

Estimation des charges

Jolita Ralyté
ralyte@cui.unige.ch



Plan

- Charge du projet
- Durée du projet
- Besoins d'estimation
- Méthodes d'estimation



Charge du projet

■ La charge

- représente une quantité de travail nécessaire, indépendamment du nombre de personnes qui vont effectivement réaliser ce travail
- s'exprime en **jour/homme**, **mois/homme** ou **année/homme**
1 mois/homme représente l'équivalent du travail d'une personne pendant un mois (20 jours)
- permet d'obtenir un coût prévisionnel

Exemple :

- si 1 mois/homme coûte 10 000 F
- et le projet est estimé à 60 mois/homme
- ➡ Le coût du projet 600 000 F



Charge du projet

■ La taille de projet

- La charge < 6 mois/homme – un très petit projet
- La charge **entre 6 et 12** mois/homme – un petit projet
- La charge **entre 12 et 30** mois/homme – un projet moyen
- La charge **entre 30 et 100** mois/homme – un grand projet
- La charge > 100 mois/homme – un très grand projet (souvent mesuré en année/homme)

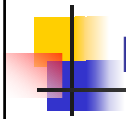


Durée du projet

- La durée dépend du nombre de personnes travaillant dans le projet

Exemple: 60 mois/hommes:

- 1 personne pendant 5 ans
- 5 personnes pendant 1 an
- 10 personnes pendant 6 mois
- 60 personnes pendant 1 mois
- Il est important de respecter un rapport charge/durée



Besoins d'estimation

Les besoins d'estimation de la charge se situent à différents niveaux :

- Projet
 - Ex.: *Projet de mise en place d'un SI gestion de stock*
- Etape dans la réalisation d'un projet
 - Ex.: *Etapes du projet: Etude préalable, Analyse des besoins, Conception, etc.*
- Phase d'une étape
 - Ex.: *Phases de l'Analyse des besoins : Identification des fonctionnalités, Spécification des fonctionnalités, Analyse des contraintes, etc.*
- Tâche dans une phase
 - Ex.: *Tâches de la Spécification des fonctionnalités : Spécification de la fonctionnalité 'Commande de fournitures', Spécification de la fonctionnalité 'Rupture du stock'*



Besoins d'estimation : *Niveau projet*

- Estimation de la charge du projet complet, de l'étude préalable à la mise en œuvre
- Mesure: mois/homme
- Objectifs :
 - Déterminer le budget du projet
 - Déterminer les efforts nécessaires
 - Estimer la rentabilité d'investissement
 - Evaluer la durée du projet



Besoins d'estimation : *Niveau étape*

- Estimation de la charge d'une étape spécifique
Ex.: l'étape Analyse des besoins
- Mesure : mois/homme ou semaine/homme
- Objectifs :
 - Ajuster le découpage : si la charge de l'étape est importante il est préférable de la diviser en deux sous-projets qui pourrait être livrés à des dates différentes
 - Sous-traiter : fournisseur et client font chacun une estimation
 - Prévoir des ressources, pour planifier l'affectation d'intervenants (internes ou externes) sur le projet



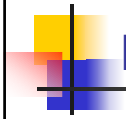
Besoins d'estimation : *Niveau phase*

- Estimation de la charge d'une phase spécifique (on peut prévoir plusieurs phases dans une étape)
 - Ex.: Phases de l'étape *d'Analyse des besoins* : *Identification des fonctionnalités, Spécification des fonctionnalités, Analyse des contraintes, etc.*
- Mesure : semaine/homme ou jour/homme
- Objectifs :
 - Faire une planification précise
 - Etablir un calendrier de livraison des différents résultats intermédiaires
 - Prévoir et effectuer un suivi du projet ou sous-projet, pour surveiller les écarts
 - Prévoir l'affectation des ressources



Besoins d'estimation : Niveau tâche

- Estimation de chacune des tâches qui font généralement l'objet d'une affectation individuelle
 - Ex.. Tâches de l'étape *Spécification des fonctionnalités* :
 - *Spécification de la fonctionnalité 'Commande de fournitures'*
 - *Spécification de la fonctionnalité 'Rupture du stock'*
- Mesure : jour/homme
- Objectifs :
 - Faire une planification au niveau le plus fin qui est indispensable pour le suivi du travail d'équipe



