

# Management de projet

## SOMMAIRE

<b>SOMMAIRE</b>	<b>1</b>
<b>QU'EST-CE QU'UN PROJET</b>	<b>3</b>
Première approche	3
Qu'est-ce que la gestion d'un projet ?	5
Approche structurelle : le triangle de la gestion de projet	5
Approche dynamique	5
Le problème du risque	6
L'origine du risque : aspects économiques d'un projet	6
Définition du risque	6
Quels sont les risques ?	6
Statistiques (plus ou moins récentes !) sur les échecs d'un projet	7
Quelles sont les conséquences ?	7
Comment limiter les risques ?	7
Les deux critères de découpage d'un projet	8
Découpage temporel	8
Découpage structurel	8
<b>METHODOLOGIE : CYCLE EN V ET AGILITE</b>	<b>9</b>
<b>ESTIMATION DES CHARGES</b>	<b>10</b>
Présentation	10
7 méthodes d'estimation	11
Un exemple de la méthode Delphi	12
La méthode Cocomo	13
La méthode d'évaluation analytique	15
La méthode des points de fonction d'Albrecht	16
Les composants fonctionnels	16
La complexité des composants fonctionnels	17
Le nombre de points de fonction des composants fonctionnels	18
Le calcul	18
La transformation du nombre de point en charge	18
Conclusion sur la méthode	18
La méthode de répartition proportionnelle	19
Conclusion	19

<b>PLANIFICATION</b>	<b>20</b>
<b>Introduction</b>	<b>20</b>
<b>Le PERT</b>	<b>21</b>
Présentation	21
Formalisme	21
Les notions	21
<b>Le GANTT</b>	<b>23</b>
Présentation	23
Diagramme de GANTT	23
MS-Project	25
<b>ESTIMATION DES COUTS</b>	<b>26</b>

# QU'EST-CE QU'UN PROJET

## Première approche

### Définition du dictionnaire Petit Robert

- 1) Image d'une situation, d'un état que l'on pense atteindre.
- 2) Travail préparatoire.

### Définition normalisée ISO 10006 :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 telles que les **contraintes** de délais, de coûts et de ressources.

### Définition normalisée AFNOR

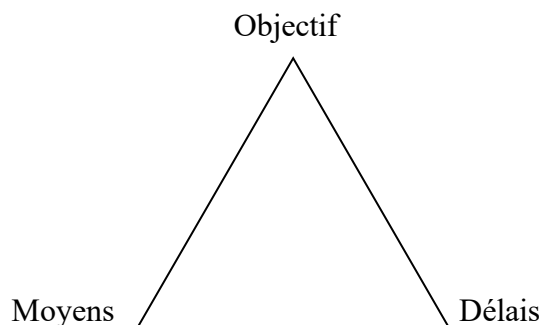
Un projet est un ensemble d'**actions** à réaliser pour satisfaire un **objectif** défini, dans le cadre d'une **mission** précise, et pour la réalisation desquelles on a identifié non seulement **un début**, mais aussi **une fin**.

### En résumé

Un projet vise à l'obtention d'un résultat satisfaisant pour son destinataire mais aussi pour son fournisseur.

### Le triangle du projet : objectif, moyens, délais

Chaque définition met l'accent sur des activités finalisées et soumises à contraintes de temps et de moyens.



### Les objectifs centraux de toute méthode

- ⇒ Répondre aux besoins et même aider à l'exprimer : **meilleure qualité fonctionnelle** (bien externe).
- ⇒ Répondre au **meilleur coût**.
- ⇒ Répondre avec les **meilleurs délais**.
- ⇒ Répondre avec une architecture qui permet de **faciliter les évolutions** (évolutions au meilleur coût et dans les meilleurs délais) = répondre avec **la meilleure qualité technique** (bien interne).

### Le trio impossible : Vite - Bien - pas Cher

- ⇒ Vite et Bien      => Cher
- ⇒ Vite et pas Cher      => pas Bien
- ⇒ Bien et pas Cher      => pas Vite

### Conclusion

- ⇒ La méthode s'adapte toujours aux contraintes de temps et d'argent.

