

Estimation des projets informatiques

Estimation des charges : méthodes COCOMO et Points de fonctions

Tatiana AUBONNET
Conservatoire National des Arts et Métiers

Plan

- ◆ Estimation des charges
 - Charge
 - Durée
- ◆ Les besoins d'estimation
- ◆ Les méthodes d'estimation
 - COCO MO
 - Points de Fonctions
- ◆ Autres approches
 - méthode Delphi,
 - évaluation analytique

ESTIMATION DES CHARGES (1)

Notions de base : **Charge et durée**

- ◆ **La CHARGE** représente une quantité de travail nécessaire, indépendamment du nombre de personnes.
 - s'exprime en
 - jour/homme
 - année/homme
 - mois/homme
 - permet d'obtenir un coût prévisionnel,
 - Exemple : 60 mois/homme représente l'équivalent du travail d'une personne pendant 60 mois. Si on évalue le coût complet du mois/homme à 50 000 euros, le projet sera estimé à 3 000 000 euros.
 - aide à définir la taille d'un projet
 - Projet < 6 m/H => très petit
 - Entre 12-30 m/H => projet moyen
 - Projet > 100 m/H => très grand (année/homme).

ESTIMATION DES CHARGES (2)

Notions de base : **Charge et durée**

- ◆ **La DURÉE** est le temps consommé par le projet.
 - ▶ Elle dépend du nombre de personnes, mais l'évaluation n'est pas isotrope
 - ▶ 100 personnes pendant un mois ne sont pas équivalentes à 1 personne pendant 100 mois

Les besoins en estimation (1)

◆ Au niveau du projet global

- Déterminer une enveloppe budgétaire
- Faire une estimation de la rentabilité de l'investissement
- Évaluer une durée

◆ Au niveau de l'étape

- Ordre de grandeur : mois/homme ou semaine/homme
- Ajuster le découpage
- Sous-traiter
- Prévoir des délais pour planifier l'ordonnancement des étapes

Les besoins en estimation (2)

- ◆ Au niveau de la phase (on peut estimer une ou plusieurs phases d'une étape)
 - Faire une planification précise
 - Annoncer un calendrier de remise des différents résultats intermédiaires
 - Prévoir et effectuer un suivi, pour surveiller les écarts
 - Prévoir l'affectation des ressources

Les besoins en estimation (3)

- ◆ Au niveau de la tâche
 - Affectation des ressources individuelles
 - Planification au niveau le plus fin
- ◆ Visibilité croissante du projet vers la tâche
- ◆ Utilisation de techniques différentes selon le niveau de granularité

LES MÉTHODES D 'ESTIMATION

- ◆ **Loi de Parkinson** : « le travail se dilate jusqu 'à remplir le temps disponible »
- ◆ « **Méthode du marché** » : la charge correspond au prix à proposer pour remporter l 'appel d 'offre.
- ◆ **Méthodes** :
 - COCO MO
 - Points de fonctions
 - Delphi,
 - évaluation analytique
 - ...

LES MÉTHODES D 'ESTIMATION

◆ Schéma général

- Construire une BC (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
- Ajuster la taille ou la charge brute en fonction des spécificités du projet
- Répartir la charge entre les différentes étapes.

LA MÉTHODE COCOMO (1)

- ◆ COnstructive COst Model (COCOMO), modèle des construction des couts, Boehm 1981
- ◆ Deux hypothèses :
 - Un informaticien évalue mieux la taille du logiciel à développer que la quantité de travail nécessaire
 - Il faut toujours le même effort pour écrire un nombre donné de lignes de programme, quel que soit le langage (3eme génération)

LA MÉTHODE COCOMO (2)

- ◆ L 'unité : l 'instruction source
- ◆ Le modèle permet d 'obtenir la charge de réalisation en m/H et le délai normal recommandé
- ◆ Formules de calcul :
 - **Charge en mois/Homme = a (Kisl)^b**
 - ▶ Kisl = kilo instruction source livrée (lignes de programme source testées)
 - **Durée normale en mois = c(charge en mois/Homme)^d**

LA MÉTHODE COCOMO (3)

- ◆ Les paramètres **a**, **b**, **c** et **d** dépendent de la catégorie du projet. Soit **L** la taille du logiciel.
 - Projet simple si $L < 50$ Kisl, spécifications stables, petite équipe.
 - Projet moyen logiciel comporte entre 50 Kisl et 300 Kisl (spécifications stables, petite équipe).
 - Projet complexe si $L > 300$ Kisl, grande équipe.