Méthodes d'estimation

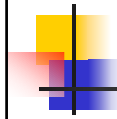
- Les méthodes basées sur le **jugement d'expert**
- Les méthodes de **répartition proportionnelle**
- Les méthodes à base de **modèles**
 - Les modèles sont basés sur une unité d'œuvre
 - Déterminer une ou plusieurs unités d'œuvre
 - Affecter un coût unitaire standard à chaque unité d'œuvre



Méthode Delphi

Delphi - une méthode basée sur le jugement d'expert

- Utilisée principalement au **niveau projet**
- Consiste à rechercher des analogies avec des projets antérieurs:
 1. Comparer le projet avec les projets analogues
 2. Identifier les références internes sur un autre domaine
 3. Prendre en compte les facteurs d'environnement



Méthode Delphi

Exemple : projet de mise en place d'un SI de gestion des clients dans une banque A

1. Comparer le projet avec les projets analogues

Projet analogue mené dans une banque B a coûté 500 mois/homme. L'entreprise utilisait une base de données DB2 pour la première fois

Projet analogue mené dans une banque C a coûté 350 mois/homme

2. Identifier les références internes sur un autre domaine

Projet de gestion de carte bancaire a coûté 700 mois/homme

Projet de gestion de crédits a coûté 1800 mois/homme

3. Prendre en compte les facteurs d'environnement

Gestion des clients dans une petite banque d'affaires coûtera moins que dans une grande banque à réseau à cause du nombre de décideurs, de la taille et de l'organisation de la fonction informatique

©J. Ralyté, Université de Genève

13




Méthode Delphi

La démarche :

- Chaque expert propose une estimation en utilisant sa propre expérience
- Tous les jugements sont rendus publics, mais restent anonymes
 - Chaque expert peut modifier sa propre estimation ou la confirmer
 - Chaque expert doit considérer que tous les autres sont également des experts
- Les nouvelles estimations sont dévoilées et chacun peut justifier son propre jugement
- Chaque expert propose une révision de son estimation

©J. Ralyté, Université de Genève

14




Méthode de répartition proportionnelle

La méthode de **répartition proportionnelle** est focalisée sur les problèmes de répartition à partir de ratios standard

- Elle s'appuie sur le découpage temporel de référence
- Elle est utilisée au **niveau d'étape**
- Trois façons d'utiliser :
 - Utilisation descendante :
 - Estimation de la charge globale du projet
 - Répartition de cette charge dans le temps
 - Utilisation ascendante :
 - Une étape à été évaluée au moyen d'une autre méthode,
 - Ceci permet de déduire la charge des autres étapes
 - Utilisation dynamique
 - En cours de déroulement du projet on observe le temps consommé sur les étapes amont
 - Ceci permet d'estimer les étapes à venir

©J. Ralyté, Université de Genève 15



Méthode de répartition proportionnelle

Répartition proportionnelle de la charge entre les étapes :

Etape	Ratio
Etude préalable	10 % du total du projet (hors mise en œuvre)
Etude détaillée	20 à 30 % du total du projet
Etude technique	5 à 15 % de la charge de réalisation
Réalisation	2 fois la charge d'étude détaillée
Mise en œuvre	30 à 40 % de la charge de réalisation

- Les ratios sont issus d'expériences positives et négatives de différents projets
- Ils doivent être considérés en partie comme des recommandations en partie comme des règles
- Il faut les prendre comme base, en comprenant les relations qui unissent les différentes étapes

©J. Ralyté, Université de Genève 16

Méthode de répartition proportionnelle

On utilise également une répartition proportionnelle entre les phases d'une étape

Phase	Pourcentage de la charge d'étude préalable
Observation	30 à 40 %
Conception	40 à 50 %
Appréciation	10 %

La charge de la phase d'observation est proportionnelle au nombre d'acteurs de l'entreprise qu'il faut interroger

- Identifier les services concernés
- Etablir la liste des interlocuteurs à rencontrer ($i = 1 \text{ à } N$)
- Affecter un poids p_i à chaque interlocuteur i (p_i entre 0.5 et 3 jours)

