

**Intitulé de la matière :**  
**Conduite de projet et entreprenariat**  
**Crédits : 2**  
**Coefficients : 1**

Enseignante :  
M<sup>me</sup> Z.Boukherrou

- **Objectifs de l'enseignement**
  - Les principaux objectifs sont d'acquérir une vision globale des projets, d'apporter des outils pratiques d'anticipation des dérives coûts, délais, performances sur les projets industriels,
- **Connaissances préalables recommandées**
  - Notion du système d'information

# Contenu de la matière

- **Chapitre 1 :** Introduction au Management de Projet
  - Définition de Projet et du Management
  - Modèles de Cycle de Vie d'un Projet
  - Management d'un Projet Système d'Information (SI)
  - Estimation des Charges : Cas du Modèle COCOMO
- **Chapitre 2 :** Techniques de Planification de Délais
  - Diagramme de Gantt
  - Réseau PERT
- **Chapitre 3 :** Maîtrise de la Qualité et Management des Risques
- **Chapitre 4 :** Lancer et Gérer une Statup
  - Composition Humaine et Modèle économique
  - Incubateurs/Accélérateurs/Pépinières/FabLAB
  - Espaces Coworking
  - Mentorat/Financement/Business Angels
  - Capital Risque
- **Chapitre 5 :** systèmes et réseaux informatiques dans l'entreprise

## Définition de Projet et du Management de Projet

### 1. Définition d'un Projet

- Un projet est une entreprise temporaire ayant pour objectif de créer un produit, service ou résultat unique. Cela signifie qu'un projet a un début et une fin définis, ainsi qu'un objectif précis.
- **Exemple :** La construction d'un pont, le lancement d'une application mobile, ou l'organisation d'un événement.

## 2. Caractéristiques d'un Projet

- **Unique** : Chaque projet est différent des autres en termes d'objectif, de contexte, ou de complexité.
- **Temporaire** : Un projet a un début et une fin, et n'est pas censé durer indéfiniment.
- **Complexité et Risque** : Les projets incluent souvent des éléments d'incertitude.
- **Exemple** : Comparons un projet de développement d'une application avec la production de biens en série (ex : des voitures) ; l'application est unique et ne se répète pas de la même façon.

## 3. Définition du Management de Projet

- Le management de projet est l'application de connaissances, de compétences, d'outils et de techniques pour répondre aux exigences d'un projet.
- **Exemple** : Le gestionnaire de projet est comme un chef d'orchestre qui coordonne les membres de l'équipe pour atteindre l'objectif du projet.

## Modèles de Cycle de Vie d'un Projet

### 1. Cycle de Vie du Projet

- Le cycle de vie d'un projet comprend les phases par lesquelles le projet passe de son initiation à sa clôture.
- Les phases courantes incluent : **Initiation, Planification, Exécution, Contrôle, et Clôture.**

## Définition du Petit Robert

- Le projet est un ensemble d'actions à réaliser pour satisfaire un Objectif défini, dans le cadre d'une mission précise, et pour la réalisation desquelles on a identifié non seulement un début, mais aussi une fin.

## Selon la norme ISO 10006 (version 2003)

- « un projet est un processus unique qui consiste en un **ensemble d'activités coordonnées** et maîtrisées, comportant des dates de début et de fin, entrepris dans le but d'atteindre un **objectif** conforme à des exigences spécifiques, incluant des **contraintes** de *délais*, de *coûts* et de *ressources* ».
- La définition d'un projet comporte deux notions clés :
  - le projet est unique* et
  - le projet est temporaire*.

## Le « Triangle Projet »

- Un projet comporte 3 niveaux d'objectifs:

- Les objectifs de **qualité**
- Les objectifs de **temps**
- Les objectifs de **coût**.



- Un objectif est toujours **SMART**.



## Concepts clés

- **Le maître d'ouvrage :** Selon l'AFNOR, le maître d'ouvrage est « la personne physique ou, le plus souvent, personne morale qui sera le propriétaire de l'ouvrage. Il fixe les objectifs, l'enveloppe budgétaire et les délais souhaités pour le projet ».
- **Le maître d'œuvre :** Le maître d'œuvre est « la personne physique ou morale qui réalise l'ouvrage pour le compte du maître d'ouvrage et qui assure la responsabilité globale de la qualité technique, du délai et du coût ».

