

## Master 1- Tronc Commun

Le master présente aux étudiants des concepts, des méthodes et des techniques identifiés actuellement dans le monde industriel et de recherche. Il leur permet d'acquérir les connaissances nécessaires d'analyse et de développement des systèmes logiciels, de la conception des architectures logicielles et des modèles des projets, de la maîtrise des concepts avancés des Interfaces Homme-Machine (IHM) et des langages de programmation.

A la fin de cette formation, l'étudiant sera capable de présenter de solutions numériques aux problèmes rencontrés dans le domaine et de développer sa propre entreprise spécialisée dans le développement de logiciels, ou d'intégrer une équipe de recherche en suivant un parcours de doctorat.

Unité d'Enseignement	Matière	Crédits	Coeff-icent	Volume horaire hebdomadaire			VHS 14-16 Semaines	Mode d'évaluation	
				Cours	TD	TP		Continu	Examen
UE Fondamentale Crédits : 18 Coefficients : 9	BDA: Bases de données avancées	4	2	1h30		1h30	45h	40%	60%
	PWA: Programmation web avancée	4	2	1h30		1h30	45h	40%	60%
	AAC: Algorithmique avancé et complexité	4	2	1h30		1h30	45h	40%	60%
	SR: Systèmes répartis	6	3	1h30	1h30	1h30	67h30	40%	60%
UE Méthodologie Crédits : 9 Coefficients : 4	MS: Modélisation et simulation	4	2	1h30		1h30	45h	40%	60%
	CP: Conduite de projets	5	2	1h30	1h30	1h	60h	40%	60%
UE Transversale Crédits : 3 Coefficients : 2	ANG1 : Anglais 1	2	1	1h30	1h30		45h		100%
	CDT : Corruption et déontologie de travail	1	1	1h30			22h30		100%
<b>Total Semestre 1</b>		30	15	12h	4h30	8h30	375h		

### Matière : Conduite de projets

1. Problématique de la gestion de projet
2. Découpage d'un projet et les modèles de développement.
3. Gestion de projet.
4. Gestion de la qualité
5. Inspection du logiciel
6. Estimation du coût du logiciel
7. Gestion des ressources humaines
8. Gestion de la configuration
9. Métriques du logiciel
10. Récapitulation de la gestion des projets



## Matière : Conduite de Projets

### Table des matières

La partie : Cours .....	1
Introduction .....	1
1. Problématique de la gestion de projet .....	1
1.1. Introduction .....	1
1.2. Définition d'un projet .....	2
1.3. Gestion d'un projet .....	2
1.4. La gestion d'un projet système d'information .....	4
1.5. Conclusion .....	5
2. Découpage d'un projet et les modèles de développement .....	6
2.1. Introduction .....	6
2.2. Critères de découpages: .....	6
2.3. Le cycle de vie standard .....	7
2.4. Le découpage classique .....	7
2.5. Les modèles de développement .....	8
2.6. Analyse des risques .....	11
2.7. Les modèles de cycle de vie spécifiques .....	12
2.8. Les 2 Visions Rassemblées: Le Modèle Itératif .....	12
2.9. Conclusion .....	12
3. Gestion de projet.....	13
3.1. Introduction .....	13
3.2. Activités de gestion .....	13
3.3. Planification du projet .....	13
3.4. La programmation du projet .....	15
3.5. Gestion des risques .....	16
3.6. Conclusion.....	17
4. Gestion de la qualité .....	20
4.1. Introduction .....	20
4.2. Processus basé sur la qualité.....	20
4.3. Assurance qualité et standards .....	21
4.4. Planification de la qualité .....	22
4.5. Contrôle de qualité .....	22
4.6. Mesures du logiciel et les métriques.....	23
4.7. Conclusion.....	24
5. Inspection du logiciel .....	25
5.1. Introduction .....	25
5.2. Avantages des inspections .....	25
5.3. Inspections .....	26
5.4. Conclusion.....	28
6. Estimation du coût du logiciel .....	29
6.1. Introduction .....	29
6.2. Productivité du logiciel.....	29



6.3.	L'estimation du coût d'un logiciel.....	30
6.4.	Analyse des points de fonction.....	32
6.5.	Conclusion.....	34
7.	Gestion des ressources humaines .....	35
8.	Gestion de la configuration.....	35
9.	Métriques du logiciel.....	35
10.	Récapitulation de la gestion des projets .....	35
La partie : Méthodes.....		36
1.	Méthode Planning de PERT .....	36
1.1.	Généralités.....	36
1.2.	BUT de la Méthode P.E.R.T.....	36
1.3.	Domaine d'Application de la Méthode P.E.R.T .....	36
1.4.	Conditions de mise en œuvre de la Méthode P.E.R.T .....	36
1.5.	Principe de la Méthode basée sur une Représentation Graphique.....	36
1.6.	Exploitation du réseau : .....	44
1.7.	La méthode MPM :.....	45
2.	Méthode Planning de PERT Probabiliste .....	46
2.1.	La première étape .....	46
2.2.	La deuxième étape .....	46
2.3.	La troisième étape.....	46
2.4.	Exercice Recette .....	47
2.5.	Fonction de répartition de la loi normale standard .....	49
2.6.	Inverse de fonction de répartition de la loi normale standard.....	49
3.	Méthode Planning de GANTT .....	50
3.1.	Généralités .....	50
3.2.	Utilisation de la matrice.....	51
3.3.	Calcul des marges totales .....	53
3.4.	Calcul des marges libres .....	53
3.5.	Utilisation des marges .....	53
3.6.	Diagramme de GANTT.....	54
La partie : Travaux Dirigés.....		56
1.	TD N°1 : Ordonnancement.....	56
1.1.	Exercice 1 .....	56
1.2.	Exercice 2 .....	56
1.3.	Exercice 3 .....	56
2.	TD N°1 : Ordonnancement Corrections .....	57
2.1.	Exercice 1 .....	57
2.2.	Exercice 2 .....	58
2.3.	Exercice 3 .....	59
La partie : Travaux Pratiques .....		60
1.	TP N°1 .....	60
2.	TP N°2 .....	60



# Conduite de Projets

## La partie : Cours

### Introduction

Plébiscitée par de plus en plus d'**entreprises**, la gestion de projet s'impose dans des structures de toutes tailles comme un mode d'**organisation** particulièrement efficace.

La gestion de projets informatiques ou **conduite de projet** est une démarche visant à structurer, assurer et optimiser le bon déroulement d'un projet. Géré et animé un projet par le chef de projet, c'est d'abord savoir en négocier l'objectif mais aussi prévoir. Pour cela il faut savoir mettre en œuvre les outils de l'analyse fonctionnelle, de planification (**WBS, PBS, OBS, CBS, matrice RACI, Pert, Gantt...**), gérer un budget, maîtriser les risques, motiver et animer une équipe-projet tout cela en conciliant les intérêts du **maître d'ouvrage** et **parties prenantes**. Les **livrables** doivent être clairement définis pour un résultat conforme à des normes de qualité pour le moindre **coût** dans le meilleur **délai** possible.

La gestion de projet devient un domaine de connaissances nouveau. Elle est devenue très complexe voire exigeante, comparativement il y a plus de deux décennies. Dans une large mesure, la réussite des projets d'aujourd'hui nécessite non seulement un développement accéléré de techniques et outils, mais également d'habiletés de gestion et de communication, afin de maîtriser cette nouvelle discipline. D'où le défi actuel de la profession de gestion de projet.

### 1. Problématique de la gestion de projet

L'objectif de ce chapitre est de fournir une introduction sur la gestion de projets informatiques à savoir :

- Définition du terme **Projet**
- Définition de ce qu'est la gestion de projet
- Gestion de projet système d'information
- Rôles et responsabilités des principaux acteurs impliqués dans le développement d'un projet

#### 1.1. Introduction

Les premières réflexions sur la **conduite de projets** (grands projets engagés dans les différents domaines industriels: aéronautiques, travaux publics, armement) datent des années 50.

La conduite d'un projet correspond à une mise en œuvre d'une organisation méthodologique pour faire en sorte que l'ouvrage réalisé par le **maître d'œuvre** réponde aux attentes du **maître d'ouvrage** dans les contraintes de **délai, coût et qualité**.

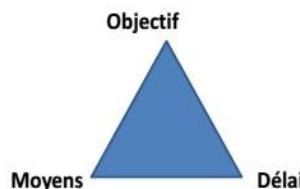
L'objectif est de développer des techniques et des méthodes pour augmenter la maîtrise des travaux et la coordination des différents corps métiers. La formalisation mathématique des problèmes de gestion pour prendre des décisions optimales.

Certaines offres variées de prestation comprenant la planification et la surveillance des délais et des coûts, la gestion de la qualité du projet, etc.... sont apparues.



## 1.2. Définition d'un projet

- Un projet est l'image plus ou moins précise d'un futur que l'on pense atteindre.
- Un projet est défini comme un ensemble d'activités à effectuer pour atteindre un but défini de façon spécifique.
- On le représente sous la forme d'un triangle:



- Donc, un projet consiste à vouloir réaliser une idée ayant un caractère **nouveau**.
- Cette réalisation est **unique**, éphémère et il faut un certain temps pour la réaliser.
- Comment réaliser l'idée en projet ?
  - A. Transformer l'idée en objectifs
    - Techniques : ce qu'on veut faire
    - De délai: en combien de temps
    - De coût: avec quel budget.
  - B. Définir les moyens nécessaires
  - C. Prévoir l'organisation et la gestion du projet.
- Qu'est ce qu'un projet
- Le terme « Projet » fait référence à une activité unique et contrainte, réalisée pour atteindre un objectif. L'objectif est toujours SMART (**Spécifique Mesurable Atteignable Réaliste Temporel**)
- Définitions normalisées :

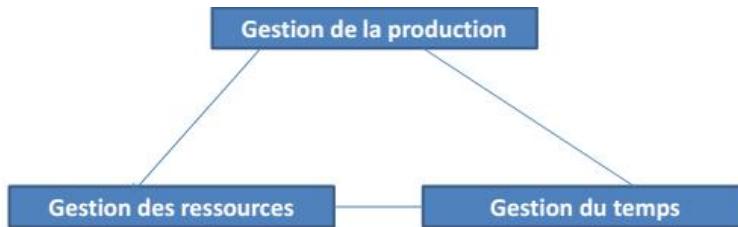
Selon ISO 10006:2003 Un projet est un processus unique, qui consiste en un ensemble d'activités coordonnées et maîtrisées comportant des dates de début et de fin, entrepris dans le but d'atteindre un objectif conforme à des exigences spécifiques telles que les contraintes de délais, de coûts et de ressources.

Le référentiel du PMI, appelé Guide du PMBOK (Project Management Body of Knowledge) donne la définition suivante: Un projet est une entreprise temporaire décidée pour obtenir un produit ou un service unique.