**Charge de la phase observation = Somme $\{p_i\}$
 $i = 1 \text{ à } N$ et p_i compris entre [0.5 jours; 3 jours]**

©J. Ralyté, Université de Genève 17


Méthodes d'estimation à base de modèles

```

graph LR
    PT[Projets terminés] --> BPR[Base de projets réalisés]
    BPR --> UO[Unité d'œuvre]
    BPR --> C[Correcteurs]
    BPR --> PS[Poids standards]
    BPR --> R[Ratios]
    UO --> T[Taille]
    C --> A[Ajustement]
    PS --> CH[Charge]
    R --> RE[Répartition]
    T --> CH
    A --> CH
    CH --> RE
    CP[Caractéristiques du projet] --> T
  
```

- Construire une base de connaissances rassemblant l'expertise des projets antérieurs
- Faire une estimation de la taille du projet à l'aide d'une unité de mesure
- Convertir la taille en charge
- Ajuster la taille ou la charge en raison de certaines particularités du projet
- Répartir la charge dans les différentes étapes


©J. Ralyté, Université de Genève 18



Méthode basée sur le modèle Cocomo

- **Cocomo** – *Constructive Cost Model* – modèle de construction des coûts
- Ce modèle est basé sur deux hypothèses :
 - Un informaticien sait plus facilement donner une évaluation de la taille du logiciel à développer que faire une estimation du travail nécessaire
 - Il faut toujours le même effort pour écrire un nombre donné de lignes de programme, quel que soit le langage employé
- Unité d'œuvre – **instruction source** (ligne de programme)
 - Il faut connaître le nombre présumé de lignes de programme en langage source

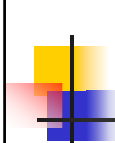
©J. Ralyté, Université de Genève 19



Méthode basée sur le modèle Cocomo

- **Charge** = $a(kisl)^b$ mois/hommes
- **Délai normal** = $c(charge)^d$ mois
 - $kisl$ = nombre de milliers d'instructions sources livrées
 - a , b , c et d – des paramètres dont les valeurs dépendent de la catégorie du projet
- Trois catégories de projet :
 - **Simple** – le logiciel comporte moins de 50000 instructions, les spécifications sont stables, le développement effectué par une petite équipe
 - **Moyen** – le logiciel comporte entre 50000 et 300000 instructions
 - **Complexe** – le logiciel comporte plus de 300000 instructions, une équipe est nombreuse, le domaine est nouveau

©J. Ralyté, Université de Genève 20



Méthode basée sur le modèle Cocomo

Les valeurs des paramètres :

Type projet	Charge en mois/homme	Délai en mois
Simple	Charge = $3,2 \text{ (kisl)}^{1,05}$	D = $2,5 \text{ (Charge)}^{0,38}$
Moyen	Charge = $3 \text{ (kisl)}^{1,12}$	D = $2,5 \text{ (Charge)}^{0,35}$
Complexe	Charge = $2,8 \text{ (kisl)}^{1,2}$	D = $2,5 \text{ (Charge)}^{0,32}$

Exemple : un projet moyen visant à développer un logiciel estimé à 40000 instructions :

Charge = $3,2 \text{ (40)}^{1,05} = 154 \text{ mois/homme}$

Délai normal = $2,5 \text{ (154)}^{0,38} = 17 \text{ mois}$

Taille moyenne de l'équipe = $154/17 = 9 \text{ personnes}$



Méthode basée sur le modèle Cocomo

- **Facteurs correcteurs** – permettent de préciser la charge du projet en fonction de ses caractéristiques :
 - Les exigences attendues du logiciel
 - Les caractéristiques de l'environnement technique
 - Les caractéristiques de l'équipe projet
 - L'environnement du projet
- **Charge nette = Charge brute * facteurs correcteurs**



Méthode basée sur le modèle Cocomo

Facteurs correcteurs :

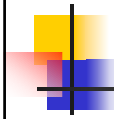
Logiciel	Personnel
Fiabilité	Compétences des concepteurs
Base de données	Expérience des concepteurs
Complexité	Compétences des développeurs
Temps d'exécution	Connaissance de l'environnement technique
	Expérience du langage
Matériel	Projet
Taille mémoire	Utilisation d'une méthode
Stabilité de l'environnement	Utilisation d'un atelier de génie logiciel
	Contrainte de délai