## Qu'est-ce que la gestion d'un projet ?

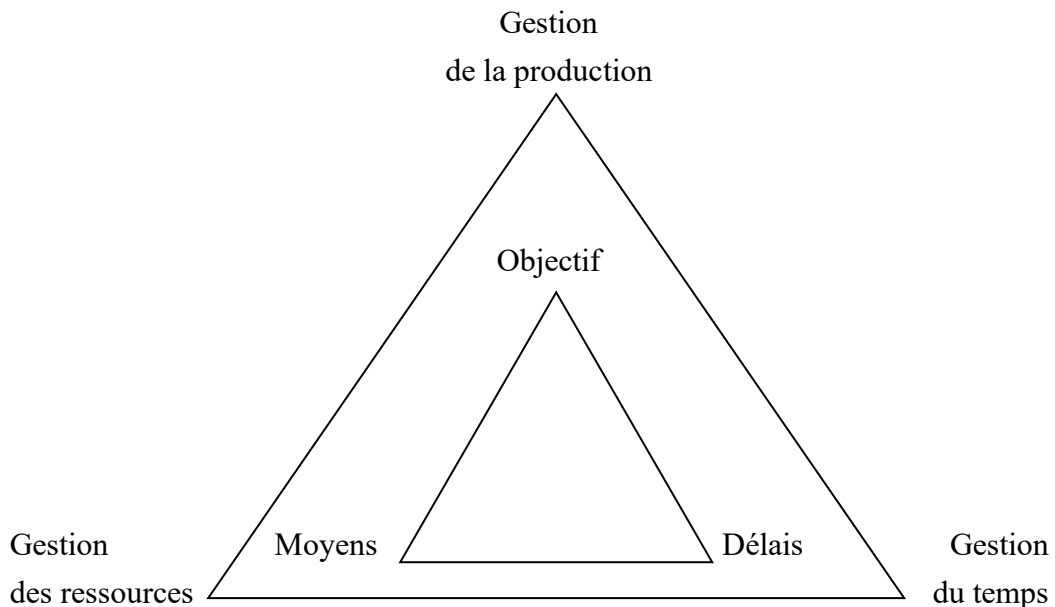
La gestion d'un projet a pour but de :

- mener un projet à son terme
- en prenant en compte les contraintes
- et en faisant face aux imprévus.

On parle aussi de « management » d'un projet, de « pilotage » ou de « conduite » d'un projet. Les termes « pilotage » et « conduite » peuvent être considéré comme plus restrictifs que ceux de « gestion » ou de « management ».

## Approche structurelle : le triangle de la gestion de projet

De là, le triangle de la gestion de projet : production (objectif), ressources (moyens), temps (délais).



## Approche dynamique

L'activité de gestion se décompose en trois activités :

- **Analyser** : déterminer les tâches à réaliser. Découper le projet. Estimer les charges.
- **Organiser** : affecter les ressources humaines, matérielles, temporelles aux tâches. Planification.
- **Piloter** : suivre l'avancement du projet. Corriger l'analyse et l'organisation en fonction des imprévus.

## **Le problème du risque**

### **L'origine du risque : aspects économiques d'un projet**

Un projet vise à l'obtention d'un résultat satisfaisant pour son destinataire (le client) mais aussi pour son fournisseur.

#### **Intérêts contradictoires des intervenants**

Le client souhaite recevoir le plus possible, le plus vite possible pour le moins cher possible.

Le fournisseur souhaite fournir le moins possible, le plus lentement possible pour au prix le plus élevé possible.

#### **Négociation**

La négociation consiste à trouver l'intérêt commun des intervenants.

#### **Mode de relations entre les intervenants**

- Relation d'affaire : pas d'amitié en affaire ! Relation donnant-donnant.
- Pas d'à peu près en affaires : objectifs, délais et coûts doivent être détaillés et formalisés.
- Discrétion : éviter de donner des informations qui mettent en position de faiblesse.
- Bonne foi : convaincre sans mentir !

### **Définition du risque**

Chez les assurances :  $\text{Risque} = \text{Coût d'un événement} \times \text{fréquence de cet événement}$   
C'est peu opérationnel pour les projets informatiques.

Risque = possibilité qu'un projet ne s'exécute pas conformément aux prévisions de

- dates d'achèvement
- coût
- spécification

Il y a risque si les écarts par rapport aux prévisions sont considérés comme difficilement acceptables voire inacceptables.

### **Quels sont les risques ?**

- Dérapage des délais
- Augmentation des coûts
- Inadéquation aux besoins du client
- Un grand nombre d'erreurs résiduelles

## Statistiques (plus ou moins récentes !) sur les échecs d'un projet

Causes de l'échec d'un projet (non exclusives : total > 100%)	
Dépassements d'échéances	87%
Dépassement des coûts	56%
Non conformité aux spécifications	45%

On peut avoir en même temps, par exemple, dépassement des échéances et dépassement des coûts.