AFITEP et AFNOR définissent un projet comme 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.

## 1.3. Gestion d'un projet

- La gestion d'un projet a pour but de mener un projet à son terme en organisant et en surveillant son déroulement.
- Les trois aspects représentés par le triangle Projet doivent être mis sous contrôle.
- Chacun fait l'objet d'une gestion spécifique et qui prend en compte l'existence des deux autres.
- **Enjeux : Améliorer la qualité, diminuer les coûts et maîtriser les délais.**
- D'où le triangle gestion de projet suivant:



Le délai donne lieu à la gestion du temps dont le rôle est définir le parcours et de le jalonnner, d'établir des calendriers et de maîtriser la consommation de l'enveloppe temps. Les moyens affectés constituent le budget du projet qui est transformé en travail, locaux, matériel, temps machine, déplacement, etc. Cette transformation nécessite une gestion des ressources portant sur les ressources humaines et les moyens matériels.

L'objectif du projet doit à son terme être concrétisé par une ou plusieurs fournitures. Ce sommet donne naissance à la gestion de la production, qui a pour but de suivre et diriger l'avancement vers l'objectif tout au long du projet.

L'activité de gestion de projet peut être décomposée en trois activités principales autour de la production proprement dite:

#### **Analyser:**

- ✓ Déterminer le chemin que l'on va emprunter pour avancer vers l'objectif.
- ✓ Étudier les caractéristiques du projet, son contexte, les risques qui les menacent et l'état de son avancement. D'où un découpage du projet en activités à entreprendre et à une estimation de l'effort est nécessaire.

#### **Organiser:**

- ✓ Repérer les contraintes d'enchaînement entre les tâches afin de les ordonner (établir un calendrier).
- ✓ L'organisation recouvre la constitution d'une équipe et la prise en compte des relations avec tous les partenaires.

#### **Piloter:**

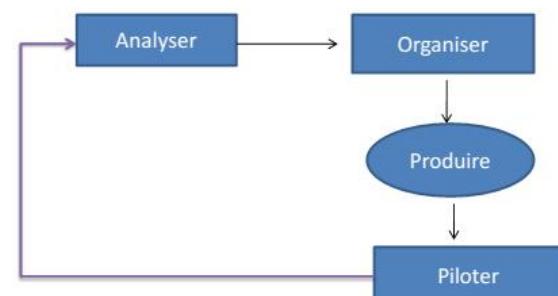
- ✓ Suivre l'avancement du projet, en quantité et en qualité ainsi que l'analyse et le traitement de l'écart avec ce qui est prévu, les orientations et les décisions à prendre ou à faire prendre.
- ✓ Inclure la gestion de l'équipe et la gestion des conflits.

#### **La boucle de rétroaction**

Un projet est managé du début jusqu'à la fin ce qui explique la boucle de rétroaction (Voir la figure ci-dessous)

#### **Définition normalisée de la gestion de projet**

Le référentiel de l'IMPA donne la définition de l'ensemble des activités de gestion de projet:





La gestion de projet consiste à planifier, organiser, suivre et maîtriser tous les aspects du projet ainsi que la motivation de tous ceux qui sont impliqués dans le projet, de façon à atteindre les objectifs de façon sûre et dans les critères définis de coûts, délais et performance.

La gestion est une responsabilité du chef de projet et se compose de quatre activités, pouvant correspondre à une fonction :

- Direction de projet
- Gestion de projet
- Maîtrise
- Pilotage

**La direction de projet** est exercée par le chef de projet et a pour mission de :

- Fixer les objectifs, la stratégie et les moyens
- Coordonner les actions successives.
- Maîtriser, modifier l'itinéraire et l'horaire si un des objectifs évolue.
- Optimiser la répartition des ressources en vue d'arriver à une solution optimale.

**La gestion de projet** (proprement dite) est effectuée afin d'accomplir la direction du projet (analyser les risques, estimer la charge, organiser le travail, le planifier et le suivre).

**L'activité de maîtrise** correspond à des tâches qui ont été extraites de la gestion de projet et qui sont centrées sur une préoccupation (la maîtrise de la qualité, par exemple)

**L'activité de pilotage** est une activité périodique d'orientation du projet faite par le comité de pilotage (si elle est isolée de la fonction de direction de projet).

Enfin, la gestion d'un projet est un processus difficile à maîtriser à cause des facteurs (de risque) suivants:

- coûts et délais à respecter
- technologies à maîtriser
- ressources humaines à gérer

## 1.4. La gestion d'un projet système d'information

### 1.4.1. Définition d'un système d'information

Un système d'information est un ensemble organisé de ressources: matériel, logiciel; personnel, données, procédures... permettant d'acquérir, de traiter, stocker, communiquer des informations (sous formes données, textes, images, sons..) dans des organisations.

### 1.4.2. Caractéristiques d'un système d'information

Le triplet (objectif, délai, moyens) dans le domaine des systèmes d'information présente trois caractéristiques:

- Q1.Il y a interaction entre l'objectif d'une part et les moyens/délais d'autre part.
- Q2.L'objectif d'un projet n'est parfaitement défini qu'à l'achèvement du projet.
- Q3.Le développement d'un système d'information se déroule dans une organisation.



#### 1.4.3. Objectifs des projets systèmes d'information

- **Productivité administrative:** possible grâce à l'automatisation d'une partie des tâches qui contribue à la diminution de la main d'œuvre.
- **Aide au management :** en bâtiissant une mémoire de l'organisation et de ce qui l'entoure à partir de laquelle on pourra construire des tableaux de bord, faire des analyses, etc.
- **Efficacité opérationnelle :** le meilleur fonctionnement est obtenu par l'usage créatif des technologies de l'information et de la communication.
- **Evolutivité :** un système flexible peut être modifié en cas d'évolution des contraintes et/ou de stratégies.
- **Utilisation d'une nouvelle technologie :** pour obtenir un «effet vitrine» vis-à-vis de l'extérieur. Un délai court est un élément essentiel de la réussite du

#### 1.4.4. Rôles et responsabilités des principaux acteurs impliqués dans le développement d'un projet

**La maîtrise d'ouvrage ou le maître d'ouvrage** est le donneur d'ordre au profit duquel l'application est conçue. Rôle du maître d'ouvrage :

- Décrire les besoins et définir le cahier des charges Etablir le financement et le planning général des projets
- Fournir les spécifications fonctionnelles générales et valider la recette fonctionnelle
- Assurer la responsabilité de pilotage du projet dans ses grandes lignes Adapter le périmètre fonctionnel en cas de retard afin de respecter la date de la livraison

**La maîtrise d'œuvre ou le maître d'œuvre** répond au programme fonctionnel déterminé par la maîtrise d'ouvrage en proposant une solution qui permette la réalisation de ce programme tout en respectant les contraintes préétablies (moyens, budget, planning, ...)

Rôle du maître d'œuvre :

- Conseiller la maîtrise d'ouvrage
- Participer à la conception de l'application
- Garantir la bonne réalisation technique de la solution proposée
- Vérifier la qualité de la réalisation (recette)

### 1.5. Conclusion

Un projet regroupe de nombreuses activités. Un projet doit concilier :

- Les objectifs fonctionnels
- Les spécifications (Aspects techniques)
- Les contraintes temporelles
- Les contraintes budgétaires
- Les contraintes matérielles (Ressources allouées).
- Les principaux acteurs impliqués dans le développement d'un projet sont le maître d'œuvre et le maître d'ouvrage dont chacun un rôle spécifique.



## 2. Découpage d'un projet et les modèles de développement.

L'objectif de ce chapitre est d'introduire la notion de découpage d'un projet logiciel et par conséquent de processus logiciel. Ce chapitre permet de :

- Q4.comprendre le principe de découpage d'un projet logiciel
- Q5.Expliciter les différents modèles de processus
- Q6.Expliciter le processus RUP, un processus moderne et générique

### 2.1. Introduction

Un processus logiciel est un ensemble d'activités qui conduit à la réalisation du produit logiciel. Les processus logiciel sont complexes et nécessitent certaines décisions prises par les différents intervenants.

Bien qu'il existe plusieurs modèles de processus logiciel, certaines activités fondamentales sont communes à tous les processus logiciels telles que la spécification du logiciel, la conception et l'implémentation du logiciel, la validation du logiciel et enfin l'évolution du logiciel.

Les principes de découpage La tâche du chef de projet est de découper le projet afin de pouvoir répartir dans le temps la production et les ressources nécessaires.

Découper un projet consiste à identifier des sous -ensembles quasi autonomes, ayant les caractéristiques suivantes:

- Chaque sous ensemble donne lieu à un résultat bien identifié
- La charge propre à chacun peut être évaluée.
- Les contraintes d'enchaînement entre les sous ensembles sont repérables: certains sous ensembles peuvent être réalisés parallèlement, d'autres sont liés par des contraintes d'antériorité.
- Le découpage est fait à des mailles différentes, un sous ensemble étant souvent à son tour décomposé.

### 2.2. Critères de découpages:

On utilise deux grands critères pour découper un projet: l'un est temporel, l'autre est structurel.

Le critère temporel est utilisé dans la plupart des projets répartir le travail dans le temps. Il se base sur les modèles de développement (process models) ou modèle de cycle de vie (Modèle de la cascade, modèle en V, modèle de la spirale. Le critère temporel est utilisé dans la plupart des projets.

Un projet se compose de phases. Chaque phase comprend un certain nombre d'activités.

Une activité est définie par une ou plusieurs tâches à effectuer. A chaque élément de décomposition, on attache un résultat à atteindre appelé livrable.

Les principaux livrables à fournir sont: le cahier de charges, le plan de projet, les spécifications détaillées, l'architecture fonctionnelle et technique, planification, le plan assurance qualité, la gestion des risques, le plan de test, le logiciel (source, exécutable), manuel d'utilisateur, etc.



L'ensemble ordonné des phases d'un projet s'appelle le cycle de vie du projet.

Le critère structurel permet d'organiser le travail en se basant sur la structure du produit final: la décomposition fait apparaître les différents modules qu'il faut obtenir. L'utilisation de ce critère nécessite une visibilité suffisante sur le résultat à produire.

En plus du découpage temporel et surtout si le projet est de taille importante, on recourt au découpage structurel pour les raisons suivantes:

- Q7. Maitriser le projet: les sous ensembles sont de taille plus réduite et plus facile à maîtriser.
- Q8. Répartir les responsabilités: l'autonomie des modules permet de confier différents responsables pour différents modules.
- Q9. Réduire les délais planifiés : certains modules indépendants sont développés en parallèle ce qui permet d'avancer la date théorique d'achèvement du projet.
- Q10. Avoir un développement incrémental: pour différentes raisons, on choisit de développer un système d'information en versions successives dont chaque version comporte un nombre croissant de modules par rapport à la précédente.