- 
- **Le Projet** : ensemble d'activités regroupant les trois caractéristiques suivantes :  
Il est entrepris pour satisfaire un besoin spécifique, sa durée et les moyens accordés sont limités et enfin il aboutit à un résultat unique censé satisfaire le besoin.
  - **Un système** est un ensemble d'*éléments* interagissant entre eux suivant un certains nombres de principes et de *règles* dans le but de réaliser un *objectif*.
  - **Un modèle** : est une représentation schématique de la réalité.

- 
- **Une analyse** : c'est un processus d'examen de l'existant
  - **Une Conception** : est un processus de définition de la future application informatique.
  - **Un système d'Information** : ensemble des moyens (*humains et matériels*) et des méthodes se rapportant au traitement de l'information d'une organisation.
  - **Une base de Données**: ensemble des données (*de l'organisation*) structurées et liées entre elles : stocké sur support à accès direct (*disque magnétique*) ; géré par un SGBD (*Système de Gestion de Bases de Données*), et accessible par un ensemble d'applications.

- 
- **L'équipe du projet :** D'après, l'AFNOR, c'est « l'ensemble des personnes placées sous l'autorité directe (et quelquefois indirecte) du chef de projet ». Mais, on peut parfois considérer que l'équipe de projet s'étend à toutes les personnes participant à la réalisation du projet. On distingue ainsi trois différents types d'acteurs dans l'équipe de projet :

- 
- Le **chef de projet**, chef de projet est défini par l'AFNOR comme : « la personne physique chargée par le maître d'œuvre, dans le cadre d'une mission définie, d'assurer la maîtrise du projet, c'est-à-dire de veiller à sa bonne réalisation dans les objectifs de technique, de coût et de délai ».
  - Le **concepteur**, qui peut être tenu par un informaticien, et qui joue le rôle d'un organisateur ou un gestionnaire selon le stade d'avancement : sa responsabilité est de concevoir le futur système aux étapes étude préalable et étude détaillée.

- **Le développeur**, qui est tenu par un informaticien : sa responsabilité est d'écrire les programmes ou de réaliser un prototype. Pour certains développements réalisés en langage de 4e génération, le rôle peut être tenu par un gestionnaire.
- **Les utilisateurs** : Les utilisateurs ont pour rôle la mise en correspondance avec les fonctions permanentes du développement du produit réalisé. Ces derniers peuvent représenter différents niveaux : décideurs, gestionnaires ou utilisateurs finals.

## Découpage de projets informatiques

- Il est recommandé de découper le processus projet en « **phases** » ordonnées dans le temps, selon la logique de construction du produit. L'ensemble des phases du projet est appelé « **cycles de vie** » du projet. Le cycle de vie est unique pour chaque projet, mais il existe selon les domaines d'application des cycles de vie génériques qui guident le découpage du projet en phases. En informatique, on les appelle souvent des **modèles de développement**.

- L'AFNOR a proposé un découpage type classique des projets informatiques (*NF Z67-101, 1984*) en **cinq phases** : *Étude préalable, Conception détaillée, Réalisation, Mise en œuvre et Évaluation*.



## Principaux modèles et méthodes de développement de projet

Modèle/Méthode	Principe	Avantages	Inconvénients	Exemples d'Application
Cascade	Modèle séquentiel où chaque phase doit être terminée avant de passer à la suivante (analyse, conception, implémentation, test, maintenance).	Simple, structuré, facile à comprendre	Manque de flexibilité, difficile de revenir en arrière	Projets simples, peu de changements attendus

Cycle en V				
	Similaire au modèle en cascade, avec une phase de test pour chaque phase de développement, garantissant une vérification constante.	Tests intégrés, réduit les erreurs en fin de projet	Rigide, difficile d'adapter les exigences	Développement de systèmes critiques (aérospatial, santé)

Prototypage				
	Création d'un prototype fonctionnel pour recueillir les retours des utilisateurs avant le développement final.	Améliore la compréhension des besoins, flexible	Peut augmenter les coûts, risque de prolongation	Conception de systèmes interactifs

Modèle en Spirale	Combinaison de la gestion de risques et de développement incrémental, avec des cycles répétitifs pour évaluer les risques et ajuster les objectifs.	Réduit les risques, flexible, idéal pour projets complexes	Coûteux, nécessite une expertise en gestion de risques	Projets à haut risque ou très complexes
-------------------	---	--	--	---

