

CHAPITRE 3:

ESTIMATION DUREE ET COUTS DU PROJET.

I. INTRODUCTION

Parmi les différentes étapes de gestion d'un projet, **l'évaluation du coût du projet est la plus importante et la plus délicate**. En effet, cette phase nécessite beaucoup de rigueur et doit être la plus exhaustive possible afin de fournir un chiffrage précis, fiable et pertinent.

Le but est d'établir le coût potentiel du projet, de son lancement jusqu'à sa clôture. Cette phase n'est pas simple et il est essentiel d'impliquer l'ensemble des parties prenantes pour le faire.

L'évaluation du coût d'un projet a lieu bien avant le lancement du projet et est étroitement liée à l'estimation des délais. Une fois qu'on a estimé le coût de la réalisation du projet et que celui-ci est validé, On doit définir le budget du projet. Bien estimer les coûts du projet, nous aide à le budgétiser correctement et à le piloter de la meilleure façon possible.

Il est indispensable pour le client comme pour le fournisseur (chef de projet et management) d'estimer à l'avance la durée (calendrier) et le coût (effort) d'un projet logiciel. Il convient également d'en mesurer les risques en recensant les dépendances extérieures qui influenceront sur les coûts et délais.

Une remarque préalable s'impose quant au processus d'estimation: Beaucoup trop souvent, le manager et les clients ont de la peine à comprendre que l'estimation est une activité comportant des difficultés intrinsèques et la rendent encore plus difficile par ce manque de compréhension. Le processus requiert des raffinements successifs, il est impossible de fournir une estimation correcte jusqu'à ce que le logiciel soit appréhendé dans son intégralité.

L'estimation des coûts et durée se fait en trois (03) étapes:

- ✓ Lors d'une réponse à un appel d'offres où il s'agit de fournir au plus vite une réponse adaptée au marché,
- ✓ Lors de la planification du projet où il s'agit d'établir le plan projet et le plan qualité qui serviront de cadre contractuel au projet,
- ✓ Lors du déroulement du projet afin d'affiner les prévisions et de les mettre à jour.

II. PROCESSUS D'ESTIMATION

La meilleure façon d'évaluer le montant à proposer dans un devis est sans doute de connaître le budget que le client est prêt à lui consacrer. Il est toutefois peu probable que le client ait la naïveté de faire part de ses prévisions de dépenses.

L'estimation du coût total d'un projet logiciel comprend le coût de développement du logiciel et du matériel, le coût de la formation, le coût des outils.

On prendra le cas du coût de développement du logiciel comme exemple à mieux étalé, celui-ci sera calculé en fonction du temps passé à développer le logiciel par les

ingénieurs dont on connaît le tarif horaire et le taux d'implication dans le projet en termes de pourcentage de leur temps.

Les coûts du développement logiciel sont donc étroitement liés à la notion de productivité des ingénieurs logiciels, c'est à dire le nombre de lignes de code (validées) produites par unité de temps.

A noter qu'à ces coûts salariaux chargés, il ne faudra pas oublier de rajouter les coûts fixes selon un mode de calcul propre à chaque entreprise. Donc lorsque nous parlerons de coûts, nous entendrons bien sur l'effort en termes d'hommes/mois.

Une méthode raisonnable pour établir une réponse à un appel d'offres, est de faire appel à des experts qui vont procéder par analogie avec des projets déjà achevés. On évalue chaque élément important du nouveau système en termes de pourcentage de coût d'un élément comparable dans le système achevé. L'estimation du coût total du nouveau projet est obtenue par sommation des estimations élémentaires.

L'estimation est un exercice difficile, impossible en théorie et faisant donc appel à des heuristiques. Dans de trop nombreux cas, l'estimation préalable est faite à la hâte dans le but de remporter un marché, la planification qui s'ensuit est souvent pénalisée par une sous-estimation initiale des coûts et délais globaux.