### **2.3. Le cycle de vie standard**

Le cycle standard se compose des phases suivantes:

- Etude de faisabilité comprend les travaux d'analyse, des travaux de recherche, des études sur terrain où il s'agit de vérifier si le projet est techniquement réalisable.
- Définition des solutions: donne une représentation précise de l'objectif à atteindre. Les solutions possibles sont étudiées de façon détaillée. À terme de cette phase, une solution est choisie et l'on dispose de spécifications exactes.
- Conception détaillée: sert à préparer les contrats de réalisation. Ces contrats contiennent le cahier des charges pour les sous-traitants.
- Réalisation: est l'exécution des contrats par les sous-traitants, conformément aux cahiers des charges. Cette phase se termine par une procédure d'acceptation officielle.

#### **Problèmes posés par le découpage standard**

- La notion de cahier des charges est déclinée à plusieurs moments. La plupart des phases du cycle de développement peuvent conduire à un cahier des charges qui oriente le travail de l'étape ultérieure.
- Le découpage temporel standard suppose que le client possède une description complète de ce qu'il attend. Or la détermination des besoins et des solutions adéquates est un problème majeur.
- L'élaboration d'un cahier des charges est un travail coûteux. L'écriture des spécifications souffre de l'absence de composants réutilisables.

### **2.4. Le découpage classique**

Les méthodes de développement des systèmes d'information ont proposé un découpage temporel de référence (cycle de vie classique) qui correspond au vocabulaire de la méthode Merise.

SD	EP	ED	ET	REAL	MEO	QUALIF
----	----	----	----	------	-----	--------



- **Le schéma directeur (SD)** : sert à définir les scénarios d'évolution du patrimoine informatique sous l'un ou l'autre de ces trois angles :
  - Evolution de l'architecture technique (matériels, réseaux).
  - Evolution de l'architecture applicative (données commune, identification des domaines, etc.)
  - Evolution des fonctions informatiques (méthodes, normes, outils)
- **L'étude préalable(EP)**: sert à repenser une application vieillissante sur un domaine bien identifié ou pour répondre à un nouveau besoin. L'objectif est double: des choix structuraux pour la future application à savoir évaluer l'adéquation de la solution aux objectifs, évaluer l'investissement (budget, temps), ajuster la solution à l'enveloppe si cela est nécessaire.

Fournir une base de référence pour la suite du projet. Le rapport de l'étude préalable est considéré comme un cahier des charges pour l'étude détaillée.

- **L'étude détaillée (ED)** : sert à concevoir et à décrire de façon exhaustive la solution qui sera ensuite complétée par l'étude technique. Le résultat comprend toute la vision externe du système.
- **L'étude technique (ET)**: concerne uniquement les informaticiens et consiste à optimiser les structures physiques de données et de construire les traitements en essayant de préparer la réutilisation du code.
- **La réalisation (REAL)** : l'objectif est de produire un logiciel testé. Elle se termine par une procédure d'acceptation officielle appelée recette.
- **La mise en œuvre(MEO)** : consiste à préparer le démarrage effectif de la nouvelle application.
- **La qualification (QUALIF)**: l'objectif est de réaliser des tests dans l'environnement opérationnel et de tirer un bilan du système d'information installé

## 2.5. Les modèles de développement

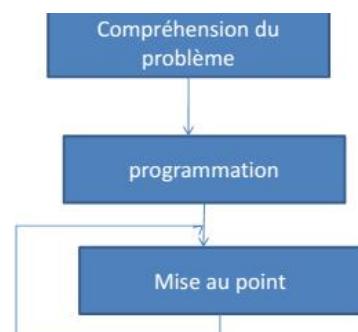
- Il n'existe pas une démarche unique mais on peut construire le découpage temporel en fonction des caractéristiques de l'entreprise et du projet. Les découpages temporels génériques, appelés les modèles de développement (process models) ou modèles de cycle de vie sont les suivants:

- Le modèle du code-and-fix.
- Le modèle de la transformation automatique.
- Le modèle de la cascade.
- Le modèle en V.
- Le modèle en W.
- Le modèle de développement évolutif.
- Le modèle de la spirale

### 2.5.1. Le modèle du code-and-fix

Après une étape brève de compréhension de l'objectif, l'application est développée.

Plusieurs cycles de mise au point, permettent d'atteindre le résultat visé, une collaboration avec l'utilisateur du futur système





### 2.5.2. Le modèle de la transformation automatique

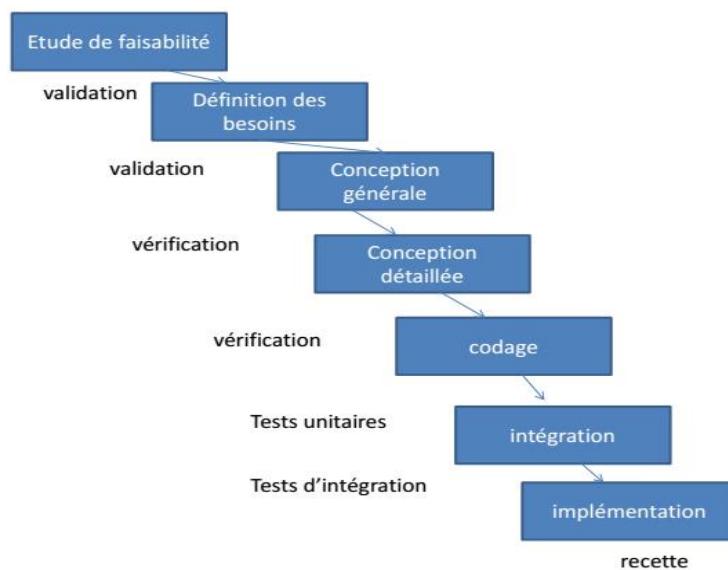
Il est basé sur la possibilité de transformer automatiquement des spécifications en programmes. Cela suppose que les spécifications sont complètement validées. Une succession de cycles de spécification/validation s'achève par la génération de code.



### 2.5.3. Le modèle de la cascade

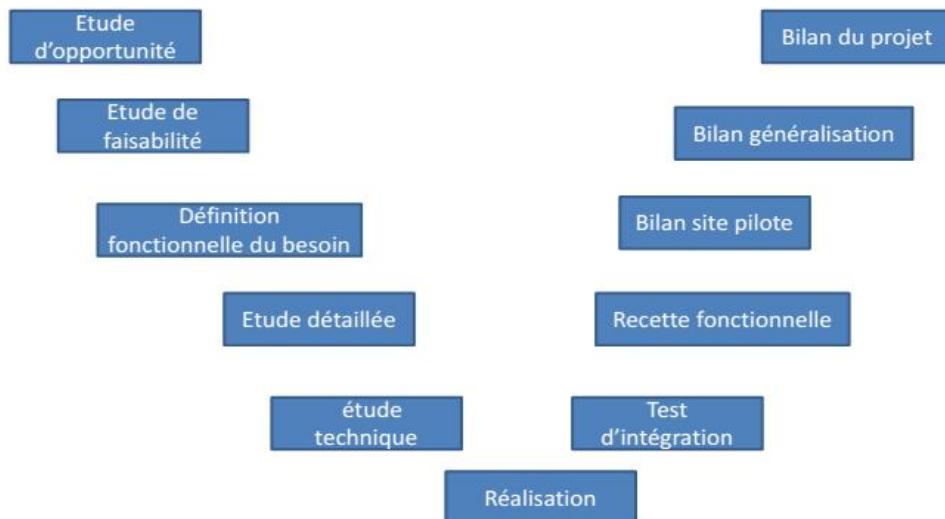
L'objectif de ce modèle est de jaloner rigoureusement le développement et de définir de manière précise les rôles respectifs du fournisseur (qui produit le livrable) et du client (qui accepte ou refuse le résultat). C'est une succession de phases qui correspond à une approche descendante.

Chaque phase donne lieu à une validation officielle : on ne passe à la suivante que si le résultat du contrôle est satisfaisant. Sinon, on modifie le livrable pour qu'il devient acceptable.



### 2.5.4. Modèle en V

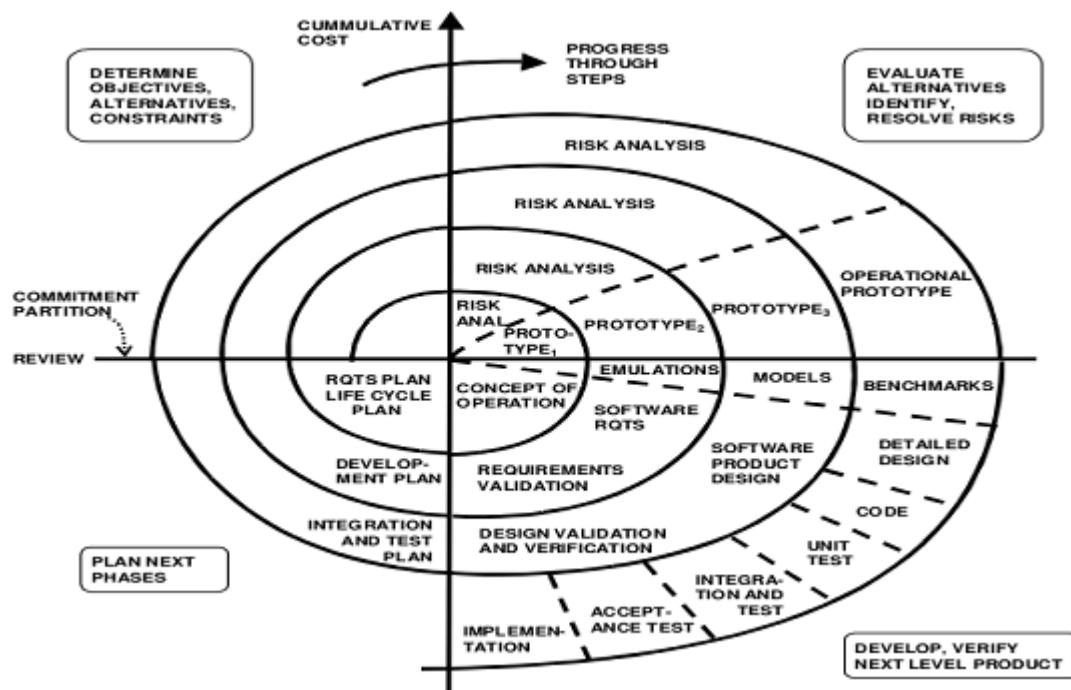
C'est une amélioration du modèle de la cascade. Chaque phase de la première branche du V va être explicitée par des critères d'appréciation et d'acceptation du système aux étapes correspondantes de la deuxième branche du V.



