

Partie théorique :

Exercice 1 : WBS (Work Breakdown Structure)

1. Qu'est-ce que WBS ?

R :

« Décomposition hiérarchique, axée sur les *livrables*, du *travail* que *l'équipe de projet* doit *exécuter* pour atteindre les *objectifs du projet* et produire les livrables voulus. »
(*Guide PMBOK*).

WBS est une décomposition successive d'une activité plus grande (le projet lui même) dans des activités plus petites.

La fiche de chacun des activités doit comporter les informations suivantes :

- un titre et une description de la tâche
- un responsable unique
- une durée d'exécution exprimée en jours ou en heures
- une description des ressources nécessaires à son exécution
 - les ressources humaines
 - les ressources matérielles
- un coût estimé

2. À quoi ça sert ?

R :

Nous avons besoin de WBS afin de faire des estimations de coût et de travail (effort) à faire et de développer un calendrier consistant.

Estimer le coût:

- Estimer le coût de toutes les activités
- Inclure le coût des éléments dans le coût total du système

Performance du calendrier:

- Savoir quelles activités sont finies
- Mesurer le progrès

3. Étude de cas simple :

La tâche consiste à implanter un mini compilateur d'un sous-ensemble de C qui offre également une interface usager pour écrire le code et le compiler.

- Identifier les activités du projet en précisant les types de décomposition
- Allouer des ressources à chaque activité
- Allouer du temps à chaque activité

R :

Compilateur C
Commencement du projet
Allouer les ressources
Etablir l'environnement du projet
Planifier la gestion du projet
Contrôle du projet
Identifier et analyser les risques
Composer le plan d'aversion de risques
Gestion de contrôle du projet
Définir les métriques
Gérer la qualité du logiciel
Besoins
Analyser les fonctions
Développer l'architecture de système
Développer les besoins et les spécifications
Définir les besoins sur l'interface
Conception
Concevoir l'interface graphique
Concevoir le parseur
Concevoir l'analyseur syntaxique et sémantique
Concevoir le générateur du code
Implantation et tests unitaires
Coder l'interface graphique et faire les tests unitaires
Coder le parseur et faire les tests unitaires
Coder l'analyseur syntaxique et sémantique et faire les tests unitaires
Coder le générateur du code et faire les tests unitaires
Intégration
Installer le compilateur
Tester l'interface graphique
Tester le parseur
Tester le générateur du code
Tester l'analyseur syntaxique et sémantique
Documentation
Écrire les documents d'utilisateur
Écrire les documents de développeur

Exercice 2: COCOMO (CONstructive COst MODEL)



COCOMO est constitué de 3 modèles.

Chaque modèle est plus évolué que le précédent et prend en compte plus de paramètre.

Modèle de Base

C'est le plus simple. Il estime l'effort (en mois-homme) en fonction du nombre de lignes de code que l'application doit contenir et en fonction de la complexité du projet :

1. « **Organic** » : applications simple et déterministes, pas de contraintes ou interactions, petites équipes..
2. « **semidetached** »: entre organique et imbriqué.
3. « **embedded** »: applications complexes et non déterministes avec contraintes et interactions nombreuses, comme pour les applications temps réel)

Une ventilation simple permet de définir l'effort nécessaire pour chaque partie du cycle de développement ;

Modèle Intermédiaire

Il reprend le calcul de l'effort et de la durée du modèle de base et y applique 15 facteurs de coûts.

Chaque facteur (de valeur nominative de 1), varie selon son importance dans le projet.

Les 15 facteurs sont multipliés pour donner un facteur d'ajustement - qui viendra modifier l'estimation de la formule de base.

Modèle Détaillé (ou expert)

Il reprend les données du modèle intermédiaire et affine les facteurs de coûts en fonction de chaque étape du cycle de développement.

Formules

Complexité	Effort (en mois homme)	Temps de développement (en mois)
S	$Effort = 2,4 * KLS^{1,05}$	$TDev = 2,5 * Effort^{0,38}$
P	$Effort = 3 * KLS^{1,12}$	$TDev = 2,5 * Effort^{0,35}$
E	$Effort = 3,6 * KLS^{1,2}$	$TDev = 2,5 * Effort^{0,32}$

1. COCOMO de base

En appliquant la méthode COCOMO estimer la taille moyenne de l'équipe qui faudrait prévoir pour développer un logiciel estimé à environ 40 000 instructions sources (SLOC), le projet est simple et l'équipe du développement est relativement réduite.

R :

Nous appliquons la méthode COCOMO et nous nous apercevons que c'est un projet organique. Nous avons donc pour le calcul de l'effort et la durée, les formules suivantes:

$$E = 2.4 \times KLOC^{1.05} \quad D = 2.5 \times E^{0.38}$$

Donc selon la formule de la charge: $E = 2.4 (40)^{1.05} \approx 115$ Personne-Mois

$D = 2.5 (115)^{0.38} \approx 15$ Mois

Ce qui nous donne: **Taille équipe = E / D = 115/15 \approx 7.6** soit 8 Personnes.