Causes de l'échec d'un projet (exclusives : total = 100%)	
Problème de management	60%
Problème de spécifications	30%
Problème technique	10%

Les problèmes de spécifications et les problèmes techniques (30% + 10%) rejoignent les problèmes de non conformités aux spécifications (45%) du tableau précédent.

## Quelles sont les conséquences ?

- Suppression de nouvelles commandes
- Non-paiement des travaux
- Perte du client
- Ternissement de l'image du développeur

## Comment limiter les risques ?

- Par l'expérience : en évitant de recommencer ses erreurs.
- Par la méthode : en évitant de recommencer les erreurs des autres !

## Les deux critères de découpage d'un projet

Deux critères de découpage :

- Temporel : le « phasage »
- Structurel : l'architecture

### Découpage temporel

Répartition du travail dans le temps : « phasage ».

Succession d'étapes et de phases, avec leur **date de début** prévue et **date de fin** visée.

#### Principales méthodes

- Agile
- Cycle en V

#### Avantages

- Baliser et guider le travail.
- Poser des jalons.
- Le client peut valider la conformité à ses exigences et éventuellement réorienter le travail.

### Découpage structurel

Organisation du travail en se basant sur la structure du produit final.

#### Méthodes

- Analyse des sous-systèmes
- Analyse des incréments
- Modélisation des données
- Analyse des
- etc.

#### Avantages

- Maîtrise du projet par le découpage en sous projet.
- Répartition des tâches et des responsabilités.
- Réduction des délais par le parallélisme des tâches.
- Possibilité d'un développement incrémental



## METHODOLOGIE : CYCLE EN V ET AGILITE

Le **cycle en V** est le modèle classique du développement d'un logiciel. Il est plus restreint que le cycle du projet (car limité à la MOE), mais sa compréhension, par un informaticien, permet de cadrer l'ensemble de la problématique de la conduite d'un projet.

Autour du cycle en V, viennent les **méthodes agiles** et les **méthodes itératives et incrémentales**.

Cf. le **document « Introduction à la méthode »**.

# ESTIMATION DES CHARGES

## Présentation

### Charge

Quantité de travail nécessaire, indépendamment du nombre de personnes qui vont réaliser le travail. Elle s'exprime en durée-homme : jour-personne, mois-personne, année-personne.

### Taille des projets

Charge (en mois-personne)	Taille du projet
< 6	Très petit projet
6 à 12	Petit projet
12 à 30	Projet moyen
30 à 100	Grand projet
> 100	Très grand projet

### Durée

Elle dépend du nombre de personnes.

60 mois-personnes avec une personne : 5 ans. Avec 10 personnes, ça peut-être un an.

### Pourquoi faire une estimation

Évaluer le coût

Évaluer la charge

Évaluer la durée

Évaluer la rentabilité

## **7 méthodes d'estimation**

### **1 : la méthode de Parkinson**

**Le travail se dilate jusqu'à remplir tout le temps disponible**, puis se contracte pour entrer dans le temps restant ! C'est une non-méthode classique : aviser en fonction de la date de rendu prévue, avec le risque de plus ou moins reporter au lendemain pour finalement s'y mettre la veille du rendu !

### **2 : la méthode du marché**

**La charge correspond au prix à proposer pour remporter le marché.** Encore une non-méthode classique : prendre le marché et aviser ensuite !

### **3 : la méthode Delphi (exemple plus bas)**

Elle est basée sur un **jugement d'expert qui juge par expérience et par analogie** avec des projets antérieurs.

### **4 : la méthode Cocomo (exemple plus bas)**

Calcul à partir de la **taille estimée du logiciel**.

### **5 : la méthode d'évaluation analytique (exemple plus bas)**

Analyse la répartition structurelle des charges.

Cette méthode dénombre les parties techniques du programme à réaliser par nature et par degré de difficulté.

C'est un calcul fait à partir de la description des composants techniques.

### **6 : la méthode des points de fonction d'Albrecht (exemple plus bas)**

Analyse la répartition structurelle des charges.

Cette méthode dénombre les parties fonctionnelles externe du programme à réaliser par nature et par degré de difficulté.

C'est un calcul fait à partir de la description des composants fonctionnels

### **7 : la méthode de répartition proportionnelle (exemple plus bas)**

**Répartition dynamique des charges.**

Cette méthode répartit les charges proportionnellement par étape de la réalisation (conception, architecture, analyse détaillée, codage, tests).