LA MÉTHODE COCOMO (4) : les valeurs des paramètres

Type de projet	Charge en mois homme	Durée en mois
Simple	$C = 3,2 (Kisl)^{1,05}$	$D = 2,5 (C)^{0,38}$
Moyen	$C = 3 (Kisl)^{1,12}$	$D = 2,5 (C)^{0,35}$
Complexe	$C = 2,8 (Kisl)^{1,2}$	$D = 2,5 (C)^{0,32}$

LA MÉTHODE COCOMO (5)

- ❖ Il faut tenir compte des « facteurs correcteurs » d'estimation de charge.
- ❖ Quatre sources de risque sur l'estimation
 - Exigences attendues du logiciel
 - Caractéristiques de l'environnement technique (matériel)
 - Caractéristiques de l'équipe projet
 - Environnement du projet lui-même

LA MÉTHODE COCOMO (6)

- ◆ Les facteurs logiciels sont :
 - Fiabilité du logiciel : influence forte si exigence dans ce sens
 - Base de données : mesuré par le ratio
 - (volume de données gérées en octets) / (taille du logiciel en lignes)
 - L'influence du facteur est faible si le ratio < 10, très forte si ratio > 1000
 - Complexité : celle des algorithmes
 - Temps d'exécution : crucial si temps réel

LA MÉTHODE COCOMO (7)

- ◆ Les facteurs matériels sont :
 - Taille mémoire : s'il est nécessaire de l'optimiser
 - Stabilité de l'environnement : celle du logiciel de base
 - Contrainte de délai : se mesure par rapport au délai calculé « normal ».

LA MÉTHODE COCOMO (8)

- ◆ La correction intervient dans la formule :
 - **Charge nette = produit (valeurs des facteurs correcteurs) * Charge brute**
- ◆ Démarche en cinq étapes:
 - Estimation du nombre d'instructions source.
 - Calcul de la charge « brute ».
 - Sélection des facteurs correcteurs
 - Calcul de la charge nette
 - Évaluation de la durée sur la charge nette.

COCOMO et le cycle de développement

- ◆ COCOMO divise en 4 grandes phases le cycle de développement :
 - Expression des besoins et planification
 - Conception général
 - Programmation
 - ▶ Conception détaillée
 - ▶ Programmation et tests
 - Tests et intégration
- ◆ Selon la complexité et la taille (en Kisl) de l'application, la charge et le temps de développement varient.
- ◆ Le modèle COCOMO exprime cela sous la forme d'un coefficient représentant le pourcentage d'effort à réaliser et le temps passé.