Il faut éviter de donner une estimation qui s'appuie sur un chiffre unique mais plutôt d'encadrer les prédictions par une fourchette optimiste/pessimiste (voir tableau ci-dessous).

Ce tableau donne une base d'estimation pour passer d'un chiffre unique à une fourchette grâce à des coefficients multiplicateurs.

Tableau.1 : Coefficients multiplicateurs pour chaque phase du projet. (Adapté de "Cost Models for Future Software Life Cycle Processes: COCOMO 2.0", Bohem et al. 1995)

Phase	Taille et effort		Durée	
	Optimiste	Pessimiste	Optimiste	Pessimiste
Analyse initiale des besoins	0.25	04	0.6	1.6
Définition approuvée des besoins	0.5	02	0.8	1.25
Spécification des besoins	0.67	1.5	0.85	1.15
Conception globale	0.80	1.25	0.9	1.1
Conception détaillée	0.90	1.10	0.95	1.05

Pour utiliser les facteurs multiplicatifs du tableau, il suffit de multiplier le point d'estimation unique par les coefficients appropriés de manière à obtenir une fourchette d'estimation dont on peut remarquer qu'elle se resserre au fur et à mesure que l'on avance dans les phases (0.90 à 1.10 en conception détaillée contre 0.25 à 4 en analyse initiale des besoins, donc au moment de la réponse à l'appel d'offres).

Plutôt que d'estimer la durée du projet dans son ensemble on peut le décomposer et estimer chacun de ses composants. Nous verrons que cette technique s'applique très bien à l'approche objets du fait de l'indépendance des différents objets entre eux.

❖ Quelques règles d'or pour l'estimation

- **Eviter de devenir**: Prendre le temps de réfléchir, ne jamais répondre sans avoir étudié la situation calmement,

- **Prévoir du temps** pour l'estimation et la planifier,
- Utiliser des données de **projets précédents**,
- Prendre **l'avis des développeurs** qui vont effectivement effectuer le travail,
- Estimer par **consensus** : consulter les différentes équipes, converger sur des dates au plus tôt et au plus tard,
- Estimer par **difficultés** (facile, moyen, difficile),
- Ne pas oublier **les tâches récurrentes** : documentation, préparation des démonstrations utilisateurs, formation utilisateurs et développeurs, intégration à l'existant, récupération des données, revues, réunions, maintenance de l'existant pendant le développement du nouveau produit, gestion qualité, absentéisme (congés maladie, TRR),
- Utiliser plusieurs techniques d'estimation,
- Changer de méthodes d'estimation,
- Prendre en compte les risques de gestion dans l'estimation.

De manière générale, le processus d'estimation passe par 03 étapes :

1. Estimer la taille du produit (nombre de lignes de code ou points de fonctions),
2. Estimer l'effort (en homme mois),
3. Estimer la durée (en mois ou semaines calendaires).

1. ESTIMATION DE LA TAILLE VIA LES POINTS DE FONCTION (ALBRECHT 79)

L'estimation des coûts passe comme on l'a cité au part avant par l'estimation de la taille et de la productivité.

Dans un premier temps on doit définir la productivité en terme de nombre de lignes de code source produites par unité de temps, cette métrique simple présente cependant certains inconvénients:

- L'écriture de lignes de code représente une toute petite partie du développement,
- Pour des langages différents un même logiciel se code avec un nombre de lignes différent,
- Comment compter le nombre de lignes (commentaires)?
- On écrit souvent du code non livré (outils de développement),
- Le nombre de lignes ne peut être compté qu'a posteriori.

Un autre moyen de mesurer où la productivité consiste à évaluer le nombre de fonctionnalités intéressantes produites par unités de temps. La méthode des points de fonction, définie chez IBM par Albrecht en 1979, puis raffinée en 83 propose ainsi d'estimer la taille d'un logiciel à partir de valeurs connues tôt dans le cycle de vie et qui sont indépendantes du langage choisi pour l'implémentation. La méthode est aujourd'hui défendue et mise à jour par l'IFPUG (the International Function Point User Group) qui publie the Function Point Counting Practices Manual.

