

Gestion de Projet Informatique

Partie 3 : Revue des charges et gestion de qualité

Licence d'Informatique 3^e Année
Tianxiao Liu
CY Cergy Paris Université

Sommaire

- Principes de l'estimation des charges
- Méthode d'estimation « point de fonction »
- Qualité de projet avec le modèle Mc Call

Objectif d'estimation des charges

- Une phase incontournable → *Planification*
 - Définir l'ensemble des tâches à réaliser
 - Quantifier la charge associée à ces tâches
- Résultats
 - Fixer une date de fin de projet (cas général)
 - Justifier des besoins en ressources humaines
 - Prioriser la mise en place de fonctionnalités
 - Organiser le travail de l'équipe

Problématique

- Les projets sont souvent **sous-estimés !**
 - Les dépassements d'échéances proviennent plus souvent d'une mauvaise estimation que d'une mauvaise réalisation
- **Pourquoi sous-estime-t-on?**
 - L'expérience limitée des acteurs
 - Le manque de capitalisation sur les expériences
 - L'irrégularité du suivi des projets
 - L'incapacité à estimer les impacts de modifications
 - **La volonté de donner une bonne image (plaire, besoin de gagner, optimisme, etc.)**

Une estimation adaptée

- Au fur et à mesure de l'avancée du projet, l'estimation pourra (devra) être revue → nouveaux facteurs identifiés
- **Facteurs susceptibles d'affecter l'exactitude des estimations**
 - Toutes les données utilisées pour estimer
 - L'exactitude de tous les calculs
 - L'expérience de l'équipe sur les technologies utilisées
 - Le cycle de vie utilisé dans le cadre du projet
 - Le contexte (rigueur de planification, conduite et suivi) de management du futur projet
 - Les incidents majeurs susceptibles d'apparaître en cours de projet

Méthode « points de fonction »

- Principe

- Une description des fonctions du projet
- **3** niveaux de complexité et **5** classifications
- Estimation de point de fonction brut (PFB)

Composant fonctionnel	Niveau de complexité		
	Faible	Moyen	Elevé
Entrées	3	4	6
Interrogations	3	4	6
Sorties	4	5	7
Interface	5	7	10
Données logiques	7	10	15

Méthode « points de fonction »

- Explication des composants
 - **Entrées** : Introduction de données dans le système
 - **Interrogations** : uniquement des consultations de données
 - **Sorties** : Restitution de données calculées aux utilisateurs
 - **Interface** : échange de données en import ou en export avec d'autres systèmes externes
 - **Données logiques** : Entités participants créées, utilisées ou gérées par le système → architecture N-tiers ($N \geq 2$)

Méthode « points de fonction »

- Démarche pour **chaque tâche**
 - Identification et dénombrement des composants du projet
 - Déterminer du niveau de complexité de chaque composant
 - Valorisation des composants
 - Totalisation des points de fonction bruts (PFB)
- Calcul
 - PFA (Points de fonction ajustés)
 - **$PFA = PFB \times FTA$**
 - **$Charge = PFA \times Effort\ unitaire$**
 - Voir l'explication de **FTA** et **Effort unitaire** plus loin

Méthode « points de fonction »

- Effort unitaire

Phase projet	Contexte d'utilisation	Effort unitaire
Expression des besoins *	Petit projet	2 unités de temps
	Moyen projet	3 unités de temps
	Grand projet	4 unités de temps
Conception détaillée *	Cas général	1,5 unités de temps
Réalisation avec un L3G ou L4G	par point de fonction	0,1 unité de temps x nb point

* A ajouter systématiquement dans le calcul.