Elle suppose d'avoir une estimation globale du projet. Cette estimation peut être fournie par la méthode de Parkinson (1), la méthode du marché (2) ou la méthode de Delphi (3).

## **Un exemple de la méthode Delphi**

### **Répartition des charges**

Études : 10 jours, 1 chef de projet

Conception : 20 jours, 1 ingénieur (ou CP junior)

Développement : 60 jours : 2 développeurs

Livraison : 15 jours : 1 CP (5 jours) et un ingé (10 jours).

### **Rémunération (à la louche !)**

CP : 1500 euros/jour

Ingé : 1000

Dév : 500

Tarifs de SSII

### **Totaux**

Total :  $(CP : 10+5)*1500 + (Ingé : 20+10)*1000 + (dév : 60)*500 = 82\ 500$

+ 40% de marge : 115 500

## La méthode Cocomo

Cocomo : modèle de construction des coûts (Constructiv Cost Model)

Barry Boehm (celui de la spirale de Boehm), 1981.

### Les coefficients de Boehm

Deux hypothèses :

- Un informaticien sait donner une évaluation de la taille en ligne d'instruction du logiciel à développer.
- La charge d'écriture d'un nombre de lignes données est la même pour tout langage de 3<sup>ème</sup> génération.

À partir de ces hypothèses et de l'analyse statistique d'une centaine de projets réalisés, Boehm a pu calculer des coefficients de corrélation entre la taille du logiciel et la charge consommée.

$$\begin{aligned}\text{Charge en mois-personne} &= a * (kisl)^b \\ \text{Délai normal en mois} &= c (\text{charge en mois-personne})^d\end{aligned}$$

Type de projet	Nombre d'instructions	a	b	c	d
Simple	Moins de 50 000	2,4	1,05	2,5	0,38
Moyen	Entre 50 000 et 300 000	3	1,12	2,5	0,35
Complexe	Plus de 300 000	3,6	1,2	2,5	0,32

### Le facteur correcteur

En deuxième étape, Boehm applique un facteur correcteur, calculé statistiquement à partir d'une analyse de certaines caractéristiques du projet :

**Sur le plan du logiciel :** importance de la fiabilité. Importance du temps d'exécution. Complexité algorithmique. Taille de la base de données.

**Sur le plan du personnel :** compétence et expérience des concepteurs. Compétence des développeurs. Connaissance de l'environnement technique. Expérience du langage.

**Sur le plan du projet :** utilisation d'une méthode. Utilisation d'un AGL. Contrainte de délai.

**Sur le plan du matériel de développement :** problème de taille de la mémoire ; problème de stabilité de l'environnement ; problème de la disponibilité d'un environnement de test.

Pour chaque point, la méthode définit un facteur.

$$\text{Charge nette} = \text{produit}(\text{valeurs des facteurs correcteurs}) * \text{charge brute}$$

### Conclusion

Estimer au préalable le nombre d'instructions sources du logiciel à développer, ce n'est pas facile.


L'analyse statistique des facteurs correcteurs s'est faite sur un échantillon trop restreint.

Les facteurs correcteurs font varier l'estimation brute de 1 à 17 !

Donc, la méthode Cocome, tout le monde connaît... et personne ne pratique !

## **Cocomo 2**

La méthode Cocomo 2 a été rendue publique en 1997. Elle affine la méthode initiale et s'appuie sur la méthode des points de fonction.



## La méthode d'évaluation analytique

### Principe

Elle consiste à dénombrer les parties du programme à réaliser par nature et par degré de difficulté.

On considère trois niveaux de difficultés : facile, moyen et difficile.

A chaque partie, on affecte un « poids » de travail : une durée en jour.

On évalue la charge d'intégration à 10% de la charge de réalisation.

On évalue la charge d'encadrement à 20% de la charge de réalisation.

Pour les projets dont la charge est comprise entre 3 et 30 mois-personne :

$$\text{Délai} = 2,5 * (\text{charge en mois homme})^{1/3}$$

### Exemple

<i>Complexité</i>	<i>Facile</i>			<i>Moyen</i>			<i>Difficile</i>			<b>Total</b>
<b>Type de composant</b>	Nb	Pds	Charge	Nb	Pds	Charge	Nb	Pds	Charge	
Menu	6	0,5	3	3	1	3				6
Écrans	8	0,5	4	20	01	20	12	2	24	48
Requêtes SQL	10	0,5	5	8	2	16	8	3	24	45
<b>Total réalisation</b>										99
Intégration									10%	9,9
Encadrement									20%	19,8
<b>Total projet</b>										<b>128,7</b>

Charge = 128,7 jours personne.