RAD (Rapid Application Development)	Accélère le développement via des prototypes, impliquant des utilisateurs finaux tout au long du processus pour des ajustements rapides.	Rapide, proche des attentes des utilisateurs	Risque de qualité faible, coûteux en ressources	Développement de logiciels d'entreprise
-------------------------------------	--	--	---	---

Méthode Agile	Développement itératif et incrémental avec des sprints courts (Scrum, Kanban) pour livrer des parties fonctionnelles du projet.	Adaptabilité, livraison continue de valeur	Complexe à gérer pour grands projets, demande une forte implication des parties prenantes	Développement logiciel nécessitant des mises à jour fréquentes
---------------	---	--	---	--

## Modèles de Cycle de Vie d'un Projet

Ces modèles et méthodes peuvent être adaptés en fonction des contraintes spécifiques du projet (taille, complexité, niveau d'incertitude), du secteur d'activité, et des objectifs du client. Pour les projets nécessitant une adaptation rapide aux changements, les méthodes agiles telles que **Scrum** ou **Kanban** (composants d'Agile) sont souvent privilégiées, tandis que des modèles plus linéaires comme **Cascade** ou **Cycle en V** conviennent mieux aux projets aux exigences bien définies dès le départ.

# Management d'un projet Système d'Information (SI)

## 1. Particularités des Projets SI

- Les projets de systèmes d'information sont complexes, incluent des défis techniques, des contraintes de temps, et une intégration de multiples parties prenantes.
- **Exemple :** La mise en place d'un ERP (Enterprise Resource Planning) dans une grande entreprise implique différents départements (finances, RH, logistique) et nécessite une coordination avancée.

## 2. Outils et Techniques pour les Projets SI

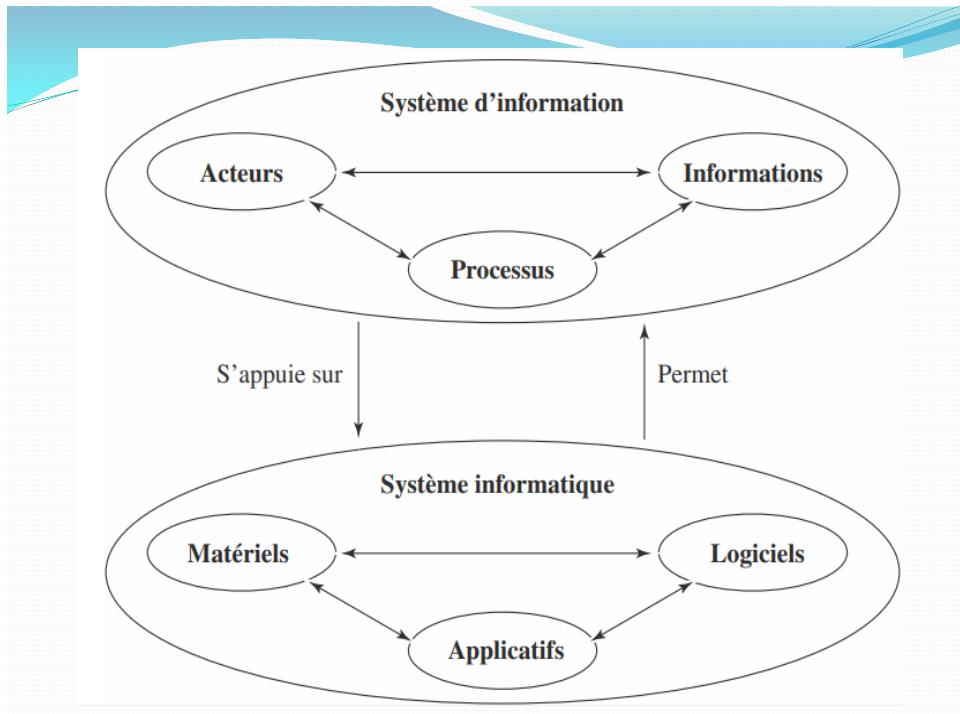
- **Gestion des Exigences** : Collecte et analyse des besoins des parties prenantes.
- **Analyse de Systèmes** : Identification des besoins du système pour assurer qu'il répondra aux attentes.
- **Exemple** : Pour un projet de développement de logiciel CRM (Customer Relationship Management), les exigences comprennent l'intégration des données clients, le suivi des ventes, etc.