Méthode « points de fonction »

- FTA : Facteurs d'ajustement
 - Communication des données
 - Distribution des données ou des traitements
 - Performance requise
 - Intensité d'utilisation de la configuration matérielle
 - Taux de transition
 - Taux de transaction
 - Saisie interactive
 - Convivialité d'exécution

Méthode « points de fonction »

- FTA (suite)
 - Mise à jour en temps réel
 - Complexité des traitements
 - Réutilisation du code
 - Facilité d'installation
 - Facilité d'exploitation
 - Portabilité
 - Facilité d'adaptations

Méthode « points de fonction »

- Pondération des FTA

- Notation de chaque facteur de 0 à 5 selon son influence sur le projet

0	Nulle (facteur non présent)
1	Très faible (insignifiante)
2	Faible ou modérée
3	Moyenne
4	Forte (significative)
5	Très forte (essentielle)

- TFAP (Total des Facteurs d'Ajustement Pondérés)

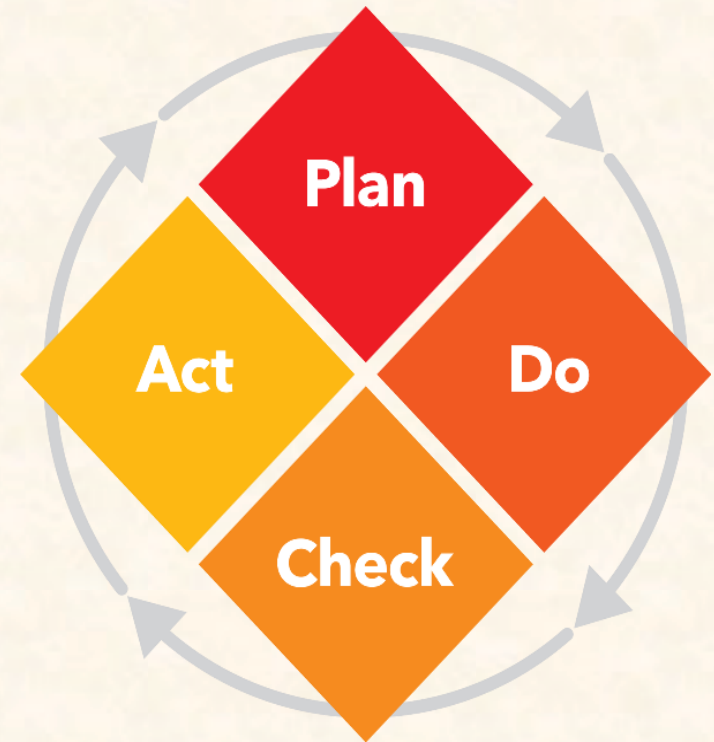
- **$FTA = 0,65 + (TFAP/100)$**

Esprit de qualité de projet

- Principe de base
 - *On dit ce que l'on fait et on fait ce que l'on a dit.*
- Une tâche parfois difficile dans un projet
 - On est souvent **réticent** à la mise en place d'une politique de qualité
 - Contrôles parfois perçus comme une **surveillance** du travail des membres de l'équipe...
- Solution envisagée
 - La qualité est bien l'affaire de tous et toute l'équipe doit être impliquée

Assurance de qualité : démarche

- **PLAN** : écrivez ce que vous faites (définissez qui, quoi, où, quand, comment assurer la qualité)
- **DO** : faites ce que vous avez écrit
- **CHECK** : vérifiez ce que vous avez fait est conforme à ce que vous avez écrit
- **ACT** : validez



Modèle Mc Call : Exploitation

- **Conformité** par rapport aux besoins (l'application répond-elle aux besoins des utilisateurs ?)
- **Fiabilité** (l'application fonctionne-t-elle correctement dans tous les cas ?)
- **Efficacité** (utilisation minimum des ressources, c'est-à-dire temps, mémoire ...)
- **Intégrité** (l'application est-elle bien protégée, le niveau de sécurité est-il suffisant ?)
- **Facilité d'emploi** (mise en œuvre, prise en main)

Modèle Mc Call : Evolution et Adaptabilité

- **Maintenabilité** (est-il facile de localiser et de corriger les erreurs ?)
- **Souplesse** (facilité de modification et d'évolution)
- **Testabilité** (quels efforts à fournir pour tester le système ?)
- **Portabilité** (le système est-il utilisable sur une autre machine ?)
- **Réutilisabilité** (peut-on reprendre certaines parties du projet et les intégrer dans un autre logiciel ?)
- **Interopérabilité** (peut-on interfacier l'application avec un autre système ?)