Soit, environ 6 mois personne (à raison de 21 jours par mois)

Délai =  $2,5 * 6^{1/3} = 4,5$  mois

## La méthode des points de fonction d'Albrecht

Le principe de la méthode est de faire une estimation à partir d'une description externe du futur système, c'est-à-dire une description de ses composants fonctionnels.

La méthode identifie **5 composants fonctionnels** et 3 degrés de complexité pour chaque composant fonctionnel.

À chaque type et à chaque degré, la méthode affecte un nombre de « points »

On calcule, pour un projet donné, son poids en « points de fonction »

## Les composants fonctionnels

### Les Groupe de Données

#### ➤ **GD**

Un Groupe de Données, GD, est un ensemble de données que l'utilisateur les perçoit comme logiquement liées.

Un groupe de données est un sous-MCD qui correspond à une unité fonctionnelle.

#### ➤ **GDI**

Un Groupe de Données Interne, GDI, est un GD créé et mis à jour à l'intérieur du domaine d'étude.

#### ➤ **GDE**

Un Groupe de Données Externe, GDE, est un GD qui n'est qu'interrogé par le domaine d'étude. Il est créé et mis à jour par un autre domaine. Un GDE est toujours un GDI dans un autre domaine.

#### ➤ **DE**

Un GD est composé de données élémentaires : DE. On compte une DE par attribut, y compris les clés étrangères.

#### ➤ **SLD**

Un GD est parfois composé de plusieurs **sous-ensembles logiques de données : SLD**.

Le SLD est un sous-groupe de données élémentaires reconnaissables par l'utilisateur.

Deux SLD peuvent avoir des données en commun.

Le SLD a une granularité plus fine que le GD. C'est par exemple un ensemble de données dans un écran de saisie.

#### ➤ **GDR**

Un **GDR, groupe de données référencées**, est un GDI ou un GDE utilisé par une ENT (en lecture ou en mise à jour) ou par une SOR ou une INT (en lecture uniquement).

### Les Fonctions

#### ➤ **ENT**

Une ENT est une fonction élémentaire, significative pour l'utilisateur, qui permet d'introduire des données à l'intérieur du domaine.



Par simplification, une entrée correspond à un écran de saisie ou à une réception de données provenant d'une autre application et provoquant une mise à jour.

A chaque GDI, correspond au moins une ENT permettant sa mise à jour.

Chaque ENT permet de saisir un certain nombre de champs, qui sont des données élémentaires : DE.

#### ➤ **SOR**

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.

Par simplification, une sortie correspond à la génération d'un état, d'un écran de visualisation ou d'un message à destination d'une autre application, avec des **données calculées**.

Sur chaque SOR figurent un certain nombre de champs, qui sont des données élémentaires : DE.

#### ➤ **INT**

Une INT est une fonction élémentaire qui a pour résultat l'extraction de données.

Une INT ne fait pas de calcul et ne met à jour aucun GDI.

Sur le résultat de chaque INT figurent un certain nombre de champs, qui sont des données élémentaires : DE.

### **La complexité des composants fonctionnels**

On détermine la complexité des composants à partir :

- du nombre de SLD dans les GDI ou GDE
- du nombre de GDR dans les ENT et dans les SOR ou les INT
- du nombre de DE dans les SLD et les GDR

<b>GDI ou GDE</b>	<b>1 à 19 DE</b>	<b>20 à 50 DE</b>	<b>51 DE ou plus</b>
<b>1 SLD</b>	Faible	Faible	Moyenne
<b>2 à 5 SLD</b>	Faible	Moyenne	Élevée
<b>6 SLD ou plus</b>	Moyenne	Élevée	Élevée

<b>ENT</b>	<b>1 à 4 DE</b>	<b>5 à 15 DE</b>	<b>16 DE ou plus</b>
<b>0 ou 1 GDR</b>	Faible	Faible	Moyenne
<b>2 GDR</b>	Faible	Moyenne	Élevée
<b>3 GDR ou plus</b>	Moyenne	Élevée	Élevée

<b>SOR ou INT</b>	<b>1 à 5 DE</b>	<b>6 à 19 DE</b>	<b>20 DE ou plus</b>
<b>0 ou 1 GDR</b>	Faible	Faible	Moyenne
<b>2 à 3 GDR</b>	Faible	Moyenne	Élevée
<b>4 GDR ou plus</b>	Moyenne	Élevée	Élevée