### 2.5.5. Le modèle de la spirale

Modèle en spirale d'après [Boehm 88]

- ✓ détermination des objectifs du cycle, des alternatives pour les atteindre et des contraintes ; à partir des résultats des cycles précédents, ou de l'analyse préliminaire des besoins;
- ✓ analyse des risques, évaluation des alternatives à partir de maquettage et/ou prototypage;
- ✓ développement et vérification de la solution retenue, un modèle « classique » (cascode ou en V) peut être utilisé ici ;
- ✓ revue des résultats et vérification du cycle suivant



C'est un modèle qui repose sur le fait qu'il y a une relation contractuelle entre le fournisseur et le client. Plusieurs cycles sont effectués. Chaque cycle donne lieu à une contractualisation s'appuyant sur les besoins exprimés lors du cycle précédent. Chaque cycle comporte les étapes suivantes :

- Analyse du risque



- Développement d'un prototype
- Simulation et essais du prototype
- Détermination des besoins à partir des résultats des essais
- Validation des besoins par un comité de pilotage
- Planification du cycle suivant

Le dernier cycle permet de développer la version finale et implémenter le logiciel.

Proposé par B. Boehm en 1988, ce modèle de cycle de vie tient compte la possibilité de réévaluer les risques en cours de développement, il emprunte au prototypage incrémental mais lui adjoint une dimension relevant de la prise de décision managériale et non purement technique couvre l'ensemble du cycle de développement d'un produit.. Il met l'accent sur l'activité d'analyse des risques. Chaque cycle de la spirale s déroule en quatre phases :

- Identifier les risques, leur affecter une priorité,
- développer une série de prototypes pour identifier les risques en commençant par le plus grand risque,
- utiliser un modèle en V ou en cascade pour implémenter chaque cycle,
- si un cycle concernant un risque a été achevé avec succès,
- évaluer le résultat du cycle et planifier le cycle suivant,
- si un risque n'a pu être résolu, terminer le projet immédiatement

## 2.6. Analyse des risques

La mise en œuvre demande des compétences managériales et devrait être limitée aux projets innovants à cause de l'importance que ce modèle accorde à l'analyse des risques. Citons, par exemple

- risques humains:
  - défaillance du personnel ; surestimation des compétences
  - travailleur solitaire, héroïsme, manque de motivation
  -
- risques processus :
  - pas de gestion de projet
  - calendrier et budget irréalistes ;
  - calendrier abandonné sous la pression des clients
  - composants externes manquants ;
  - tâches externes défaillantes ;
  - insuffisance de données
  - validité des besoins ;
  - de fonctions inappropriées
  - développement d'interfaces utilisateurs inappropriées
  -
- risques technologiques
  - miracle, "plaqué or";
  - changement de technologie en cours de route
  - de performance
  - démesurées par rapport à la technologie
  - incompréhension des fondements de la technologie



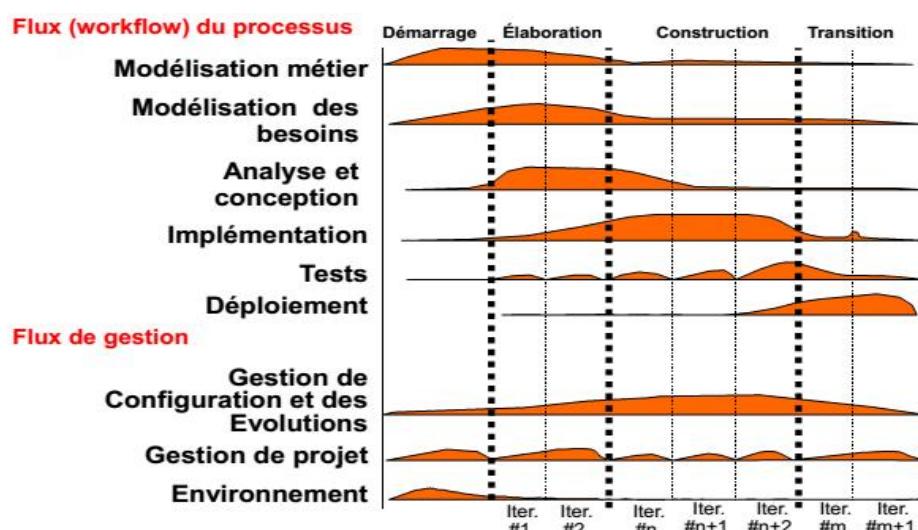
## 2.7. Les modèles de cycle de vie spécifiques

Le modèle **RUP** (Rational Unified Process) est représentatif d'une approche combinant plusieurs modèles. Sa structure fait l'objet d'un assez large accord, notamment parmi les praticiens.

Il peut être lu de la façon suivante :

- Le cycle est constitué de quatre phases principales: étude préalable (opportunité), conception de la solution détaillée (élaboration), développement de la solution (construction) et mise en œuvre (transition).
- Il existe six types de tâches qui se retrouvent à des degrés différents dans chacune des phases. exemple, l'étude des besoins peut apparaître jusqu'à la fin du projet mais la plus grande partie est effectuée dans les deux premières phases.
- Certaines phases peuvent être menées de façon cyclique. L'élaboration se fait en deux cycles, conduisant à la production des spécifications externes (vision utilisateur) et des spécifications techniques(vision développeur). La construction s'effectue en spirale.
- NB: cette section est à compléter par des lectures personnelles.

## 2.8. Les 2 Visions Rassemblées: Le Modèle Itératif



## 2.9. Conclusion

Les modèles de processus logiciel sont des représentations abstraites des processus logiciel. Ils visent à ordonner et à structurer le développement du logiciel. Tous les processus logiciel incluent certaines activités communes.

Les modèles de processus itératifs présentent le processus logiciel comme étant un cycle d'activités. Le choix d'un modèle de développement s'appuie sur l'analyse des caractéristiques et surtout sur l'analyse des risques.

Il existe aussi d'autres modèles tels que **Extreme Programming**, développement orienté service, etc. qui ne sont pas traités dans ce chapitre.



### 3. Gestion de projet.

L'objectif de ce chapitre est de fournir une vue générale sur la gestion de projet. Cela se traduit par:

- Déterminer les principales tâches du chef de projet
- Comprendre le besoin d'une planification pour tout projet logiciel
- définir les deux représentations graphiques qui correspondent à la programmation d'un projet
- Introduire la notion de gestion des risques et certains types de risques qui peuvent affecter des projets logiciels

#### 3.1. Introduction

La gestion d'un projet logiciel est une partie essentielle du génie logiciel. Une mauvaise gestion conduit à un échec de réalisation du logiciel: non satisfaction des besoins des usagers, dépassement des coûts et des délais prévus.

Le chef du projet est responsable de la planification et de la programmation du développement du projet

#### 3.2. Activités de gestion

Les activités de gestion suivies par la plupart des chefs de projets sont:

- a) Ecriture de la proposition du projet, la proposition décrit les objectifs du projet et comment y parvenir (les grandes lignes de sa réalisation). Une proposition doit aussi contenir une évaluation des risques et des coûts. Elle peut servir aussi d'argumentaire pour justifier la mise en route du projet. De plus, elle nécessite de l'expérience et de la compréhension du domaine d'activité.
- b) Planification et programmation du projet: consiste à identifier les activités, les jalons et les livrables produits par le projet. Il s'agit d'un travail d'ordonnancement qui nécessite des connaissances très précises du domaine, des équipes de développement, etc.
- c) Evaluation des coûts du projet consiste à estimer les ressources nécessaires pour accomplir le plan du projet.
- d) Contrôle et révision du projet c'est une activité continue qui consiste à observer la progression du projet et à le comparer avec le progrès planifié. Cela peut avoir lieu à l'aide d'un contrôle informelle (discussion entre équipe) ou durant les revues formelles qui sont effectuées.
- e) Sélection et évaluation du personnel: consiste à choisir un personnel compétent.
- f) Ecriture du rapport et des présentations: consiste à établir un rapport sur le projet pour le client et pour l'organisation contractuelle. C'est un document concis, cohérent qui est présenté lors des contrôles de progression. Par conséquent, le chef de projet doit pouvoir communiquer une vue synthétique du projet à différents publics (autres chefs de projet, clients, responsables, etc.).

#### 3.3. Planification du projet

Une gestion effective d'un projet dépend de la planification du progrès du projet. Le plan initialement établi doit être le meilleur possible et il doit évoluer au fur et à mesure que le projet avance.



Au début, le plan envisagé doit établir les estimations concernant certaines contraintes affectant le projet (date de livraison, équipe disponible, budget). De plus, certains autres paramètres sont aussi à prendre en considération (la taille, la structure et la répartition des fonctions) pour enfin définir les **jalons** et les **livrables**.

Dans la majorité des cas, le plan comprend les sections suivantes:

- **Introduction:** description des objectifs du projet et des contraintes (délai, budget)
- **Organisation du projet:** description de l'organisation de l'équipe, le personnel impliqué et le rôle de chacun joué dans l'équipe.
- **Analyse des risques :** description des risques possibles du projet, leur probabilité d'apparition ainsi que les stratégies proposées pour les réduire.
- **Besoins des ressources en matériel et logiciel:** description du matériel et du logiciel de support pour le développement du projet.
- **Découpage du projet:** découpage du projet en activités en identifiant les jalons et les livrables.
- **Programmation du projet:** description de la dépendance entre les activités, le temps estimé pour déterminer chaque jalon et l'affectation du personnel dans chaque activité.
- **Mécanisme de contrôle et les rapports:** définition des rapports de gestion établis, quand ils sont établis, et le mécanisme de contrôle utilisé.

### 3.3.1. Les jalons et les livrables

Le logiciel est un produit **intangible** ce qui rend impossible d'estimer l'avancée du projet, le coût du projet ainsi que la mise à jour de la programmation du projet d'où la nécessité d'établir une série de jalons.

- **Un jalon (en anglais, milestones)** est une marque de fin reconnue d'une activité du processus logiciel. Pour chaque jalon, il faut établir un sortie formelle (rapport) qui n'est pas nécessairement un document important Il peut correspondre à un court rapport synthétisant ce qui a été fait.
- **Un livrable est un résultat du projet qui est délivré au client.** Il est livré à la fin d'une phase donnée du projet telle que la phase de spécification ou de conception. Il est mesurable, tangible et/ou vérifiable.



Par conséquent, les livrables sont des jalons mais les jalons ne sont pas nécessairement des livrables.



### 3.4. La programmation du projet

Elle représente une tâche difficile du chef de projet où le délai et les ressources nécessaires sont estimés afin de compléter les activités prévues et de les organiser en une séquence cohérente. Le programme sera défini et représenté sous la forme d'un diagramme.