Méthode basée sur le modèle Cocomo

Démarche :

1. Estimer le nombre d'instructions sources
2. Calculer la charge 'brute'
3. Sélectionner les facteurs correcteurs
4. Appliquer les facteurs correcteurs à la charge brute pour obtenir la charge 'nette'
5. Evaluer le délai normal



Méthode d'évaluation analytique

- La méthode **d'évaluation analytique** s'appuie sur une typologie des programmes à développer - les unités d'œuvre sont les **différents types de programmes**
- Les poids (en jours/homme) affectés aux unités d'œuvre varie en fonction de l'environnement et de l'équipe de développement
- La charge obtenue, appelée **charge de réalisation** couvre :
 - la programmation,
 - l'élaboration des tests et
 - la mise au point des programmes
- La charge complète intègre :
 - Les testes : 10 % de la charge de réalisation
 - L'encadrement : 20 % de la charge de réalisation



Méthode d'évaluation analytique

Exemple de typologie pour l'évaluation analytique

Type de transaction	Facile	Moyen	Difficile
Menu	0,25	0,5	1
Consultation	1	2,5	4
Mise à jour	1,5	3	5
Edition en temps réel	1	2	4

Type de programme	Facile	Moyen	Difficile
Extraction	0,5	1	1,5
Mise à jour	2	3	5
Edition	1,5	2,5	4



Méthode d'évaluation analytique

Exemple d'évaluation analytique

Complexité	Facile			Moyen			Difficile			Total
Type de transaction	Nb	Poids	Charge	Nb	Poids	Charge	Nb	Poids	Charge	
Menu	5	0,25	1,25	2	0,5	1	0	1	0	2,25
Consultation	2	1	2	4	2,5	10	3	4	12	24
Mise à jour	3	1,5	4,5	2	3	6	2	5	10	20,5
Edition en temps réel	0	1	0	3	2	6	0	4	0	6

Complexité	Facile			Moyen			Difficile			Total
Type de programme	Nb	Poids	Charge	Nb	Poids	Charge	Nb	Poids	Charge	
Extraction	0	0,5	0	4	1	4	0	1,5	0	4
Mise à jour	3	2	6	6	3	18	1	5	5	29
Edition	2	1,5	3	3	2,5	7,5	5	4	20	30,5



Méthode d'évaluation analytique

Exemple d'évaluation analytique

Complexité	Facile			Moyen			Difficile			Total
Type de transaction	Nb	Poids	Charge	Nb	Poids	Charge	Nb	Poids	Charge	
Menu	5	0,25	1,25	2	0,5	1	0	1	0	2,25
Consultation	2	1	2	4	2,5	10	3	4	12	24
Mise à jour	3	1,5	4,5	2	3	6	2	5	10	20,5
Edition en temps réel	0	1	0	3	2	6	0	4	0	6
Type de programme	Nb	Poids	Charge	Nb	Poids	Charge	Nb	Poids	Charge	
Extraction	0	0,5	0	4	1	4	0	1,5	0	4
Mise à jour	3	2	6	6	3	18	1	5	5	29
Edition	2	1,5	3	3	2,5	7,5	5	4	20	30,5
Hors classification							1	7	7	7
Total réalisation										123,25
Testes								10 %		12,33
Encadrement								20 %		24,66
Total projet										160,25



Méthode des points fonctionnels

- Méthode des **points fonctionnels** – estimation à partir d’une description externe du futur système, de ses **fonctions**
 - Cinq types d’unité d’œuvre
 - Trois degrés de complexité
 - A chaque type et chaque degré est affecté un nombre de points
- La méthode permet de calculer le poids du projet en **points de fonction**



Méthode des points fonctionnels

Démarche :

1. Calculer la taille du projet en nombre de points de fonction brut
2. Ajuster la taille du projet
3. Transformer le nombre de points de fonction en charge