## Le nombre de points de fonction des composants fonctionnels

<i>Degré de complexité du composant</i>	Faible	Moyenne	Elevée
Nombre de points de fonction			
<b>GDI</b>	7	10	15
<b>GDE</b>	5	7	10
<b>ENT</b>	3	4	6
<b>SOR</b>	4	5	7
<b>INT</b>	3	4	6

## Le calcul

- 1) Dénombrement des différents composants fonctionnels et de leur degré de complexité.
- 2) Somme des points de fonction
- 3) Ajustement facultatif
- 4) Transformation du nombre de point en charge

## La transformation du nombre de point en charge

Le coefficient de transformation est variable selon l'environnement matériel et humain.  
Il est recommandé que chaque entreprise établisse ses propres coefficients !

## Conclusion sur la méthode

Avantage : c'est analytique.  
Défaut : c'est empirique pour les coefficients.

## La méthode de répartition proportionnelle

Cette méthode s'appuie sur le découpage temporel de référence (le cycle de vie MERISE, par exemple).

Elle définit un ratio pour chaque étape par rapport à la totalité.

Étape	Ratio
Étude préalable	10 %
Étude détaillée	30 %
Étude technique	5 à 15 % de la charge de réalisation
Réalisation	60 % : 2 fois la charge d'étude détaillée

### Trois façons d'utiliser le ratio

- Estimation de la charge globale et déduction des charges d'étape.
- Estimation d'une étape et déduction des autres étapes.
- Observation de la durée d'une étape et déduction des autres étapes.

### Estimation des charges complémentaires

Tâche	Ratio
Encadrement du projet à l'étape de réalisation	20 % de la charge de réalisation
Encadrement du projet aux autres étapes	10 % de la charge de l'étape
Recette	20 % de la charge de réalisation
Documentation utilisateur	5 % de la charge de réalisation

## Conclusion

Les méthodes d'estimation des charges sont difficiles à mettre en application.

Dans la pratique, on croise la méthode de **répartition proportionnelle** (cycle de vie) avec la méthode **Delphi** (expérience) et avec des **méthodes analytiques dégradées** (méthode d'évaluation analytique et méthode des points de fonction : cycle d'abstraction), ces méthodes analytiques permettant d'évaluer la complexité du travail à réaliser.

# PLANIFICATION

## Introduction

La planification constitue une activité essentielle de la phase d'initialisation. Elle doit intégrer les contraintes de temps et de ressources humaines et financières.

### La planification a deux phases

#### ➤ *La phase abstraite : PERT*

Cette phase ne s'intéresse ni aux dates ni aux ressources humaines.

Elle se divise en 3 étapes successives :

- Découper le projet en suivant le cycle de vie.
- Établir une liste des tâches et la **durée estimée** de chacune d'elles
- Analyser les contraintes d'ordonnancement (PERT) et établir une durée minimum du projet.

#### ➤ *La phase concrète : GANTT*

La phase concrète inscrit le projet dans le temps et affecte les ressources aux différentes tâches. Il s'agit d'établir le **planning** : le calendrier du projet.

Elle se divise en 3 étapes :

- Planifier le projet et intégrer les jalons : établir la durée réelle du projet.
- Organiser l'équipe de projet et affecter les ressources aux tâches
- Calculer les coûts

**Les jalons** : décision, livraison documentation, livraison de logiciel : installation, tests

## Le PERT

### Présentation

PERT : Programm Evaluation and Review Technique

Conçu à la fin des années 50 par la marine américaine, l'objectif du PERT est de **déterminer l'ordre d'exécution d'un ensemble de tâches** entre lesquelles des **contraintes d'antériorité** sont à respecter.

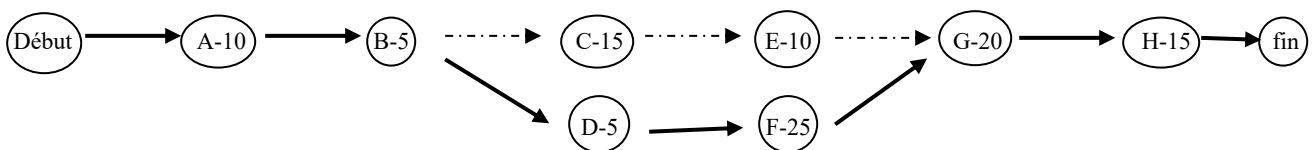
Le réseau PERT peut être utilisé à différentes mailles de décomposition.