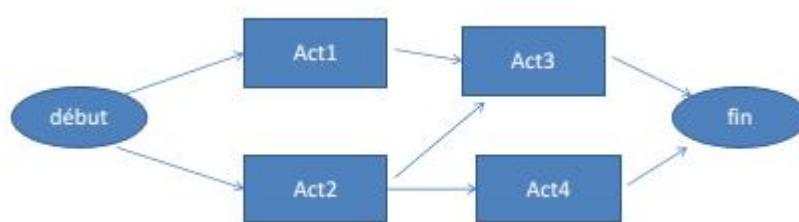
Deux techniques complémentaires sont utilisées: La technique des graphes et la technique de Gantt.

#### 3.4.1. La technique des graphes

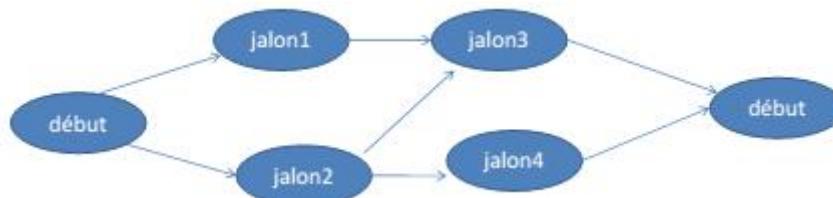
Deux méthodes existent: la méthode des antécédents et la méthode du diagramme fléché.

**1. Graphe des antécédents :** L'établissement de ce graphe correspond à la méthode du Chemin critique ou encore CPM (Critical Path Method). Ce graphe ou réseau est aussi appelé PDM (Precedence Diagramming Method).

Chaque activité y est représentée par une boîte. Les activités sont liées entre elles par des liaisons de dépendances représentées par des flèches. C'est une représentation synthétique des relations logiques entre activités, construit de gauche à droite pour représenter la chronologie d'un projet.



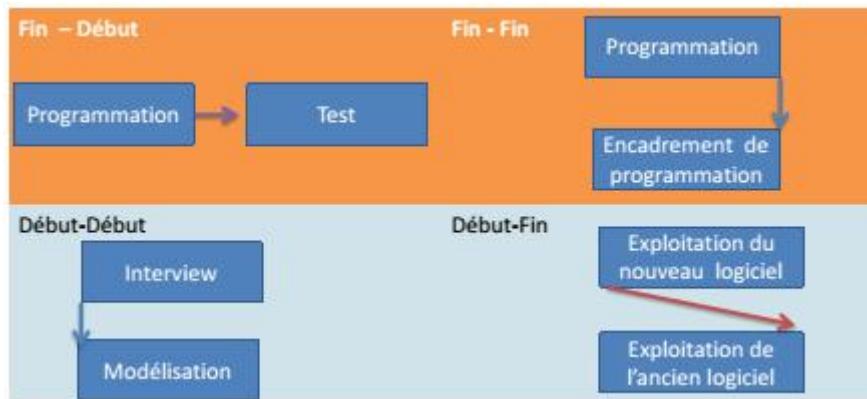
**2. Diagramme fléché:** les activités figurent sur les flèches et les ronds représentent les jalons



**Remarque:** Dans la majorité des progiciels de planification, le formalisme du graphe des antécédents est retenu parmi lesquels on cite PMW (Project Management Workbench) et Microsoft Project.

L'avantage du graphe des antécédents est qu'il permet une visualisation claire de la logique des dépendances. Il donne aussi la possibilité de relations avec des délais (écart/recouvrement).

**Types de liens :** Les liens entre les tâches représentent des contraintes provenant de la nature des tâches elles mêmes. Les types de liens qui existent sont les suivants:



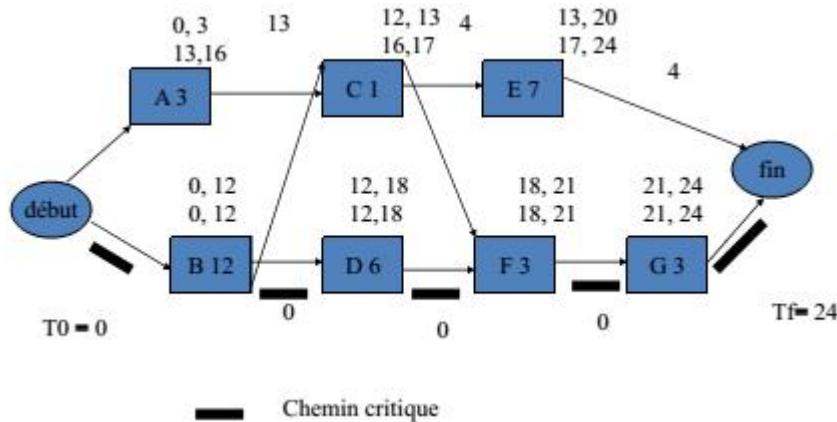
### 3.4.2. Méthode du chemin critique

L'analyse du graphe permet de mettre en évidence des chemins qui comportent des tâches critiques dits chemins critiques. Le chemin critique correspond à la séquence de tâches qui détermine la durée totale du projet. Ce chemin est continu depuis le début jusqu'à la fin du projet. Tout retard affectant une tâche du chemin critique est intégralement répercute sur la durée du projet et donc sa date de fin. La tâche critique est une tâche du chemin critique.

Pour cela, on calcule les paramètres clés attachés à chaque graphe:

- **Dates au plus tôt:** Début au plus tôt et fin au plus tôt.
- **Dates au plus tard:** Début au plus tard et fin au plus tard.
- **Marges :** Marge totale et marge libre

Le réseau Pert : le chemin critique, exemple :



### 3.5. Gestion des risques

Un risque correspond à la probabilité qu'un événement indésirable ait lieu. Une des principales tâches du chef de projet : anticiper les risques qui peuvent affecter la programmation du projet ou la qualité du logiciel et par conséquent déterminer les actions à entreprendre a fin de les éviter.

Différents types de risque peuvent être identifiés, parmi lesquels on cite les catégories des risques Suivants :



- **Risques liés au projet** : risques qui affectent la programmation du projet ou les ressources qui lui sont alloués.
- **Risques du produit** : risques qui affectent la qualité ou la performance du logiciel.
- **Risques métiers ou commerciaux** : risques qui affectent l'organisation développant ou procurant le logiciel.

Risque	Type	Description
Rotation du personnel	Projet	personnel expérimenté abandonne le projet avant sa fin
Changement du besoin	Produit	grand nombre de changement des besoins est effectué
Compétition du produit	Métier	produit compétitif est disponible sur le marché avant que le système soit complet

Le processus de gestion des risques comprend les étapes suivantes :

- A. **Identification des risques** : L'identification du risque consiste à découvrir les risques possibles liés au projet. Elle se base sur une approche intuitive ou simplement sur les expériences.
- B. **Appréciation des risques** : Elle comporte deux tâches distinctes : la première consiste à estimer la probabilité que l'événement se produise et la seconde consiste à évaluer l'impact, le coût des conséquences d'un tel événement.
- C. **Calculs des risques** : L'attribution d'une valeur chiffrée à un risque consiste à multiplier sa probabilité qu'il se produise par le coût estimé de ses conséquences.
- D. **Atténuation des risques** : C'est une stratégie proactive pour tenter de diminuer la probabilité ou d'en réduire l'impact. La démarche couramment utilisée consiste à traiter les points potentiellement problématiques le plus tôt possible.
- E. **Plan de gestion des risques** : Il doit contenir, pour chacun des risques, un identifiant, une description, une estimation de sa probabilité et de son impact, une liste des stratégies d'atténuation envisageables, des plans de secours, des événements déclencheurs qui détermineront l'activation des plans de secours et enfin la liste des personnes responsables.

### 3.6. Conclusion

Une bonne gestion des projets est essentielle car ceux – ci doivent être développés en respectant certaines contraintes (coût, délai par exemple).

Le plus important rôle du chef de projet est la planification, l'estimation du coût et la programmation du projet.

La programmation d'un projet implique l'établissement d'un diagramme des antécédents déterminant les activités et leurs relations ainsi qu'un diagramme de Gantt.

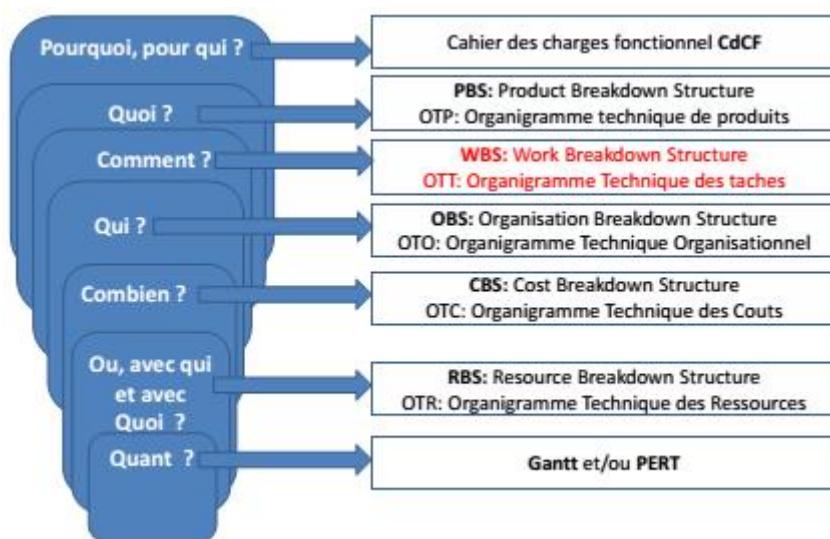
Les risques associés à un projet doivent être identifiés, leurs probabilités sont à estimer ainsi que leurs conséquences. Des plans sont à établir afin de les éviter ou de réduire leurs conséquences.