Coefficients de charge

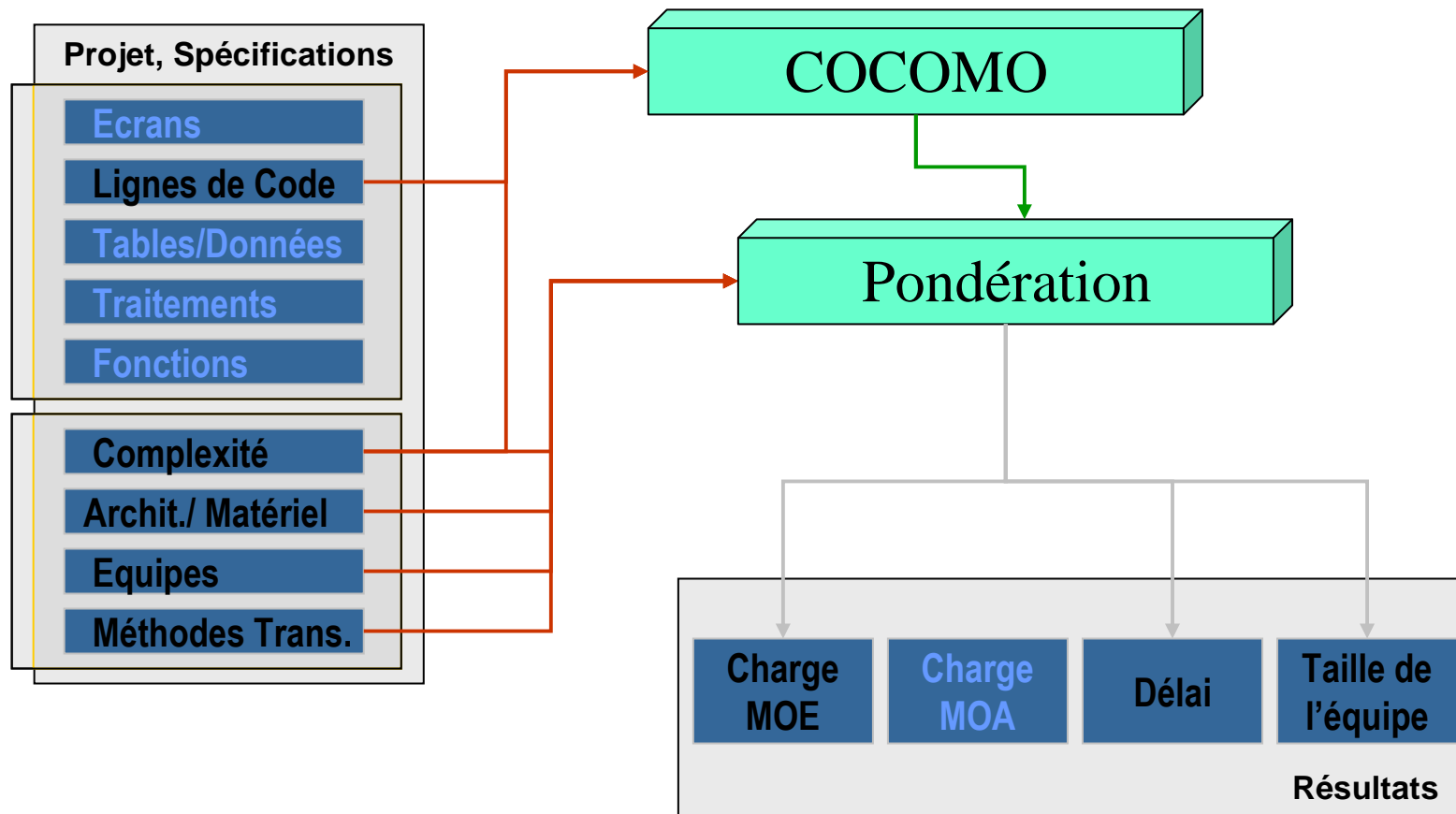
Distribution de charge (par phase) :

Complexité	Phase	Taille de 2 Kisl	Taille de 8 Kisl	Taille de 32 Kisl	Taille de 128 Kisl	Taille de 512 Kisl
S	Expression des besoins et planification	6	6	6	6	
	Conception général	16	16	16	16	
	Programmation	68	65	62	59	
	Conception détaillée	25	25	24	23	
	Programmation et tests unitaires	42	40	38	36	
	Tests et intégration	16	19	22	25	
M	Expression des besoins et planification	7	7	7	7	7
	Conception général	17	17	17	17	17
	Programmation	64	61	58	55	52
	Conception détaillée	27	26	25	24	23
	Programmation et tests unitaires	37	35	33	31	29
	Tests et intégration	19	22	25	28	31
C	Expression des besoins et planification	8	8	8	8	8
	Conception général	18	18	18	18	18
	Programmation	60	57	54	51	48
	Conception détaillée	28	27	26	25	24
	Programmation et tests unitaires	32	30	28	26	24
	Tests et intégration	22	25	28	31	34

Coefficients du temps de développement

Distribution du temps de développement (par phase) :						
Complexité	Phase	Taille de 2 Kisl	Taille de 8 Kisl	Taille de 32 Kisl	Taille de 128 Kisl	Taille de 512 Kisl
S	<i>Expression des besoins et planification</i>	10	11	12	13	
	<i>Conception général</i>	19	19	19	19	
	<i>Programmation</i>	63	59	55	51	
	<i>Tests et intégration</i>	18	22	26	30	
M	<i>Expression des besoins et planification</i>	16	18	20	22	24
	<i>Conception général</i>	24	25	26	27	28
	<i>Programmation</i>	56	52	48	44	40
	<i>Tests et intégration</i>	20	23	26	29	23
C	<i>Expression des besoins et planification</i>	24	28	32	26	40
	<i>Conception général</i>	30	32	34	36	38
	<i>Programmation</i>	48	44	40	36	32
	<i>Tests et intégration</i>	22	24	25	28	30

Résumé de la Méthode



LA MÉTHODE COCOMO (4) : les valeurs des paramètres

Type de projet	Charge en mois homme	Durée en mois
Simple	$C = 3,2 (Kisl)^{1,05}$	$D = 2,5 (C)^{0,38}$
Moyen	$C = 3 (Kisl)^{1,12}$	$D = 2,5 (C)^{0,35}$
Complexe	$C = 2,8 (Kisl)^{1,2}$	$D = 2,5 (C)^{0,32}$

La méthode COCOMO / étude de cas

- ❖ Estimer un projet visant à développer un logiciel de 40 000 instructions source (sans tenir compte des « facteurs correcteurs ») ?