### Formalisme

#### Tableau des tâches

Tâches	Libellé	Durée en jours	Tâches immédiatement antérieures	Durée totale min
A	Études préalables	10	/	10
B	Conception	5	A	15
C	Conception	15	B	30
D	Développement	5	B	20
E	Développement	10	C	40
F	Développement	25	D	45
G	Développement	20	E, F	65 (F)
H	Livraison	15	G	80
Fin	Fin	0	H	80

#### Graphe des tâches



### Les notions

#### Lien et type de lien

Les liens relient les tâches entre elles.

➤ *Lien fin-début*

Le lien fin-début est le plus courant.

Une tâche précède une autre. Pour toute tâche on précise quelles sont la ou les tâches qui la précèdent.

➤ ***Lien fin-fin***

La fin de l'un qui commande la fin de l'autre. Par exemple : la fin de la programmation commande la fin de l'encadrement de la programmation.

➤ ***Lien début-début***

Le début de l'un commande le début de l'autre. Par exemple : le début des interviews commande le début de la modélisation.

➤ ***Lien début-fin***

Le début de l'un commande la fin de l'autre. Par exemple : le début de l'exploitation d'un nouveau logiciel commande la fin de l'exploitation de l'ancien logiciel.

**Délai - avance - retard**

Un lien peut-être caractérisé par un délai, exprimé en jours. Le délai peut être positif ou négatif. Quand le délai est négatif, on parle d'avance. Par exemple la tâche « programmation » est suivie par la tâche « test » avec un délai négatif, une avance, de 10 jours.

Quand le délai est positif, on parle de retard. Par exemple, la tâche « installation d'un prototype » sera suivie par la tâche « prise en compte des suggestions » avec un délai positif, un retard, de 10 jours.

Tâche A  $\xrightarrow{+10}$  Tâche B

**Chemin critique - tâches critiques - durée totale minimum**

Le PERT permet de déterminer la **durée totale minimum** du projet.

Dans le tableau des tâches, on peut calculer la durée totale minimum pour chaque tâche : c'est la somme de la durée de la tâche et du maximum des durées totales minimum des tâches immédiatement précédentes.

Dans le tableau des tâches, on précise si nécessaire, dans la colonne de la durée totale minimum, le nom de la tâche précédente.

**Le chemin critique** est donné par la succession des tâches qui conduit à la durée totale minimum.

**Pour déterminer le chemin critique, on part de la fin, et on remonte les tâches en passant à chaque fois par la durée totale minimum.**

**Les tâches critiques** sont celles qui participent au chemin critique.

Le chemin critique signifie que si la durée d'une tâche du chemin critique augmente, alors c'est la durée totale du projet qui augmente d'autant.

Donc, aucun retard acceptable sur les tâches critiques.

Le chemin critique est visualisé par des flèches en gras : **————→**

**Parallélisme**

Le PERT permet de mettre à jour les possibilités de parallélisme dans l'exécution des tâches.

## Le GANTT

### Présentation

Technique d'ordonnancement inventée par Henri Gantt au début du XXe siècle pour représenter de façon graphique la répartition du travail en atelier.

C'est la plus connue et la plus populaire des techniques d'ordonnancement.

Le GANTT permet d'établir le planning : le calendrier du projet. Pour cela, il faut faire des hypothèses de ressources et affecter les tâches à des personnes ou à des profils de personnes.

On pourra faire des simulations selon la taille de l'équipe envisagée.

On prend en compte les contraintes de calendrier : jours non ouvrables, jours fériés, etc.

### Diagramme de GANTT

Tableau à 2 dimensions :

- En abscisse : le temps
- En ordonnée : soit les tâches, soit les personnes affectées aux tâches.

Les tâches sont représentées par des « barres » horizontales de longueur proportionnelle à leur durée.

### PERT complété

Dans le tableau des tâches, sont donnés : la durée et le ou les prédécesseurs de chaque tâche.

De là, on peut calculer : total minimum, début minimum, début maximum, fin minimum, et fin maximum.

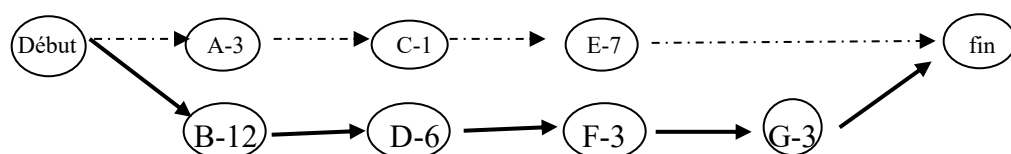
On commence par calculer le total minimum.

De là, on détermine le chemin critique.