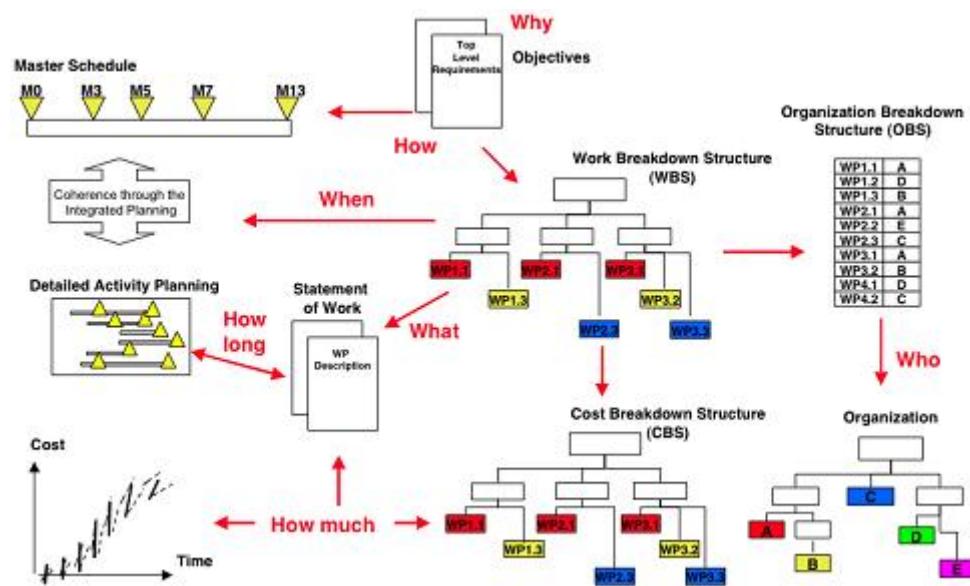
## Annexe A : Organisation d'un projet: des approches et des découpages



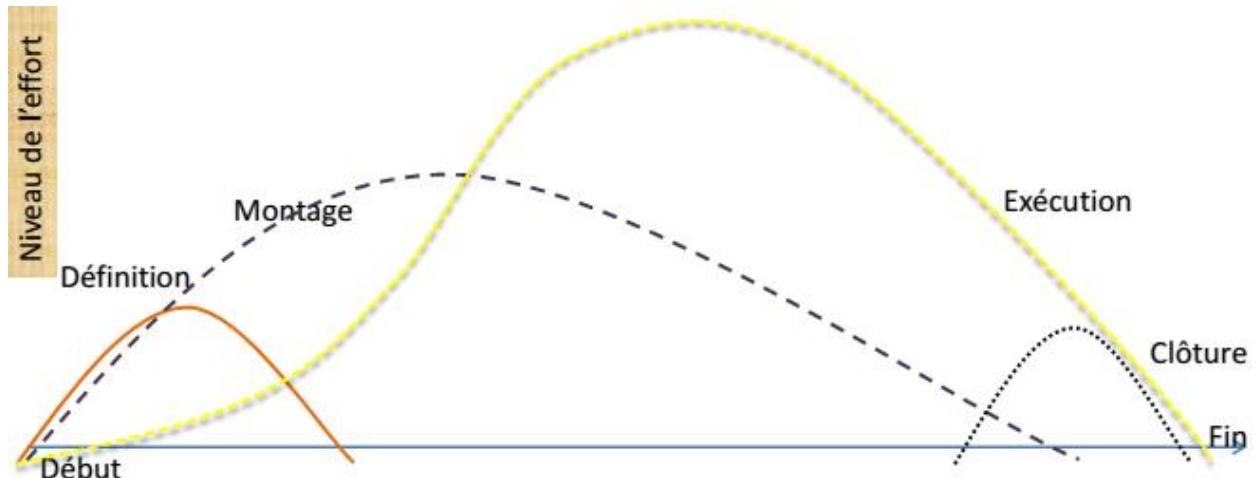
## Annexe B : Outils d'aide à la conduite de projets Principaux formalismes de décomposition structurée



### Annexe C : Synthèse des processus et outils pour la gestion de projets



### Annexe D :



Définition	Montage	Exécution	Clôture
1 – Objectifs g <sup>aux</sup>	1 – C. des charges	1 – Suivi / pilotage	1 – Livraison/transfert
2 – Fiche de Déf.	2 – Lots de travail	2 – Modifs CdC	2 – Formation MOA
3 – CR, Todo list	3 – Responsabilités	3 – Gestion des risques	3 – Post Mortem
4 – Planning		4 – Prototypage, tests	4 – Réaffectation
s – Budget / Ressources			

**Définition :** petits projet = suffisant, par contre si vous lancer dans une entreprise plus ambitieuse : en équipe, sur plusieurs mois, il faut plus que cela – c'est le but ici; Nous allons couvrir les phases de montage.



## 4. Gestion de la qualité

L'objectif de ce chapitre est d'introduire la gestion de la qualité du logiciel et les mesures du logiciel. Cela permettra de:

- ✓ Connaitre et comprendre le processus de gestion de la qualité et les activités associées: l'assurance qualité, la planification de la qualité et le contrôle de la qualité.
- ✓ Comprendre l'importance des standards dans le processus de gestion de la qualité.
- ✓ Connaitre Certaines métriques du logiciel.
- ✓ Comprendre comment les mesures peuvent aider à estimer certains attributs de qualité du logiciel

### 4.1. Introduction

La qualité du logiciel est un concept complexe et qui n'est pas directement comparable avec la qualité dans le domaine de la production. Cela est dû à:

- ✓ La spécification vise à définir les caractéristiques du produit demandé par l'utilisateur (besoins fonctionnels) alors que l'organisation de développement peut être intéressée par certains besoins non fonctionnels (maintenabilité) et qui ne figurent pas parmi ces spécifications.
- ✓ La spécification de certaines caractéristiques non fonctionnelles (maintenabilité, par exemple) est difficile.
- ✓ La spécification d'un produit logiciel n'est pas toujours (ou n'est jamais) complète. La gestion de la qualité se base sur des standards qui encapsulent certaines bonnes pratiques.

D'une manière générale, les bonnes pratiques peuvent être considérées comme des solutions (voire meilleures solutions) à un problème posé dans un contexte donné.

Elles permettent donc d'indiquer ce qui a lieu de faire et ce qui a lieu d'éviter. La gestion de la qualité formalisée est particulièrement importante pour l'équipe développant des systèmes complexes. Elle peut être structurée en trois activités:

- Assurance qualité
- Planification de la qualité
- Contrôle de la qualité.

Pour les petits systèmes, une approche informelle à la gestion de la qualité est adoptée à travers l'établissement d'une culture de qualité qui sera partagée par tous les membres de l'équipe.

Le processus de gestion de la qualité a pour but de vérifier les livrables du projet avec les standards de l'organisation. La gestion de la qualité est affectée à une équipe indépendante de l'équipe de développement.

### 4.2. Processus basé sur la qualité

La qualité du processus de développement affecte directement la qualité du produit. Mais, la relation entre la qualité du processus de développement et la qualité du produit est plus complexe. Il est difficile de mesurer les attributs de qualité du logiciel (maintenabilité) même après avoir exploité longuement le logiciel →difficulté de savoir comment les caractéristiques du processus influent sur les attributs de qualité.



### Qualité du produit et la qualité du processus

La gestion de la qualité d'un processus consiste à:

- ✓ définir les standards : comment et quand les revues sont (ou ont été) conduites.
- ✓ contrôler le processus de développement afin de s'assurer que les standards sont bien suivis.
- ✓ reporter le processus logiciel à la gestion de projet.

### 4.3. Assurance qualité et standards

L'assurance qualité est le processus de définition complète de la qualité du logiciel et comment l'organisation de développement connaît le niveau de qualité requis d'un logiciel. Le processus d'assurance qualité consiste à définir ou à sélectionner les standards qui seront appliqués aux processus de développement ou au produit logiciel.

Le processus d'assurance qualité regroupe deux standards:

- Standards du produit: applicables aux produits logiciels, inclut les standards des documents (structure des documents des requis)
- Standards de processus: définition du processus à suivre durant le développement du logiciel, inclut la définition des besoins, la description des documents générés durant ces processus.

Les standards du logiciel sont importants car ils:

- sont basées sur la connaissance des meilleures pratiques d'une organisation.
- fournissent un Framework pour implémenter un processus d'assurance qualité.
- assurent la continuité des travaux car les membres se partagent les mêmes pratiques.

#### Exemples de standards :

- A. Standards du produit :
  - forme de revues de conception
  - structure du document des requis (besoins ou exigences)
  - format d'un plan de projet
- B. Standards du processus :
  - processus des livrables d'une version
  - processus de contrôle de changement.



C. Les standards de documentation : La documentation est l'unique moyen tangible pour représenter le logiciel et le processus logiciel. Ils sont répartis en trois types:

- Standards du processus de documentation: définissent le processus suivi pour produire une documentation
- Standards des documents: définissent la structure et la présentation des documents.
- Standards d'échange de documentation: assurent que toutes les copies électroniques des documents sont compatibles

#### 4.4. Planification de la qualité

C'est le processus de développement du plan de qualité d'un projet. Le plan de qualité formule les qualités du logiciel et comment elles sont évaluées.

Exemple de structure d'un plan de qualité proposé par Humphrey:

- **Introduction du produit** : description du produit, le marché visé, les prévisions de qualité du produit.
- **Planning du produit**: dates de livraison critiques et les responsabilités ainsi que les plannings de distribution et de service du produit.
- **Description du processus**: processus de développement et de service qui seront utilisés pour le développement du produit et pour sa gestion.
- **Objectifs de qualité**: les objectifs et les plans de qualité du produit incluant l'identification et la justification des attributs critiques de qualité du produit.
- **Risques et la gestion des risques**: Les risques importants qui peuvent affecter la qualité du produit et les actions à entreprendre pour traiter ces risques.

Le plan de qualité diffère selon la taille et le type du système à développer. Il doit être le plus court possible. Le plan de qualité doit définir les attributs de qualité les plus importants d'un logiciel (maintenabilité, robustesse). Il doit aussi inclure le processus d'estimation de la qualité.

#### 4.5. Contrôle de qualité

Le contrôle de la qualité implique le suivi du processus de développement du logiciel afin de s'assurer que les procédures de qualité et les standards sont bien suivis. Deux approches complémentaires peuvent être utilisées pour vérifier la qualité des livrables.

- Les revues de qualité: le logiciel, sa documentation et le processus utilisé pour produire ce logiciel sont révisés par une équipe afin de vérifier le suivi des standards du projet et la conformité du logiciel et de sa documentation aux standards.
- L'évaluation automatisée du logiciel: le logiciel et ses documents sont traités par certains programmes et comparés aux standards mesurer certains attributs du logiciel et les comparer avec certains niveaux désirables.