La méthode COCOMO / étude de cas - solution

- ❖ Estimer un projet visant à développer un logiciel de 40 000 instructions source (sans tenir compte des « facteurs correcteurs ») ?
 - Charge = $3,2 (40)^{1,05} = 154$ mois/homme
 - Durée normale = $2,5 (154)^{0,38} = 17$ mois
 - Une taille moyenne de l'équipe = $154 / 17 = 9$ personnes.

LA MÉTHODE POINTS DE FONCTIONS (1)

- ◆ Méthode d'Albrecht (IBM) 1979
- ◆ Groupe d'utilisateurs : en 1984
- ◆ En France : en 1992
- ◆ Principe :
 - Estimation à partir d'une description externe du futur système, et de ses fonctions.
 - 5 types d'unité d'œuvre (composants fonctionnels)
 - 3 degrés de complexité

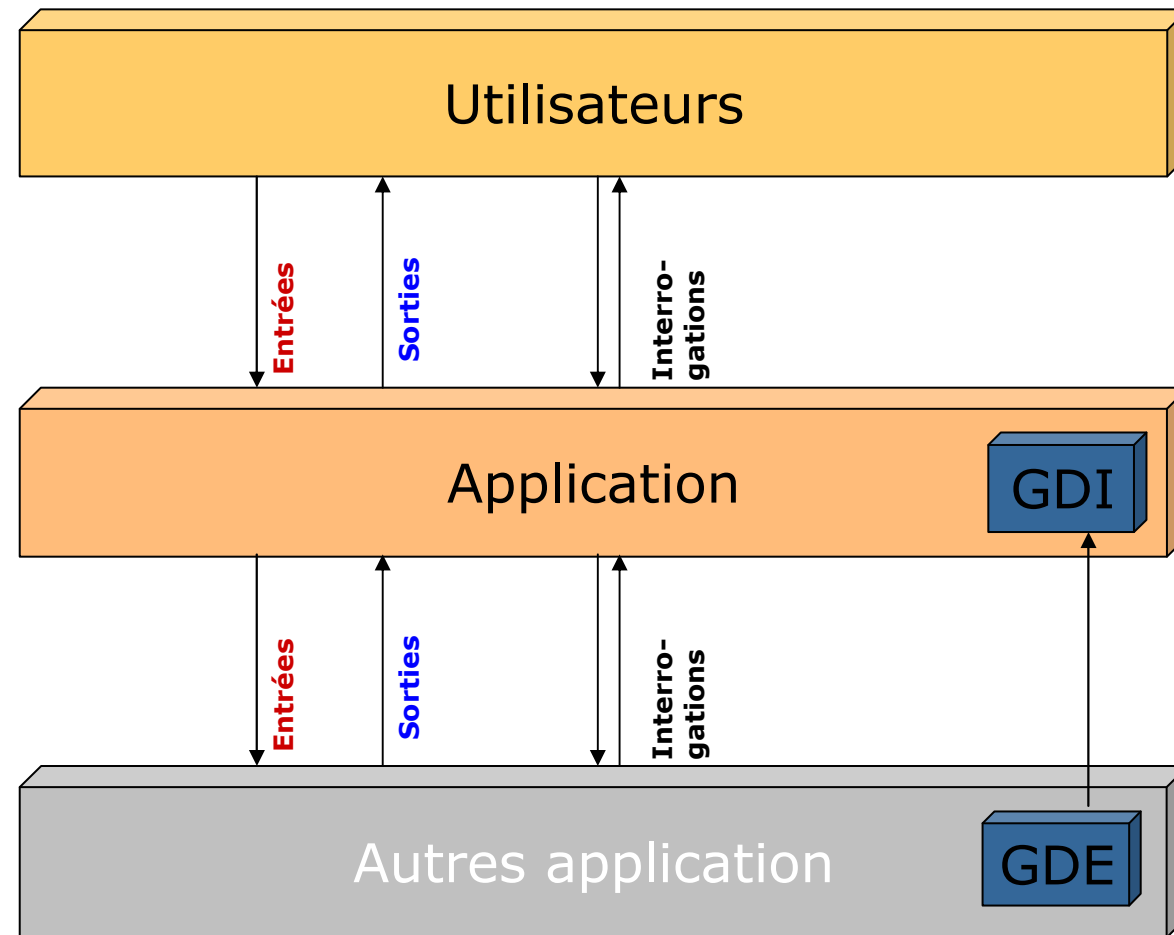
LA MÉTHODE POINTS DE FONCTIONS (2)