### 3. Challenges dans les Projets SI

- **Complexité Technique** : Problèmes d'intégration avec les anciens systèmes.
- **Dépendances Multiples** : Les projets SI dépendent souvent d'autres systèmes ou services.
- **Exemple** : Lors de la mise en place d'un système de gestion hospitalier, il faut prendre en compte les anciennes bases de données, les systèmes d'ordonnance, et la compatibilité avec le matériel.

## Système d'information et système informatique.

- Le **système d'information** est la partie du réel constituée d'informations organisées, d'événements ayant un effet sur ces informations et d'acteurs qui agissent sur ces informations ou à partir de ces informations, selon des processus visant une finalité de gestion et utilisant les technologies de l'information.
- Un **système informatique** est un ensemble organisé d'objets techniques — matériels, logiciels, applications — dont la mise en œuvre réalise l'infrastructure d'un système d'information.



## Le management des projets système d'information

- le **management de projet** comprend « *les tâches de direction, gestion, maîtrise, pilotage* ». Ces tâches peuvent être assurées « *par une même personne ou plusieurs, appartenant à une même entreprise ou à plusieurs entités, parties prenantes du projet* ».
- Le management de projet est donc de la responsabilité du chef de projet. Le management de projet se compose donc de quatre activités, pouvant correspondre à une fonction :
  - Direction de projet.
  - Gestion de projet.
  - Maîtrise.
  - Pilotage.

## Estimation des charges d'un projet

### 1. Définition et Importance de l'Estimation des Charges

- Estimer les charges signifie déterminer le temps, les efforts et le coût nécessaires pour mener à bien un projet.
- **Importance :** Une estimation réaliste permet d'allouer les ressources et de fixer des objectifs réalistes.

### 2. Le Modèle COCOMO (COnstructive COst MOdel)

- Le modèle COCOMO est une méthode d'estimation des coûts de développement logiciel basée sur la taille du projet, mesurée en lignes de code.

## Productivité des programmeurs

- **Définition :** Mesure de la vitesse à laquelle les ingénieurs produisent des programmes et la documentation associée
- **Objectif :** mesurer le nombre de fonctionnalités utiles produites par unité de temps

## La qualité du logiciel

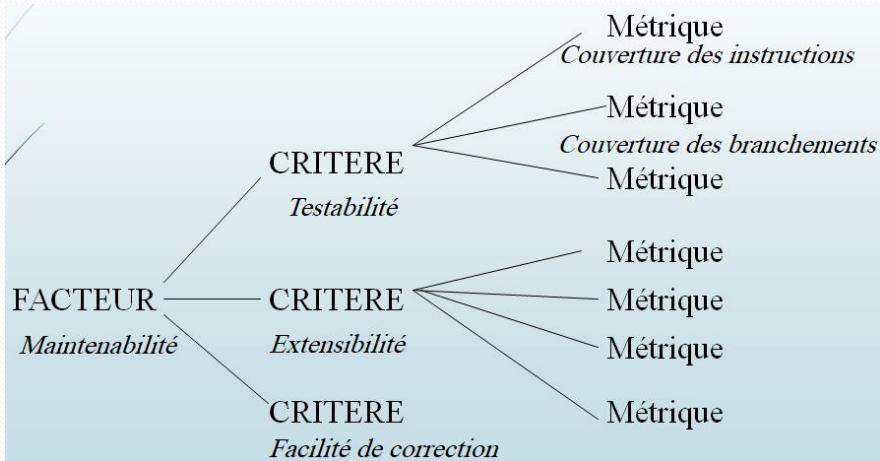
En Génie Logiciel

La qualité d'un logiciel est son aptitude à satisfaire les besoins (exprimés ou potentiels) des utilisateurs [Martin, 1987]

## Modèle de qualité

- **Facteurs de qualité** : Attributs *externes* du logiciel
  - fiabilité, facilité d'utilisation, maintenabilité,...
- **Critères de qualité** : Attributs *internes* du logiciel
  - la testabilité est un des critères de maintenabilité
- **Métriques** : attributs mesurables associés à chaque critère
  - Pour la testabilité : Degré de test mesuré par la couverture des instructions, la couverture des branchements, etc.