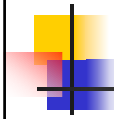
Méthode des points fonctionnels

1. Calculer la taille du projet en nombre de points de fonction brut (PFB)
 - Composants fonctionnels servant d'unité d'œuvre :
 - Relatifs aux données :
 - Groupe logique de données internes (GDI)
 - Groupe logique de données externes (GDE)
 - Relatifs aux traitements :
 - Entrée (ENT)
 - Sortie (SOR)
 - Interrogation (INT)



Méthode des points fonctionnels

- Un GDI est un groupe de donnée que l'utilisateur perçoit comme logiquement liées. Il est créé et mis à jour à l'intérieur du domaine d'étude
 - Ex.: une classe dans un modèle objet (classe *Personne*), ou une association ayant des propriétés qui correspondent à un objet de gestion
- Un GDI est composé des données élémentaires (DE) qui correspondent aux propriétés d'un GDI
 - Ex.: les propriétés de la classe *Personne*: *Nom*, *Prénom*, *Adresse*
- On peut identifier plusieurs sous-ensembles logiques de données (SDL) à l'intérieur d'un GDI
 - Ex.: les sous-classes *Employé* et *Client* de la classe *Personne*
- La complexité d'un GDI est fonction de nombre de DE et du nombre de SDL



Méthode des points fonctionnels

Complexité des GDI

	1 - 19 DE	20 - 50 DE	51 DE ou plus
1 SDL	Faible	Faible	Moyenne
2 – 5 SDL	Faible	Moyenne	Elevée
6 SDL ou plus	Moyenne	Elevée	Elevée

Le nombre de points de fonction correspondant au degré de complexité

Complexité	Faible	Moyenne	Elevée
Nombre de points de fonction du GDI	7	10	15



Méthode des points fonctionnels

- Un **GDE** est un groupe de données existant en dehors du domaine d'étude que l'utilisateur perçoit comme logiquement liées. Le domaine d'étude ne fait que l'interroger
- Il est créé et mis à jours par un autre domaine
- La complexité d'un GDE est fonction du nombre de **DE** et du nombre de **SDL**

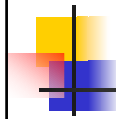
Nombre de points de fonction des GDE :

Complexité	Faible	Moyenne	Elevée
Nombre de points de fonction du GDE	5	7	10



Méthode des points fonctionnels

- Une **ENT** est une fonction élémentaire, significative pour l'utilisateur, qui permet d'introduire des données à l'intérieur du domaine :
 - Des données spécifiques au domaine
 - Des paramètres de traitement
Ex.: mise à jour de l'information sur un *employé* (un écran de saisie)
- Une ENT permet de saisir un certain nombre de champs, qui sont des **données élémentaires (DE)**
Ex.: *nom, prénom, adresse, position, etc. de l'employé*
- Une ENT utilise, en lecture ou mise à jour, différents groupes logiques de données (internes ou externes) appelés des **GDR, groupes de données référencées** (GDI ou GDE)
 - Ex.: classe *Employé*, classe *Personne*



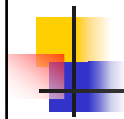
Méthode des points fonctionnels

Complexité des ENT

	1 - 4 DE	5 - 15 DE	16 DE ou plus
0 ou 1 GDR	Faible	Faible	Moyenne
2 GDR	Faible	Moyenne	Elevée
3 GDR ou plus	Moyenne	Elevée	Elevée

Nombre de points de fonction des ENT :

Complexité	Faible	Moyenne	Elevée
Nombre de points de fonction de l'ENT	3	4	6



Méthode des points fonctionnels

- Une **SOR** est une fonction élémentaire, significative pour l'utilisateur, qui envoie des données vers l'extérieur du domaine et qui n'effectue aucune mise à jour à l'intérieur du domaine
- Les données sont les données calculées ou dérivées, obtenus à partir d'autres données
Ex.: calculer le temps du travail de chaque *employé*, calculer l'âge de chaque *employé*
- Une SOR a un certain nombre de **données élémentaires (DE)**
 - Ex.: *nom, prénom, date de naissance* de l'employé, *temps du travail par fonction*, etc.
- Une SOR utilise, en lecture, différents GDR, **groupes de données référencées (GDI ou GDE)**
 - Ex.: classes *Employé, Département, Fonction*