2. COCOMO intermédiaire

Soit à développer un logiciel de gestion d'un système de gestion de manutention dans un atelier d'assemblage de voiture (ateliers flexibles). Le système logiciel doit fonctionner sous des contraintes particulièrement fortes. Le système à développer est une partie d'un système complexe et fortement connecté de matériels et de logiciels se trouvant dans l'atelier entre autre le système de pilotage des robots. Des normes et des procédures opérationnelles surtout de sécurité doivent être prises en compte. En conséquence, les modifications de spécifications destinées à contourner des problèmes logiciels sont en général impossibles et les coûts de validation extrêmement élevés.

Nous avons calculé les PF de ce système. Cette tâche de comptage nous a coûté 2 jours de travail (5 heures/jour) ; la productivité de l'équipe d'estimation était de 200

PF/heures. Le système est développé avec les langages C et C++. Admettons qu'un PF correspond à 65 lignes de code C++ et 85 lignes de code C. On prévoit que 70% du système serait développé avec C++.

Les consignes données par les responsables de l'atelier sont les suivantes :

- Une défaillance pose de sérieux problème particulièrement de sécurité. Une défaillance peut mettre en péril la vie humaine.
- Le système fonctionne 16h/j et 65% de la puissance matérielle disponible sera utilisée.
- La taille de la base de données à utiliser (en octets) est entre 8 à 10 fois le nombre de lignes sources livrées.

Les conditions de développement se caractérisent par :

- Des outils CASE couvrant l'intégralité du cycle de vie sont disponibles.
- Méthode de programmation moderne, évoluée et expérimentée par l'équipe de développement.

La complexité du produit est très élevée à cause de traitement parallèle et gestion de données complexes.

- a) Après avoir déterminé le type de projet, calculer l'estimation de l'effort et de la charge ainsi que la taille moyenne de l'équipe en utilisant COCOMO de base.
- b) Identifier les facteurs qui influencent les estimations dans ce projet ainsi que leurs valeurs respectives (voir tableau ci-dessous).
- c) Calculer l'effort, la durée et la taille moyenne de l'équipe de développement en tenant compte des contraintes et consignes données dans le texte

1: Multiplicateurs d'attributs de projet

Attributs	Valeurs					
	TB	B	M	E	TE	TTE
FIAB	0,75	0,88	1,00	1,15	1,40	--
DONN	--	0,94	1,00	1,08	1,16	--
CPLX	0,70	0,85	1,00	1,15	1,30	1,65
TEMP	--	--	1,00	1,11	1,30	1,66
ESPA	--	--	1,00	1,06	1,21	1,56
VIRT	--	0,87	1,00	1,15	1,30	--
CSYS	--	0,87	1,00	1,07	1,15	--
APTA	1,46	1,19	1,00	0,86	0,71	--
EXPA	1,29	1,13	1,00	0,91	0,82	--
APTP	1,42	1,17	1,00	0,86	0,70	--
EXPV	1,21	1,10	1,00	0,90	--	--
EXPL	1,14	1,07	1,00	0,95	--	--
PMOD	1,24	1,10	1,00	0,91	0,82	--
OLOG	1,24	1,10	1,00	0,91	0,83	--
DREQ	1,23	1,08	1,00	1,04	1,10	--

R :

1) Type de projet :

En examinant les définitions et les caractéristiques suivantes des trois classes de projet :

a) Projets de mode organique : Ces projets sont réalisés par une équipe de taille relativement petite travaillant dans un environnement familier et dans un domaine d'application connu de l'équipe. En conséquence, le surcoût dû à la communication est faible, les membres de l'équipe savent ce qu'ils ont à faire et le font rapidement

b) Projets de mode semi-détaché : Ce mode représente un intermédiaire entre le mode organique et le mode embarqué décrit ci-dessous. Pour des projets de mode semi-détaché, l'équipe projet peut être composée de programmeurs de divers niveaux d'expérience. Les membres de l'équipe ont une expérience limitée de ce type de système. Ils peuvent être totalement inexpérimentés en ce qui concerne quelques-uns des aspects du système à développer, mais pas tous.

c) Projets de mode embarqué : La caractéristique principale d'un projet de mode embarqué est que le système doit fonctionner sous des contraintes particulièrement fortes. Le système à développer est une partie d'un système complexe et fortement connecté de matériels et de logiciels, de normes et de procédures opérationnelles. En conséquence, les modifications de spécifications destinées à contourner des problèmes logiciels sont en général impossibles et les coûts de validation extrêmement élevés. Du fait de la nature même de ces projets, il est habituel de disposer d'ingénieurs logiciels expérimentés dans le domaine d'application.