## Modèle de qualité (Norme IEEE)



## Définition de la mesure

- Mesurer c'est affecter une valeur à un attribut d'une entité de telle manière que cela représente bien l'entité
- Exemple : On affecte la valeur 10 à l'attribut « temps passé en homme-mois » de l'entité « codage du programme P5 »

## Exemples de métriques

- **Lignes de code** : mesure de la taille d'un logiciel
- **Points fonctions** : mesure des fonctionnalités du logiciel
- **Homme-mois** : mesure de l'effort de développement

## Effort ? Durée ? Taille ?...

- **Effort ou charge** : quantité de travail nécessaire, indépendamment du nombre de personnes qui vont réaliser ce travail
  - Permet d'obtenir un coût prévisionnel
  - S'exprime en homme/jour, homme/mois ou homme/année
  - Un homme/mois (HM) représente l'équivalent du travail d'une personne pendant un mois, généralement 20 jours
  - Un homme/mois (HM)=152 heures de travail par mois

## Effort ? Durée ? Taille ?...

On mesure la taille des projets à leur charge

- Si Charge < 6 HM : **très petit projet**
- Si 6 HM < Charge < 12 HM : **petit projet**
- Si 12 HM < Charge < 30 HM : **projet moyen**
- Si 30 HM < Charge < 100 HM : **grand projet**
- Si Charge > 100 HM : **très grand projet**

## Effort ? Durée ? Taille ?...

La **Durée** dépend du nombre de personnes

- 60 HM peut correspondre à :
  - 1 personne pendant 5 ans ,
  - 5 personnes pendant un an ,
  - 10 personnes pendant 6 mois
  - 60 personnes pendant 1 mois

## L'Homme-Mois

- Unité de mesure de l'effort
  - Un homme pendant un mois
  - Deux hommes-mois = 2 hommes pendant 1 mois, ou 1 homme pendant deux mois.

## Effort ? Durée ? Taille ?... Les lignes de code

- Définition [Fenton 1991]
  - Une ligne de code est toute ligne du texte d'un programme qui n'est pas une ligne de commentaires ou une ligne blanche, sans considération du nombre d'instructions ou de fragments d'instructions dans la ligne. Sont incluses toutes les lignes contenant des en-têtes de programmes, des déclarations et des instructions exécutables et non exécutables.

## Problèmes avec les lignes de code

- Selon les définitions , le nombre de lignes d'un même programme peut varier de 1 à 5
- Que mesurer avec ce nombre ?
  - L'effort ?
  - Certaines lignes étant plus difficiles à coder que d'autres, il faudrait une pondération de ces lignes
  - Fonctionnalité du produit ?
  - Selon le langage, la même fonctionnalité prendra plus ou moins de lignes

## LES BESOINS D'ESTIMATION : Les niveaux :

- à un **Niveau** correspond un objectif avec son degré de précision

Niveau :	Cas :
<b>Projet</b>	Ex : projet de gestion des commandes( GC )
<b>Etape</b>	Ex : étude préalable du projet GC
<b>Phase</b>	Ex : phase « observation » de l'étude préalable du projet GC
<b>Tâche</b>	Ex : spécification de la fonction « achat »

## Formules de calcul :

Charge en mois-personne = a (kisl)<sup>b</sup>

Délai normal en mois = c (charge en mois-personne)<sup>d</sup>

Taille moyenne de l'équipe = charge/délai.

- avec **kisl** = *nombre de milliers d'instructions sources livrées*, c'est-à-dire le nombre de milliers de lignes de programme source testées.
- Les paramètres a, b, c et d prennent des valeurs différentes selon la catégorie de projet. En effet, l'analyse statistique a conduit à distinguer trois types de projet : **simple, moyen et complexe**.

## Types de projet : simple, moyen et complexe.

- Un projet est **simple** si le logiciel comporte moins de 50 000 instructions, si les spécifications sont stables et le développement effectué par une petite équipe.
- Un projet est **moyen** si le logiciel comporte entre 50 000 et 300 000 instructions.
- Un projet est **complexe** si le logiciel comporte plus de 300 000 instructions et si l'on prévoit une équipe nombreuse ; il s'applique souvent à un domaine nouveau.