Type de revue	But principal
inspections de programmes et de conception	Déetecter les erreurs au niveau de la conception ou du code
Revue de progression	fournir l'information sur le progrès du projet: revues du produit et du processus concernant le coût, la programmation



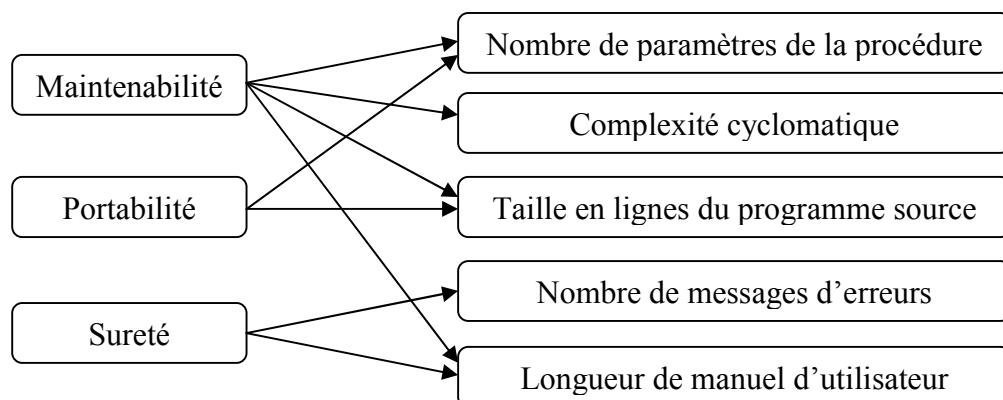
## 4.6. Mesures du logiciel et les métriques

Les revues de qualité sont couteuses et consomment du temps. Pour accélérer le processus de revues, il est nécessaire d'utiliser des outils dans le but d'évaluer la qualité du logiciel ou du processus. Les mesures du produit logiciel peuvent être utilisées pour:

- ✓ faire des prédictions sur le système: dériver une estimation concernant certains attributs du système (nombre d'erreurs dans le système)
- ✓ identifier les composants défectueux: identifier certains composants qui violent certaines normes (composants ayant une complexité importante).

Certains attributs externes ne sont pas directement mesurables (maintenabilité, compréhensibilité). Ils sont affectés par plusieurs attributs internes (taille) → il faut donc établir des relations entre attributs internes et externes ;

**Exemple:**



Cependant, la nature de la relation n'est pas bien définie. Pour que les attributs internes soient des prédicteurs utiles, il faut:

- ✓ mesurer précisément les attributs internes
- ✓ définir la relation entre ce qui a été mesuré et l'attribut externe d'intérêt
- ✓ établir la formule ou le modèle exprimant la relation déjà définie.

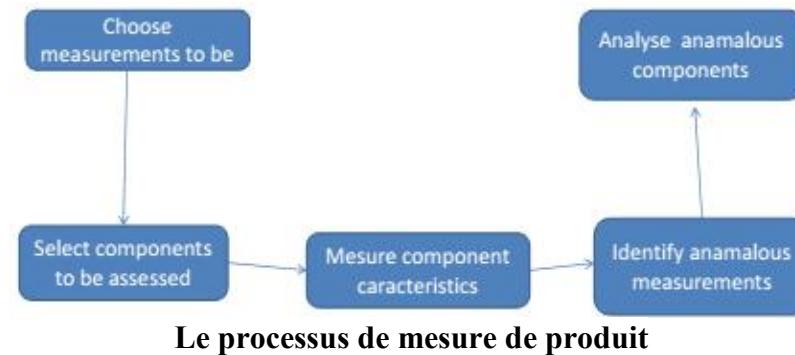
### 4.6.1. Fiabilité d'un logiciel

Elle se définit par la probabilité qu'a un logiciel de ne pas connaître de défaillances Pendant une période donné. Elle est notée  $F(n)$  où  $n$  est le nombre d'unités temporelles.

**Exemple :** Si l'unité choisie est le jour,  $F(1)$  désigne la probabilité que le programme ne connaisse pas de défaillances pendant une journée d'où la probabilité qu'un programme connaisse une défaillance pendant une durée donnée correspond à :  $D(n) = 1 - F(n)$ .

### 4.6.2. Processus de mesure

Il représente une partie importante du processus de qualité. Chaque composant du logiciel est analysé séparément et les valeurs obtenues sont comparées aux données collectées à partir des projets précédents. Les mesures anormales servent à déterminer les composants présentant des problèmes de qualité. Le processus de mesure du produit est illustré par la figure suivante:



#### 4.6.3. Métriques du logiciel

Les métriques du logiciel se divisent en deux catégories:

- A. Les métriques statiques: collectées à partir des mesures établis sur la représentation du système (conception, programme, documentation).
  - ✓ Elles permettent de calculer la complexité, la taille d'un programme, des prédicteurs pour certains attributs de qualité (maintenabilité, compréhensibilité)
- B. Les métriques dynamiques: collectées à partir des mesures effectuées sur un programme en exécution.
  - ✓ Elles servent à estimer la sûreté et l'efficacité d'un programme.
  - ✓ Exemple de métriques statiques : Métriques génériques:
    - ✓ Fan-in / Fan-out, complexité cyclomatique, la longueur du code, la longueur des identificateurs.
    - ✓ Métriques spécifiques(orientées objet): profondeur de l'arbre d'héritage, méthode fan-in / fan-out.

#### 4.7. Conclusion

- ✓ La gestion de la qualité du logiciel a pour but de réduire le nombre de défauts.
- ✓ Les activités de gestion de la qualité incluent l'assurance qualité, la planification de la qualité et le contrôle de la qualité.
- ✓ Les standards du logiciel sont importants à l'assurance qualité car ils représentent une identification des bonnes pratiques.
- ✓ Le processus de contrôle de la qualité consiste à vérifier la conformité de la qualité du produit et du processus avec les standards.
- ✓ Les revues représentent les techniques les plus utilisées d'estimation de la qualité.
- ✓ Les métriques de qualité du produit permettent de déterminer les composants anormaux. Ces composants doivent être analysés en détail.
- ✓ Il n'y a pas de métriques du logiciel applicables à tout logiciel. Les organisations doivent sélectionner les métriques et analyser les mesures sur la base des connaissances locales.



## 5. Inspection du logiciel

L'objectif de ce chapitre est d'introduire la notion d'inspection du logiciel dans le but d'améliorer la qualité du logiciel. Ce chapitre permet de:

- ✓ fournir les avantages des inspections du logiciel
- ✓ Expliciter ce qu'est une inspection et plus particulièrement l'inspection de Fagan.
- ✓ expliciter l'inspection des documents du logiciel

### 5.1. Introduction

Les programmes, dont on dispose, présentent souvent certaines erreurs telles que variables inutilisées, tests booléens incorrects, commentaires qui ne correspondent pas au code, etc. Ce qui nécessite souvent des correction/améliorations. Les inspections du logiciel, les critiques, sont des techniques prouvées pour réduire le nombre de défauts dans un programme.

Dans une organisation de développement des logiciels, une activité d'inspection devrait faire partie du processus de développement du logiciel. L'inspection de Fagan correspond à un processus structuré destiné à trouver des défauts dans des documents du logiciel tels que le code, la (les) conception (s) etc. et évidemment pendant les diverses phases du procédé de développement du logiciel.

### 5.2. Avantages des inspections

Les inspections présentent certains avantages tels que :

- Le coût de la correction d'un défaut augmente considérablement s'il est rencontré tardivement, dans le cycle de développement du logiciel. Pour cela, s'il est possible de détecter et de corriger les erreurs plus tôt, cela permettra d'économiser du temps ainsi que de l'argent tout le long du cycle de vie du logiciel.
- L'inspection ne porte pas sur le code source uniquement mais, tout artefact lisible produit pendant le développement du logiciel peut être consulté à savoir les spécifications des exigences, des documents de conception et de modèles, les plans de test, la documentation du système, etc.
- Les inspections permettent de raccourcir les délais de livraison, en réduisant le temps passé dans les phases de test. Une meilleure qualité du produit fini permet aussi de réduire le temps de maintenance. Ainsi, ce dernier peut être utilisé pour le développement de nouveaux projets.
- Dans une organisation à un niveau élevé de maturité des processus logiciels, les données recueillies lors d'inspections peuvent être utilisées pour améliorer davantage le processus. Les données provenant des rapports de synthèse d'inspection peuvent être utilisées pour identifier les types les plus communs ou les plus coûteux de bugs, de déterminer leurs causes profondes, et changer la façon dont le travail est effectué afin d'éviter ces types d'erreurs.

De plus, Il ya quelques avantages secondaires à savoir:

- ✓ tous les participants peuvent apprendre de chaque inspection. Les inspections offrent une occasion de voir comment les autres membres de l'équipe travaillent. Une inspection fournit un forum efficace pour les membres de l'équipe les moins expérimentés d'en apprendre beaucoup tout en continuant à faire des contributions utiles.



- ✓ Les inspections ont tendance à révéler différents types d'erreurs qui ne sont pas détectées par les activités de test. La combinaison d'inspections formelles et structurées et des tests systématiques fournit un outil puissant pour créer des programmes sans défaut.

### 5.3. Inspections

L'inspection formelle (inspection de Fagan, par exemple) Correspond au fait que quelqu'un d'autre que le créateur d'un produit logiciel examine le produit avec l'intention spécifique de trouver des erreurs. C'est une lecture critique d'un document (spécification, conception, code, plan d'intégration...); Elle est destinée à améliorer la qualité d'un document.

D'une manière générale, l'inspection est faite par une équipe indépendante de l'équipe de développement du projet. Pour qu'elle puisse être bénéfique, une inspection doit donner lieu à la rédaction de fiches de défauts avec une échelle de gravité et la définition des responsabilités concernant la correction des défauts. L'inspection de Fagan est une méthode de revue de groupe employée pour évaluer le rendement d'un processus donné.

Les exemples des activités pour lesquelles l'inspection de Fagan peut être employée sont :

- ▲ Spécifications
- ▲ Architecture de système de logiciel/d'information
- ▲ Programmation (par exemple pour /dans des itérations de XP)
- ▲ Essai de logiciel (par exemple en créant des jeux d'essai) «Un défaut est un exemple dans lequel une condition n'est pas satisfaite.»

#### 5.3.1. Inspection de logiciel

En cours d'inspection de logiciel les défauts qui sont trouvés sont classés en deux catégories : des défauts principaux et des défauts mineurs.

- Les défauts qui concernent une fonctionnalité incorrecte ou même absente peuvent être classifiés en tant que défauts principaux : le logiciel ne fonctionnera pas correctement quand ces défauts ne sont pas résolus.
- Les défauts mineurs ne menacent pas le fonctionnement correct du logiciel, mais sont la plupart du temps de petites erreurs comme des erreurs d'appellation dans les documents ou erreurs du type positionnement incorrect des commandes dans une interface de programme.

Les inspections sont à la base des décisions prises en revues. Une revue est une réunion permettant de valider une des phases du cycle de vie. On distingue

- les revues des produits: état d'un projet sous ses différents aspects: Techniques, Financiers, commerciaux, Calendrier, ...
- les revues techniques: elles permettent de fournir au marketing et à l'unité de développement une évaluation des aspects techniques du projet et des coûts de réalisation
- les réunions de décision: elles valident le passage à la phase suivante et font bien souvent suite à l'une des deux précédentes.