- ❖ Pour un projet donné on calcule son poids en « points de fonctions ».
- ❖ Méthode:
 - Comptage des points au début du projet
 - Comptage en fin
 - Écart = changement d'envergure
 - Évaluation :
 - ▶ Calcul de la taille, ajustement de la taille, transformation en charge.

LA MÉTHODE POINTS DE FONCTIONS : calcul de la taille

- ◆ Composants fonctionnels :
 - Groupe logique de données internes (GDI)
 - Groupe logique de données externes (GDE)
 - Entrée de traitement (ENT)
 - Sortie de traitement (SORT)
 - Interrogation (INT)

Points de Fonctions



Les sorties

- ◆ Les sorties (SOR) = fonctions qui extraient des données de l'application en sollicitant un traitement avant restitution
 - Exemples : impressions, rapports d'anomalie, restitution d'informations, informations destinées à une autre application

Les interrogations

- ◆ Les interrogations (INT) = combinaisons d'E/S qui :
 - ne font pas de mise à jour de GDI
 - Résultent d'une extraction traitement de données, sans traitement;
- ◆ Exemple : visualisation des fichiers

LA MÉTHODE POINTS DE FONCTIONS : calcul de la taille

- ◆ Complexité d'un composant :
 - Faible
 - Moyenne
 - Élevée
- ◆ Nombre de points de fonction du composant :
 - Tableau de correspondance entre la complexité et le type du composant = > poids

Calcul du nombre de points de fonction brut : exemple

Entité	Complexité	Nb de composants	Poids	Nb de Points de fonction
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
SORT	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

LA MÉTHODE POINTS DE FONCTIONS : l'ajustement (1)

- ◆ On corrige le nombre de PFB en appréciant les spécificités du projet pouvant influencer l'effort :
 - 14 points appelées CGS (Caractéristiques Générales du Système sont identifiées) sont identifiées :
 1. Communication des données
 2. Systèmes distribués
 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'utilisation
 12. Facilité d'exploitation
 13. Portabilité de l'application
 14. Facilité d'adaptation

LA MÉTHODE POINTS DE FONCTIONS : l'ajustement (2)

- ◆ A chaque CGS est attribuée une note de 0 à 5 en fonction du degré d'influence (DI).
- ◆ L'analyse des caractéristiques permet de calculer un degré d'influence total, le DIT compris entre 0 et 70.
 - degré d'influence total (DIT)
 - ▶ $DIT = (SOMME (DI_i))_{i = 1 \text{ à } 14}$
- ◆ Le facteur d'ajustement FA permet d'ajuster le PFB de + ou – 35%
 - facteur d'ajustement
 - ▶ $FA = 0,65 + DIT / 100$

LA MÉTHODE POINTS DE FONCTIONS : l'ajustement (3)

◆ Le PFA (Nombre de Ponts de fonctions ajusté) :

■ $PFA = FA * PFB$

ou $PFA = (0,65 + (SOMME (Di_{i, i = 1 \text{ à } 14})/100) * PFB$

LA MÉTHODE POINTS DE FONCTIONS : l'ajustement (4)