## Formules du modèle Cocomo

Type projet	Charge en mois-personne	Délai en mois
Simple	Charge = 2,4 (kisl) <sup>1,05</sup>	D = 2,5 (Charge) <sup>0,38</sup>
Moyen	Charge = 3 (kisl) <sup>1,12</sup>	D = 2,5 (Charge) <sup>0,35</sup>
Complexé	Charge = 3,6 (kisl) <sup>1,2</sup>	D = 2,5 (Charge) <sup>0,32</sup>

### exemple

- Soit par exemple un projet simple visant à développer un logiciel estimé à 40 000 instructions sources.
- Charge = ?

- Charge =  $2,4 * (40)^{1,05} = 116$  mois-personne (arrondi)
- Délai normal =  $2,5 * (116)^{0,38} = 12$  mois (arrondi)
- Taille moyenne de l'équipe =  $116 / 12 = 9$  personnes.

## Type de CoCoMo

- **COCOMO Basique** : Fournit une estimation rapide, basée uniquement sur la taille du logiciel.
- **COCOMO Intermédiaire** : Prend en compte des facteurs supplémentaires (fiabilité requise, complexité, etc.).
- **COCOMO Détailé** : Ajoute des facteurs détaillés comme l'efficacité de l'équipe et l'expérience.

## CoCoMo Intermédiaire : Risque sur l'estimation

- 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

## Démarche

- Estimation de la taille (Nb d'instructions sources)
- Calcul de la charge « brute »
- Sélection des facteurs correcteurs
- Application des facteurs correcteurs à la charge brute pour obtenir la charge « nette »
- Evaluation du délai normal

## Les facteurs logiciels :

- Fiabilité du logiciel : influence forte si exigence dans ce sens
- Base de données : mesuré par le ratio
  - (volume de données générées 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

## Les facteurs matériels

- 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 ».

## 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 Avancé

- **Principe :** Ce modèle divise le projet en **sous-composants** et applique l'estimation COCOMO pour chacun d'eux en tenant compte des variations de complexité et de taille entre modules.
- **Exemple :** Pour un projet avec plusieurs modules (UI, backend, API), chaque composant est estimé séparément en fonction de sa taille en LOC et de ses facteurs de coût uniques, puis les résultats sont additionnés pour obtenir l'estimation globale du projet.

## Différence entre COCOMO Intermédiaire et COCOMO Avancé

- **COCOMO Intermédiaire** : Utilise les multiplicateurs d'effort globaux pour ajuster les estimations en fonction de l'environnement du projet (comme les facteurs de produit, d'équipe, et de projet).
- **COCOMO Avancé (COCOMO 81 Détail)** : Effectue une décomposition plus poussée. Au lieu d'une estimation globale, chaque phase ou sous-système du projet est évalué séparément, en appliquant les facteurs d'effort spécifiques à chaque partie du projet, puis en cumulant pour obtenir l'estimation finale.

## COCOMO et la maintenance

Effort annuel de maintenance (en HM)

=

Taux annuel de modification (% de code modifié)

x

Temps de développement (en HM)

NB: On peut tenir compte des facteurs de coût qui ont un impact sur le coût de la maintenance et calculer un facteur d'ajustement d'effort.

## La méthode de répartition proportionnelle

- le niveau de l'estimation : **l'étape**
- 3 modalités :
  - à partir de l'estimation de la charge globale du projet (répartition descendante)
  - à partir de l'évaluation d'une des étapes ( utilisation ascendante )
  - au cours du projet, après observation du temps consommé sur les étapes à l'amont (utilisation dynamique)

- Emploi de Ratios :
  - cas : ratios constatés sur d'autres projets
- Ordre de grandeur

<b>Etape</b>	<b>Ratio</b>
<b>Etude préalable</b>	10% du projet
<b>Etude détaillée</b>	23% du projet
<b>Etude technique</b>	10% de la charge de réalisation
<b>Réalisation</b>	2 fois la charge de l'étude détaillée
<b>Mise en œuvre</b>	35% de la charge de réalisation

## Cas de l'étude préalable :

- 3 phases : Observation, Conception/organisation, Appréciation

- la phase « observation » :

si  $N$  = nombre d'interlocuteurs à enquêter

$P_i$  = poids affecté à chaque interlocuteur ( de 0,5 à 3 jours )

alors :

la charge de la phase « observation » = Somme (  $P_i$  ) pour  
 $i = 1$  à  $N$

- les 2 autres phases = Charges déduites de la phase « observation »