Le nombre de points de fonctions d'un programme est basé sur le nombre et la complexité de cinq paramètres:

1.1. Les données

1.1.1. Entrées externes (External Inputs EI)

Ces données peuvent venir d'un écran ou d'une autre application, elles peuvent être utilisées pour mettre à jour un ou plusieurs fichiers internes, il peut s'agir d'informations ou de

données de contrôle. Les écrans, boîtes de dialogue, messages à travers lesquels un utilisateur entre une donnée fait partie des entrées externes.

1.1.2. Sorties externes. (External outputs EO)

Ce sont les sorties d'un programme, elles peuvent prendre la forme de fichiers de sortie envoyés à d'autres application, d'écrans, de rapports de graphes destinés à l'utilisateur final; ils sont créés à partir de fichiers logiques internes.

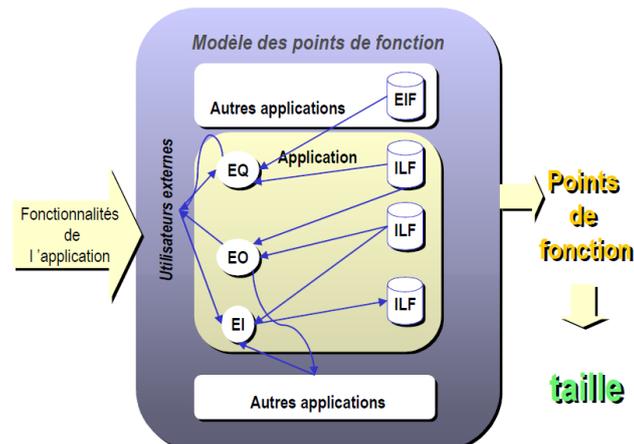


Fig. 1: Modèle des points de fonction

1.2. Les transactions

1.2.1. Requêtes externes (External Inquiry EQ)

Un procédé élémentaire mettant en jeu des entrées et des sorties qui résulte en la production de données provenant de plus d'un fichier logique interne. Le processus ne met pas à jour de fichier interne, combinaison d'entrées /sorties où le résultat apparaît sous forme simple et immédiate, généralement en utilisant une seule clé d'accès. La frontière entre requête externe et donnée externe est assez floue. Nous dirons que les requêtes concernent exclusivement les accès à une base de données alors que les données externes combinent ces accès avec des calculs et traitements plus ou moins complexes.

1.2.2. Fichiers logiques internes (Internal Logical Files ILF's)

Un groupe de données corrélées entièrement résidant dans les limites de l'application, mis à jour par les données externes, entièrement contrôlé par le programme. Par exemple un fichier, une table dans une base de données.

1.2.3. Fichier interfaces externes (External Interface Files EIF's)

Un groupe de données corrélées utilisées pour des références seulement. Les données résident entièrement hors de l'application et sont mises à jour par une autre application (c'est donc un ILF pour une autre application).

Après que les composants du projet aient été ainsi classifiés, chaque paramètre est affecté d'un score : faible, moyen, élevé. Pour les transactions (EI's, EO's, EQ's) le score est calculé sur le nombre de fichiers mis à jour ou référencés (FTR's) et le nombre de types de données impliqués (DET's).

FTR's	Données		
	1 - 4	5 - 15	>15
0 – 1	Faible	Faible	Moyen
2	Faible	Moyen	Haut
>2	Moyen	Haut	Haut

FTR's	Éléments de Données		
	1 - 5	6 - 19	>19
0 – 1	Faible	Faible	Moyen
2 – 3	Faible	Moyen	Haut
>3	Moyen	Haut	Haut

RET's	Data Elements		
	1 - 19	20 - 50	>50
1	Faible	Faible	Moyen
2 – 5	Faible	Moyen	Haut
>5	Moyen	Haut	Haut