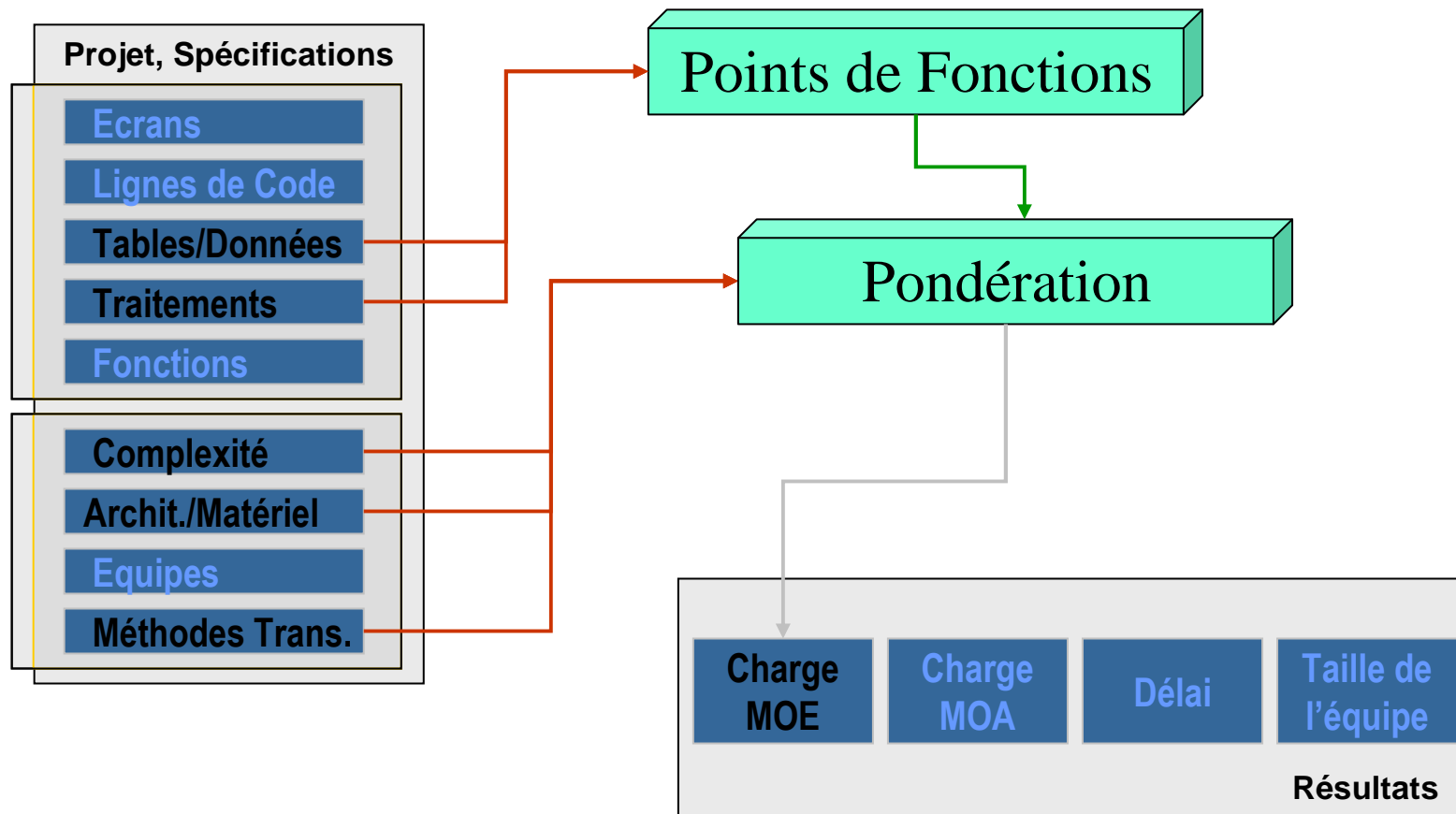
- ◆ Pour établir une correspondance entre la taille "fonctionnelle" et la taille de logiciel
 - Le PF permet de donner le nombre d'instructions source utile pour COCOMO avec la formule :

$$ISL (I_{\text{procédural}}) = 118,7 * PFA - 6490.$$

LA MÉTHODE POINTS DE FONCTIONS

- ◆ On calcule la charge en convertissant directement les points :
 - En fin d'étude préalable
 - ▶ 3 jours par point de fonction (PF)
 - ▶ 2 jours si petit projet
 - ▶ 4 jours si grand projet
 - En fin d'étude détaillée : 1 à 2 j par PF selon l'environnement (grand système, client/serveur)
 - Avec un L4G : 1j pour 10 PF *en réalisation*.
 - En RAD , productivité élevée : 0,5 j/homme par PF