Nous concluons que le projet est de type embarqué vu sa complexité, ses contraintes fortes de sécurité et surtout sa forte connexion avec le matériel et les autres systèmes de l'atelier.

Donc les formules de l'effort et la durée sont les suivantes :

$$E = 3.6 \times KLOC^{1.2} \quad D = 2.5 \times E^{0.32}$$

Calcul de la taille de projet en PF

$$PF = 200 \times 5 \times 2 = 2000 \text{ PF}$$

Calcul de la taille de projet en KLOC

$$2000 \times 0.7 \times 65 + 2000 \times 0.3 \times 85 = 142000 \text{ LOC} = 142 \text{ KLOC}$$

$$\text{Calcul de l'effort et de la durée : } E = 3.6 \times (142)^{1.2} = 1377.363 \text{ PM}$$

$$D = 2.5 \times (1377.363)^{0.36} = 33.728 \text{ Mois}$$

Taille moyenne de l'équipe

$$E/D = 40.83 \text{ (41 Personnes)}$$

2) Les facteurs d'influences selon le texte sont :

Les multiplicateurs associés à ces attributs sont montrés à la table 1 qui est extraite de Boehm (1981). Notez que TB signifie très bas, B bas, M moyen, E élevé, TE très élevé et TTE très, très élevé.

1: Multiplicateurs d'attributs de projet						
Attributs	6c Valeurs					
	TB	B	M	E	TE	TTE
FIAB	0,75	0,88	1,00	1,15	1,40	--
DONN	--	0,94	1,00	1,08	1,16	--
CPLX	0,70	0,85	1,00	1,15	1,30	1,65
TEMP	--	--	1,00	1,11	1,30	1,66
ESPA	--	--	1,00	1,06	1,21	1,56
VIRT	--	0,87	1,00	1,15	1,30	--
CSYS	--	0,87	1,00	1,07	1,15	--
APTA	1,46	1,19	1,00	0,86	0,71	--
EXPA	1,29	1,13	1,00	0,91	0,82	--
APTP	1,42	1,17	1,00	0,86	0,70	--
EXPV	1,21	1,10	1,00	0,90	--	--
EXPL	1,14	1,07	1,00	0,95	--	--
PMOD	1,24	1,10	1,00	0,91	0,82	--
OLOG	1,24	1,10	1,00	0,91	0,83	--
DREQ	1,23	1,08	1,00	1,04	1,10	--

3) Calcul de l'effort, de la durée et de la taille moyenne de l'équipe (COCOMO intermédiaire)

$$E = E_{nominal} \times FA$$

$$FA = \prod_i FA_i$$

$$FA = FIAB \times DONN \times CLPX \times TEMP \times PMOD \times OLOG = 1.40 \times 0.94 \times 1.65 \times 1.11 \times 0.82 \times 0.83 = 1.62$$

$$E = 1377.363 \times 1.62 = 2238.87 \text{ Personne-Mois}$$

$$D = 2.5 \times (2238.87)^{0.36} = 40.17 \text{ MOIS}$$

Taille moyenne de l'équipe :

E/D = 55.7 (56 Personnes)

P(Productivité) :

Size(KLOC)/Effort = 142/2238.87 = 0.06342 KLOC/PM = 63.42 LOC/PM

Partie pratique:

Durant la séance, nous présenterons une démonstration de différents outils (open source) pour la planification de projets utilisés dans l'industrie:

1. *Open Workbench 1.1.6*

Open Workbench est une application libre de gestion et de planification de projet. Il s'agit d'un équivalent libre de Microsoft Project possédant une interface simple et de nombreuses fonctionnalités.

Il permet de définir le cycle de vie d'un projet à travers une série de tâches. Il permet notamment de faire un :

- découpage en activités (WBS - work breakdown structure) avec des tâches et des jalons.
- diagramme de GANTT du projet

Il est disponible en téléchargement libre sur : <http://www.openworkbench.org>

2. *GanttProject 2.0.10*

GanttProject permet la planification d'un projet à travers la réalisation d'un diagramme de Gantt. L'outil permet de créer des diagrammes de Gantt, des diagrammes de ressources et des réseaux PERT. Nous donnerons aussi, durant la démo, une brève description théorique sur les digrammes de Gantt et les réseaux de PERT.

Il est disponible en téléchargement libre sur : <http://www.ganttproject.biz/download>

3. *OpenProj 1.4*

OpenProj est une solution idéale pour remplacer des applications comme Microsoft Project. D'ailleurs, il permet d'ouvrir des fichiers créés avec ce logiciel et d'exporter au format XML, en facilitant ainsi la migration de l'un à l'autre. **OpenProj** a des fonctions multiples et des outils pour créer divers types de graphiques avec la distribution et l'attribution des recours dans le temps (par exemple, diagrammes Gantt et de PERT), et même pour calculer un prix et le profit.

Il est disponible en téléchargement libre sur : <http://openproj.org/>