## ► ordre de grandeur

Phase	Pourcentage / charge d'étude préalable
Observation	30-40%
Conception/organisation	50-60%
Appréciation	10%

## Méthode des Points de Fonction (Function Point Analysis - FPA)

- Un **point de fonction** est une unité de mesure utilisée pour estimer la taille fonctionnelle d'un système ou d'une application logicielle, sans se baser sur les lignes de code (LOC) mais plutôt sur les fonctionnalités que le logiciel offre aux utilisateurs.
- La méthode des points de fonction se base sur l'idée que chaque fonction du logiciel (comme la saisie de données, le traitement d'informations, les sorties d'affichage, etc.) peut être attribuée un certain nombre de points en fonction de sa complexité.
- Le calcul des points de fonction prend en compte :

### 1. Types de Fonctions :

- **Entrées Externes (EE)** : les données ou commandes que les utilisateurs entrent dans le système (ex. formulaire de connexion).
- **Sorties Externes (EO)** : les données que le système affiche ou fournit aux utilisateurs (ex. génération de rapports).
- **Requêtes Externes (EQ)** : les interactions utilisateurs qui incluent des entrées et des sorties (ex. requêtes d'informations via une interface).
- **Fichiers Internes (ILF)** : les fichiers ou bases de données internes que le système doit gérer (ex. gestion des utilisateurs dans la base de données).
- **Fichiers Externes (EIF)** : les fichiers externes utilisés par le système mais gérés par d'autres systèmes (ex. fichiers clients d'un autre système).

**2. Complexité de Chaque Fonction :** Selon la méthode des points de fonction, chaque fonction est catégorisée en fonction de sa complexité :

- **Simple**
- **Moyenne**
- **Complex**
- Chaque niveau de complexité est associé à un nombre fixe de points de fonction, déterminé par des tables de pondération.

**3. Calcul Total des Points de Fonction :** Une fois que chaque fonction est évaluée, on additionne tous les points pour obtenir la **taille fonctionnelle** du logiciel en points de fonction.

## Tables de Pondération

Type de Fonction	Simple	Moyenne	Complex
Entrée Externe (EE)	3	4	6
Sortie Externe (EO)	4	5	7
Requête Externe (EQ)	3	4	6
Fichier Interne (ILF)	7	10	15
Fichier Externe (EIF)	5	7	10

- **Exemple :**
- Prenons un projet de développement d'un logiciel de gestion des ressources humaines (RH) avec les fonctionnalités suivantes :
  - Saisie des employés (complexité simple) : 4 points de fonction.
  - Génération de rapports (complexité moyenne) : 7 points de fonction.
  - Gestion des départements (complexité élevée) : 10 points de fonction.
- Le total des points de fonction est donc de 21. Sachant qu'un point de fonction prend en moyenne 5 heures à développer, l'équipe estime la charge de travail totale à :
 
$$21 \text{ points} \times 5 \text{ heures} = 105 \text{ heures.}$$

## Estimation du nombre de jours de travail

- Si taille fonctionnelle du projet est de 34 points, et
- si nous savons qu'un développeur traite en moyenne **5 points de fonction par jour**, nous pouvons estimer la charge de travail nécessaire pour compléter le projet.

$$\frac{34 \text{ points}}{5 \text{ points/jour}} = 6.8 \text{ jours}$$

- Ainsi, on peut estimer que le développement de ce système nécessitera environ **7 jours de travail** pour un développeur.