Résumé de la Méthode



LA MÉTHODE de POINTS DE FONCTIONS

◆ Étude de cas

Étude de cas

◆ Description du projet d'attribution des autorisations de parking

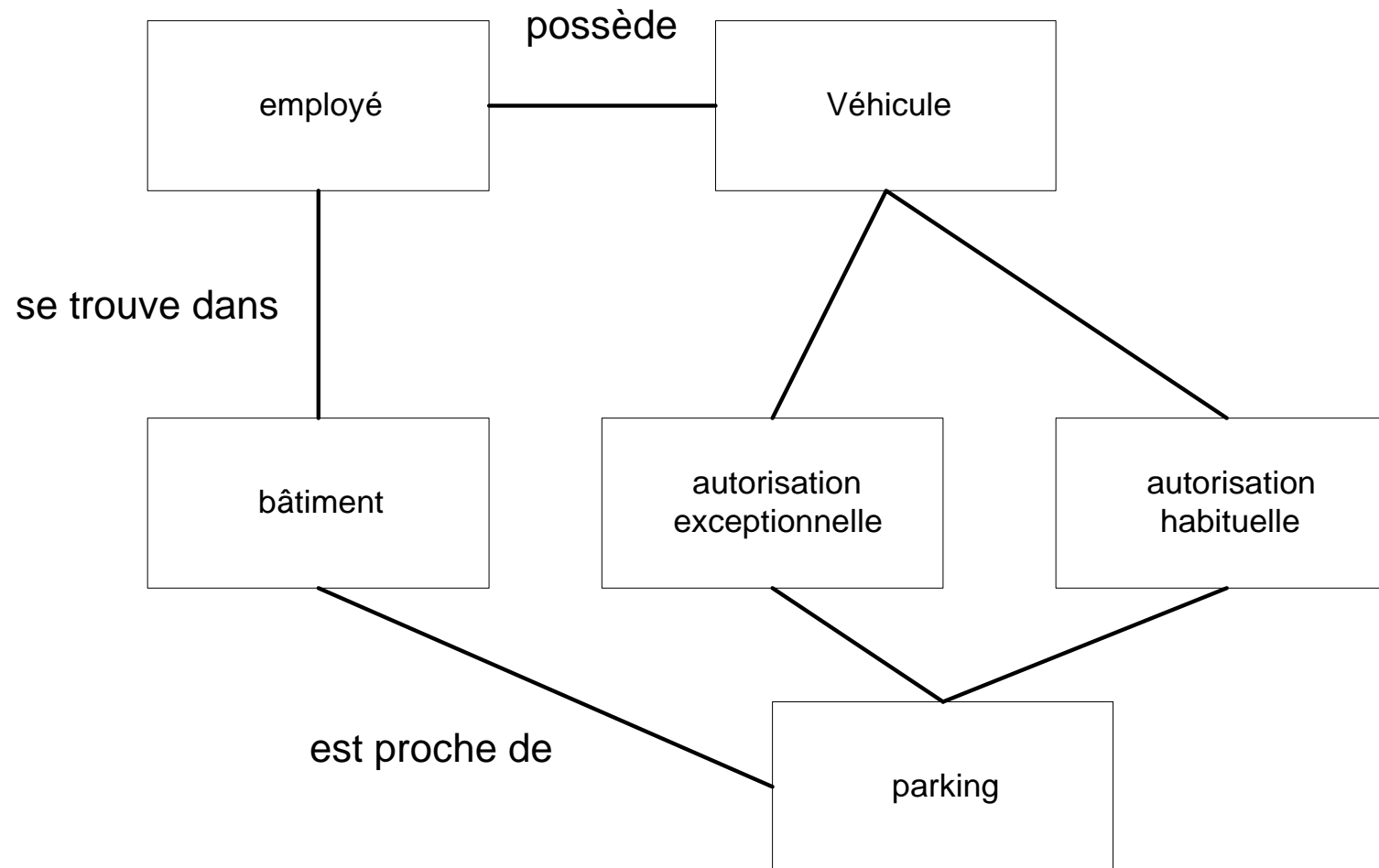
Dans un centre de recherche les employés sont répartis sur des différents bâtiments (site) parfois éloignés les uns des autres. On veut gérer l'accès aux différents parkings.

On définit, pour chaque parking, les bâtiments qui sont accessibles à partir de ce parking. L'attribution des places de parking se fera en fonction du lieu d'affectation de l'employé.

Les employés peuvent obtenir des autorisations exceptionnelles de parking pour la participation à des réunions sur les autres sites.

- ➡ Utilisez la méthode de points de fonctions pour estimer la charge de ce projet en faisant des hypothèses sur les fonctionnalités.