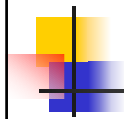
Méthode des points fonctionnels

Complexité des SOR

	1 - 5 DE	5 - 19 DE	20 DE ou plus
0 ou 1 GDR	Faible	Faible	Moyenne
2 - 3 GDR	Faible	Moyenne	Elevée
4 GDR ou plus	Moyenne	Elevée	Elevée

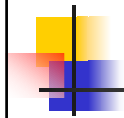
Nombre de points de fonction des SOR :

Complexité	Faible	Moyenne	Elevée
Nombre de points de fonction de SOR	4	5	7



Méthode des points fonctionnels

- Une **INT** est une fonction élémentaire, qui a pour résultat l'extraction de données qui ne sont ni calculées ni dérivées
Ex.: afficher l'information sur un *employé* (un écran d'affichage)
- Sur le résultat d'une INT figure un certain nombre de champs, qui sont des **données élémentaires** (DE)
Ex.: *nom, prénom, adresse, position, etc. de l'employé*
- Une ENT utilise, en lecture, différents GDR, **groupes de données référencées** (GDI ou GDE)
 - Ex.: classes *Employé, Département*



Méthode des points fonctionnels

Complexité des INT

	1 - 5 DE	5 - 19 DE	20 DE ou plus
0 ou 1 GDR	Faible	Faible	Moyenne
2 - 3 GDR	Faible	Moyenne	Elevée
4 GDR ou plus	Moyenne	Elevée	Elevée

Nombre de points de fonction des INT :

Complexité	Faible	Moyenne	Elevée
Nombre de points de fonction de l'INT	3	4	6



Méthode des points fonctionnels

Calcul de nombre de points de fonction brut PFB

Entité	Complexité	Nombre de composants	Poids	Nombre de PFB
GDI	Faible	3	7	21
	Moyenne	1	10	10
	Elevée	1	15	15
GDE	Faible	2	5	10
	Moyenne	2	7	14
	Elevée	3	10	30
ENT	Faible	4	3	12
	Moyenne	6	4	24
	Elevée	2	6	12
SOR	Faible	3	4	12
	Moyenne	4	5	20
	Elevée	0	7	0
INT	Faible	2	3	6
	Moyenne	5	4	20
	Elevée	4	6	24
PFB				230

©J. Ralyté, Université de Genève

41



Méthode des points fonctionnels

2. Ajuster la taille du projet – corriger le nombre de points de fonction brut PFB en fonction des spécificités du projet

- 14 **CGS** - caractéristiques générales du système
- **DI** – degré d’influence de chaque caractéristique (de 0 à 5)
- **DIT** – degré d’influence total

$$\text{DIT} = \text{Somme (DI}_i\text{)} \text{ avec } i = 1 \text{ à } 14$$

- **FA** – facteur d’ajustement

$$\text{FA} = 0,65 * \text{DIT}/100$$

- **PFA** – nombre de points de fonction ajusté

$$\text{PFA} = \text{FA} * \text{PFB}$$



Méthode des points fonctionnels

Caractéristiques :

1. Communication des données
2. Système distribué
3. Performance
4. Intensité d'utilisation de la configuration matérielle
5. Taux de transaction
6. Saisie interactive
7. Convivialité
8. Mise à jour en temps réel des GDI
9. Complexité des traitements
10. Réutilisation du code de l'application
11. Facilité d'installation
12. Facilité d'exploitation
13. Portabilité de l'application
14. Facilité d'adaptation



Méthode des points fonctionnels

3. Transformer le nombre de points de fonction en charge

- Le coefficient de transformation est variable selon l'environnement matériel et humain
- Chaque entreprise devrait établir sa base de projets pour déterminer ses propres coefficients
- Estimations générales :
 - En fin d'étude préalable :
 - 3 jours par point de fonction
 - 2 jours s'il s'agit d'un petit projet
 - 4 jours s'il s'agit d'un grand projet
 - En fin d'étude détaillée :
 - 1 à 2 jours par point de fonction selon l'environnement
 - Ex. un projet de 80 points de fonction correspond à 160 jours/homme