Les min et max des tâches critiques sont faciles à calculer dans l'ordre chronologique.

Pour les autres, on part de la fin.

	Durée	Pred	Tot min	Déb min	Déb max	Fin min	Fin max
A	3	/	3	1	14	3	16
<b>B</b>	12	/	12	1	1	12	12
C	1	A, B	13 (B)	13	17	13	17
<b>D</b>	6	B	18	13	13	18	18
E	7	C	20	14	18	20	24
<b>F</b>	3	C, D	21 (D)	19	19	21	21
<b>G</b>	3	F	24	22	22	24	24
Fin	Fin	E, G	24 (G)				



### Diagramme de Gantt avec planification au plus tôt

On planifie les tâches au plus tôt et on montre les périodes d'étalement possible (en gris).

Périodes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Tâches																								
B	B																							
D													D											
F																			F					
G																						G		
A	A																							
C													C											
E														E										

### Diagramme de Gantt avec planification au plus tard

On planifie les tâches au plus tard et on montre les périodes d'étalement possible (en gris).

La planification au plus tard est adaptée à la méthode de Parkinson.

Périodes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Tâches																								
B	B																							
D													D											
F																			F					
G																						G		
A																								
C																								
E																								

### Diagramme de Gantt avec affectation des ressources

Périodes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Ressources																								
R1	B																							
													D											
																			F					
																						G		
R2	A																							



## Diagramme de Gantt optimisé

On évite l'attente entre A et C

Périodes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24		
Ressources																										
R1	B																									
													D													
																			F							
																					G					
R2											A															
													C													
														E												

### MS-Project

MS-Project est un logiciel qui permet de réaliser des diagrammes de GANTT.

Il fabrique aussi le PERT correspondant.

Il permet une estimation des coûts.

# ESTIMATION DES COUTS

L'estimation du coût consiste à savoir **qui, à quel prix, fait quoi pendant combien de temps.**  
 Etant donné qu'il peut y avoir des moments de sous charge, l'estimation des coûts se fait en fonction de l'estimation des charges et de la planification.

L'estimation des coûts est faite dans un tableau :

**En ligne** : les intervenants du projet (chef de projet, ingénieur, analyste programmeur, etc.)

**En colonne** : les tâches, regroupées par phases.

**Dans chaque case** : le nombre de jour de participation.

On peut faire des totaux par personne, par phase ou par tâche.

	Chef de projet	Ingénieur	Programmeur	Total
<b>1 : Conception</b>				
Prise de connaissance	2			2
Étude des données	2	5		7
Spécifications fonctionnelles	2	7		9
<b>Total conception</b>	<b>6</b>	<b>12</b>		<b>18</b>

<b>2 : Réalisation</b>				
Compléments de spécification	2	3		5
Écriture des programmes		22	45	67
Revue de code		5		5
Tests unitaires			8	8
<b>Total réalisation</b>	<b>2</b>	<b>30</b>	<b>53</b>	<b>85</b>

<b>3 : Intégration</b>				
Rédaction du plan de test	3			3
Tests d'intégration	1		6	7
<b>Total intégration</b>	<b>4</b>		<b>6</b>	<b>10</b>

<b>4 : Recette</b>				
Installation des composants		3	2	5
Recette	3		2	5
<b>Total recette</b>	<b>3</b>	<b>3</b>	<b>4</b>	<b>10</b>

<b>5 : Suivi de projet</b>				
Organisation et réunions	8	1	1	10
Suivi avancement	10			10
<b>Total suivi projet</b>	<b>18</b>	<b>1</b>	<b>1</b>	<b>20</b>

<b>Total général</b>	<b>33</b>	<b>46</b>	<b>64</b>	<b>143</b>
----------------------	-----------	-----------	-----------	------------

Les totaux par personne permettent ensuite de définir le coût de chaque intervenant par phase.

	Chef de projet	Ingénieur	Programmeur	Total
<b>Coût unitaire</b>	<b>700</b>	<b>600</b>	<b>400</b>	
<b>Total conception</b>	4 200	7 200		<b>11 400</b>
<b>Total réalisation</b>	1 400	18 000	21 200	<b>40 600</b>
<b>Total intégration</b>	2 800		2 400	<b>5 200</b>
<b>Total recette</b>	2 100	1 800	1 600	<b>5 500</b>
<b>Total suivi projet</b>	12 600	600	400	<b>13 600</b>
<b>Total général</b>	<b>23 100</b>	<b>27 600</b>	<b>25 600</b>	<b>76 300</b>