MCD simplifié



Les groupes de données

- ◆ employé
- ◆ véhicule
- ◆ bâtiment
- ◆ parking
- ◆ autorisation exceptionnelle
- ◆ autorisation habituelle

Identifier les groupes de données GDE et GDI

- ◆ **Hypothèse** : employé, bâtiment, **parking** sont déjà gérés par d'autres domaines

GDE	GDI
employé	véhicule
bâtiment	autorisation exceptionnelle
parking	autorisation habituelle

- ◆ Hypothèse : à part l'employé (complexité moyenne) tous les groupes de données, internes ou externe sont de faible complexité.

Type	Complexité	Nombre
GDI	Faible	3
GDE	Faible	2
	Moyenne	1

Dénombrement des entrées (ENT)

- ◆ On s'appuie sur les données internes
 - un écran de saisie véhicule
 - un écran d'affectation d'une autorisation habituelle
 - un écran de saisie de demande exceptionnelle
 - un écran de saisie d'affectation d'une autorisation exceptionnelle

- ▶ Nombre de composants = 4
- ▶ Complexité moyenne

Dénombrement des sorties (SOR)

il s'agit de statistiques sur l'occupation des parkings,
les autorisations exceptionnelles ...

Type	Complexité	Nombre
SOR	Faible	3
	Moyenne	2

Dénombrement des interrogations

- ◆ Chaque groupe des données interne doit pouvoir être consulté :
 - véhicule
 - autorisation exceptionnelle
 - autorisation habituelle
- ◆ Plus les listes croisées:
 - employé ayant reçu des autorisations exceptionnelles
 - véhicule affecté à un parking
 - ...

Type	Complexité	
INT	Faible	3
	Moyenne	2

Estimation de la charge

A calculer

Entité	Complexité	Nb de composants	Poids	Nb de Points de fonction
GDI	Faible	4	7	28
	Moyenne		10	
	Elevée		15	
GDE	Faible	2	5	10
	Moyenne	1	7	7
	Elevée		10	
ENT	Faible	5	3	20
	Moyenne		4	
	Elevée		6	
SORT	Faible	3	4	12
	Moyenne	2	5	10
	Elevée		7	
INT	Faible	3	3	24
	Moyenne	2	4	8
	Elevée		6	
PFB				119

Charge du projet

- ◆ La taille du logiciel est de 119 points de fonction.
- ◆ En prenant une valeur moyenne de 2 jours par pF

■ Charge du projet = 238 jours/hommes
soit 11, 9 mois /hommes

- ▶ En fin d'étude préalable
 - 3 j/H /pF
 - 2 jours si petit projet
 - 4 jours si grand projet
- ▶ En fin d'étude détaillée : 1 à 2 j / pf selon l'environnement
- ▶ Avec un L4G 1j /10 pf en réalisation.
- ▶ En RAD , productivité élevée : 0,5 j/H/pF

La méthode DELPHI

- ◆ Élaborée en 1948 par la Rand Corporation
- ◆ Fondée sur le jugement d 'experts
- ◆ Consiste à rechercher des analogies avec des projets antérieurs.
- ◆ Repose sur un raffinement successif de jugements porté par plusieurs experts jusqu 'à obtention d 'une convergence.

LA MÉTHODE ANALYTIQUE (1)

- ◆ S'appuie sur la typologie des programmes à développer
- ◆ Affecte un poids par type de programme et niveau de difficulté dans l'environnement
 - UNITÉ : jour/homme
- ◆ La charge obtenue est celle de réalisation
- ◆ Pour les test d'enchaînement : 10% charge
- ◆ Pour l'encadrement : 20% charge

LA MÉTHODE ANALYTIQUE (2)

TYPE DE PROGRAMME	FACILE	MOYEN	DIFFICILE
MENU	0.25	0,5	1
CONSULTATION	1	2,5	4
MISE A JOUR	1,5	3	5
EDITION EN TEMPS RÉEL	1	2	4
EXTRACTION	0,5	1	1,5
MISE A JOUR PAR LOT	2	3	5
EDITION PAR LOT	1,5	2,5	4

LA MÉTHODE ANALYTIQUE (3)

- ◆ Charge de réalisation = somme ($p_i * t_i$)
 - Où p est le poids
 - t nombre de programmes du type i
- ◆ Charge globale = $1,3 * Cr / 22$ (en m/H)
- ◆ Pour les projets dont la charge est comprise entre 3 et 30
 - Durée incompressible = $2,5 (Cg \text{ en m/H})^{1/3}$ en mois