau	5	J
H	Câblage	10	J
I	Commande de Novell Netware 5	4	D
J	Choix des fournisseurs	5	A
K	Mise à jour logicielle des postes	1	M
L	Mise à jour matérielle des postes	2	D
M	Installation Novell Netware 5	2	L I H G

Le cycle de vie d'un logiciel : Planification

DIAGRAMME DE GANTT : EXERCICES

Tracer le diagramme de gantt du projet suivant :