## Avantages des Points de Fonction

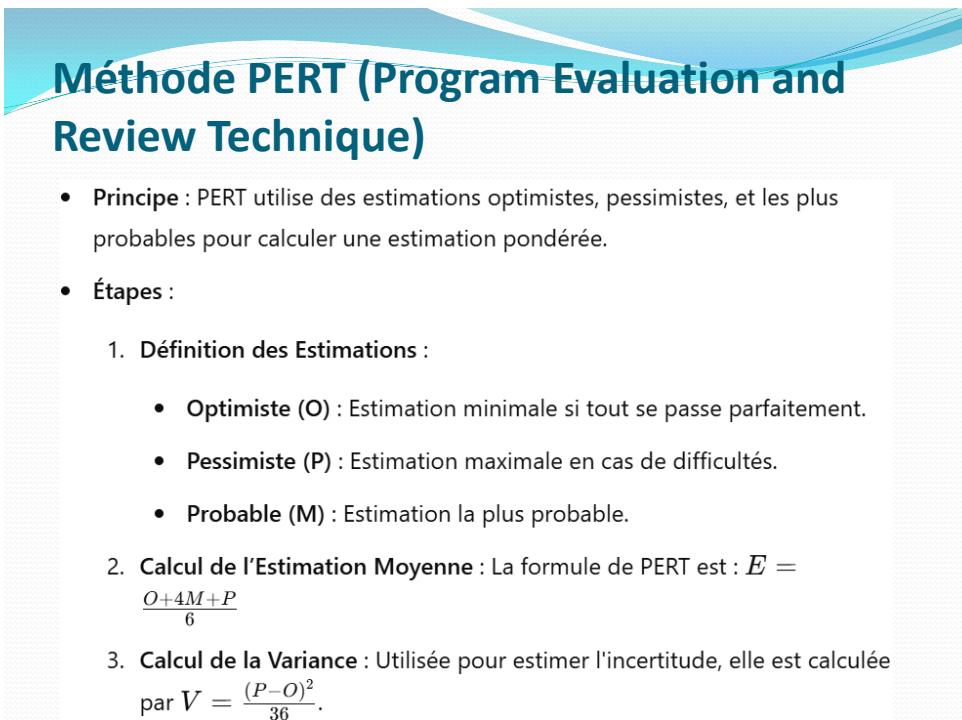
- **Indépendance du Code** : Les points de fonction ne sont pas liés à une technologie ou un langage particulier, ils mesurent les fonctionnalités offertes.
- **Précision pour Estimation** : Ils permettent de faire des estimations précises pour les projets, surtout en phase de planification.
- **Utiles pour Comparaisons** : Les points de fonction facilitent la comparaison de différents projets en fonction des fonctionnalités.

## Méthode de la Décomposition (Work Breakdown Structure - WBS)

- **Principe** : Le WBS découpe le projet en sous-tâches gérables et estimables individuellement.
- **Étapes** :
  1. **Décomposition** : Diviser le projet en sous-ensembles de tâches spécifiques.
  2. **Estimation des Tâches** : Estimer la charge de travail de chaque tâche de manière individuelle.
  3. **Agrégation** : Additionner les charges de toutes les tâches pour obtenir une estimation globale.



- **Avantages :** Convient aux projets complexes car il permet de visualiser chaque élément individuellement.
- **Exemple :** Pour un projet de développement d'application, les tâches peuvent inclure l'analyse, le développement d'interface utilisateur, et les tests, avec des charges spécifiques pour chacune.



- **Avantages :** Permet de tenir compte de l'incertitude.
- **Exemple :** Si pour une tâche, les estimations sont 5 jours (optimiste), 8 jours (probable), et 12 jours (pessimiste), alors :  $E = \frac{5+4\times8+12}{6} = 8$  jours

## Méthode des Estimations de Groupe (Delphi)

- **Principe :** Faire appel à plusieurs experts pour obtenir une estimation par consensus.
- **Étapes :**
  1. **Session d'Estimation :** Les experts fournissent des estimations de manière indépendante.
  2. **Synthèse et Discussion :** Comparer les estimations, discuter des écarts et ajuster.
  3. **Itérations :** Répéter jusqu'à obtenir une convergence vers une estimation finale.

- **Avantages** : Utilise l'expertise collective, réduisant le risque d'erreur individuelle.
- **Exemple** : Dans un projet complexe, les experts donnent des estimations de 100, 120, et 110 jours. Après discussion, ils conviennent d'une estimation de 115 jours.

## Références Bibliographiques :

- "*L'essentiel de la gestion de projet : Succès et pièges des projets*" 3ème édition, Pearson, 2016.  
Ce livre explore des études de cas et donne des conseils pour comprendre les succès et les échecs des projets en gestion de projet. Il met l'accent sur les compétences pratiques.
- "*A Guide to the Project Management Body of Knowledge (PMBOK Guide)*", 7th Edition, Project Management Institute, 2021.  
Ce guide est un standard mondial de la gestion de projet. Il présente les meilleures pratiques, les processus de gestion, et les outils indispensables pour le suivi de projet.
- **Microsoft Project** – Un des outils les plus utilisés pour planifier, suivre et gérer des projets de toutes tailles, avec des fonctionnalités pour les diagrammes de Gantt et les WBS.