Le processus d'inspection formelle de logiciels développé par Michael Fagan; nécessite une équipe d'inspection composée de 3 à 8 membres ayant les rôles suivants :



- 1) **Le modérateur:** conduit l'inspection, les horaires des réunions, le contrôle des réunions, les résultats des rapports d'inspection et le suivi des questions à reprendre. Les modérateurs devraient être formés sur la manière de mener les inspections, y compris la façon de s'entretenir avec les participants ayant de fortes compétences techniques.
- 2) **L'auteur:** créé ou maintient le produit du travail à inspecter. L'auteur ne peut répondre aux questions posées sur le produit pendant l'inspection, et il examine aussi des défauts. L'auteur ne peut pas servir en tant que modérateur, lecteur ou secrétaire.
- 3) **Le lecteur :** décrit les sections du produit du travail à l'équipe lors de l'inspection. Il peut paraphraser ce qui se passe dans le produit, tel que la description d'une section de code, mais il n'a pas l'habitude de lire mot à mot le produit.
- 4) **Le secrétaire:** classe et enregistre les défauts et les questions soulevées lors de l'inspection.

Le modérateur pourrait remplir ce rôle dans une petite équipe d'inspection.

- 5) **L'inspecteur :** tente de trouver des erreurs dans le produit. Tous les participants sont réellement considérés comme des inspecteurs, en plus de toute autre responsabilité. Les inspecteurs comprennent: la personne qui a créé la spécification du produit du travail inspecté (par exemple, le concepteur d'une inspection de code), ceux chargés de l'application, des tests, ou du maintien du produit, un représentant d'assurance qualité pour agir selon les normes.

### 5.3.2. Le processus d'inspection

Le processus d'inspection se compose d'activités suivantes (d'après Fagan) dont chacune a un objectif spécifique;

- **Planification :** Le modérateur sélectionne l'équipe d'inspection, prépare les matériaux qui doivent être inspectés, les distribue ainsi que d'autres documents pertinents à l'équipe d'inspection à l'avance. Le matériel devrait être distribué au moins deux ou trois jours avant l'inspection.
- **Vue d'ensemble-** Cette réunion donne à l'auteur l'occasion de décrire les caractéristiques importantes du produit à l'équipe d'inspection. Il peut ne pas la faire si cette information est déjà connue par les autres participants.
- **Préparation** - Chaque participant est responsable de l'examen des artefacts de travail avant la réunion de l'inspection réelle, en notant tous les défauts trouvés ou questions qui seront soulevées. Peut-être 75% des erreurs constatées lors des inspections sont identifiés lors de l'étape de préparation. Le produit doit être comparé à ses prédecesseurs (spécification) des documents pour évaluer l'exhaustivité et l'exactitude.
- **Réunion d'inspection-** Lors de cette session, l'équipe se réunit et est dirigé par le produit du travail par le modérateur et le lecteur. Si le modérateur détermine au début de la réunion que le manque de temps a été consacré à la préparation par les participants, la réunion devrait être reportée.

Pendant la discussion, tous les inspecteurs peuvent signaler des défauts ou soulever d'autres questions, qui sont documentés sur un formulaire par le secrétaire. La réunion ne devrait pas durer plus de deux heures. Lors de sa conclusion, le groupe s'engage sur une évaluation du produit: Accepté comme tel; accepté avec révisions mineures; révisions majeures nécessaires et une deuxième inspection requise; ou reconstruire le produit.

- **Reprise-** L'auteur est chargé de résoudre toutes les questions soulevées pendant l'inspection. Cela ne signifie pas nécessairement faire tout changement qui a été suggéré, mais une décision explicite doit être faite sur la manière dont chaque défaut sera traité.



- **Suivi-** Pour vérifier que la reprise nécessaire a été effectuée correctement, le modérateur est responsable du suivi avec l'auteur. Si une fraction importante (soit, 10 pour cent) du produit du travail a été modifiée, une inspection supplémentaire peut être nécessaire.

Ce travail semble être important et même fastidieux, mais l'expérience a montré que ce processus d'inspection formelle est le moyen le plus efficace permettant de trouver des défauts dans un produit logiciel

### 5.3.3. Inspection des documents

Le document est l'élément d'examen. Il ya trois aspects à cette tâche: les défauts enregistrés lors de la réunion d'inspection, collecte de données auprès de multiples inspections, et l'analyse des tendances d'un défaut pour évaluer l'efficacité d'inspection et d'identifier la façon d'améliorer le processus de développement logiciel afin de prévenir les types courants de défauts.

Chaque erreur trouvée est décrite avec suffisamment de détails afin de permettre à l'auteur de se référer à cette liste (de détails) au cours de l'étape de reprise et de comprendre le problème qui a été soulevé. Les références à des numéros de ligne où les erreurs ont été trouvées sont importantes pour des défauts particuliers, mais des observations plus générales sur les violations des normes ou des suggestions de style peuvent s'appliquer à l'ensemble du produit.

La liste des défauts d'une inspection devrait aboutir à un rapport de synthèse avec un comptage des défauts de chaque catégorie. Le but est de permettre aux gestionnaires de savoir comment le projet avance et de chercher des domaines où des améliorations doivent être apportées.

Le modérateur est habituellement responsable de la préparation de ces rapports post-inspection.

Un processus d'inspection efficace permet à une organisation de comparer les données provenant de multiples inspections afin de mieux comprendre la qualité à la fois du processus d'examen et les produits en cours de révision.

L'objectif ultime est d'avoir une base de données d'inspection afin que des conclusions quantitatives puissent être tirées des tendances des défauts ainsi que des informations sur les processus d'inspection.

## 5.4. Conclusion

- ✓ L'inspection du logiciel est considérée comme une méthode de test statique permettant de vérifier que le logiciel répond aux besoins.
- ✓ La mise en œuvre d'inspection du logiciel est importante car elle permet d'aboutir à un processus de développement logiciel plus mature. L'amélioration continue des processus mène à l'amélioration simultanée de la qualité du produit logiciel et de la productivité et les inspections peuvent y jouer un rôle majeur.
- ✓ L'inspection permet de détecter un taux de défauts entre 60% et 90%.
- ✓ Un important avantage à long terme d'un programme d'inspection est l'idée qu'il peut fournir des types de défauts créés et des changements de processus dans le but de les prévenir.
- ✓ Enfin, la qualité de l'inspection a un effet direct sur la qualité du produit.



## 6. Estimation du coût du logiciel

L'objectif de ce chapitre est d'introduire les techniques d'estimation du coût et de l'effort nécessaires à la production d'un logiciel. Il s'agit de :

- ✓ comprendre que le coût du logiciel n'est pas directement lié au coût du développement
- ✓ introduire l'estimation de la productivité du logiciel
- ✓ introduire les techniques d'estimation du coût
- ✓ introduire le modèle d'estimation du coût COCOMO
- ✓ introduire la notion des points de fonction et de la valeur acquise

### 6.1. Introduction

Dans la planification d'un projet logiciel, l'estimation consiste à déterminer :

- l'effort nécessaire pour compléter chaque activité.
- Le temps nécessaire pour compléter chaque activité.
- Le coût total pour chaque activité.

L'estimation du coût est effectuée en même temps que la programmation du projet. Trois paramètres interviennent dans le calcul du coût d'un projet de développement du logiciel:

- coût du matériel et du logiciel ainsi que le coût de la maintenance
- coût de formation et de déplacement
- coût de l'effort.

Pour la majorité des projets, le coût de l'effort est le plus dominant relativement aux différents coûts qui s'y ajoutent tels que le coût des charges des bureaux, le coût de l'équipe support (administrateur), le coût de communication, etc.

### 6.2. Productivité du logiciel

La productivité correspond aux nombres d'unités produites par le nombre de heures-hommes (ou mois-hommes) nécessaires à leur production.

L'estimation de la productivité des ingénieurs est une tâche difficile pour le chef de projet. L'estimation de la productivité est basée sur les mesures des attributs du logiciel par l'effort total nécessaire au développement.

On dispose de deux types de métriques:

- Métriques liées à la taille : c'est la métrique la plus utilisée. La taille peut être mesurée en:
  - Nombre de lignes du code source livré
  - Nombre d'instructions du code objet livré
  - Nombre de pages de documentation du système
- Métriques liées aux fonctions : liées aux fonctionnalités générales du logiciel livré. La productivité est exprimée en termes de total des fonctionnalités utiles produites pendant un certain temps donné. Les points fonctions et les points objets sont les métriques les plus utilisés.



La détermination du nombre de points de fonction est donnée dans la section suivante. Les points objet représentent une alternative des points de fonction qui peuvent être utilisés avec les langages de programmation des bases de données.

**Exemple :** Le nombre d'écran séparé qui est affiché représente un point objet ayant comme coefficient: Simple écran →1, modérément complexe →2, très complexe →3.

### 6.3. L'estimation du coût d'un logiciel

L'estimation du coût du logiciel consiste à déterminer les ressources nécessaires à sa réalisation. Cette estimation est faite généralement en **Mois-Hommes (MH)**.

Il existe deux approches très différentes pour estimer le coût d'un logiciel. La plus ancienne est appelée estimation **LOC (Lines Of Code)** car elle s'appuie sur l'estimation du nombre de lignes de code qui devront être écrites pour le projet. La plus moderne s'appuie sur le nombre de points de fonction dans la description du projet.

#### 6.3.1. Estimation des lignes de code

La première étape dans l'estimation fondée sur le nombre de ligne de code consiste à estimer leur nombre dans le code du projet terminé. Cette estimation peut s'appuyer sur la taille des précédents projets, celle d'un produit concurrent ou bien diviser le projet en plusieurs parties, plus simples à estimer.

La démarche standard consiste, pour chaque partie  $i$  du projet, à estimer sa taille maximale  $max_i$ , sa taille minimale  $min_i$  et une approximation au plus juste  $j_i$ .

L'estimation de la taille des différentes parties est la suivante:

$$\text{Taille estimée} = \text{Somme}((\text{Taille}_{min} + 4 * \text{approximation} + \text{Taille}_{max})/6) \text{ LOC}$$

**Exemple :** L'équipe WRT(West River Telecom) a décomposé son projet en 7 parties, dont l'estimation des tailles respectives figure dans le tableau suivant:

Partie	Taille <sub>min</sub>	Approximation	Taille <sub>max</sub>	Taille estimée
p1	20	30	50	31,67 LOC
p2	10	15	25	15,83 LOC
P3	25	30	45	31,67 LOC
P4	30	35	40	35,00 LOC
P5	15	20	25	20,00 LOC
P6	10	12	14	12,00 LOC
P7	20	22	25	22,17 LOC

#### 6.3.2. Estimation du coût fondé sur les lignes de code

L'approche LOC repose sur une formule qui utilise des données historiques. La version la plus simple comprend trois paramètres :  $\text{Coût} = a \times \text{KLOC}^b + c$

Avec : a: le coût marginal par KLOC (*Kilo LOC*), c'est-à-dire le coût de chaque millier de lignes de code supplémentaires.



b: est un exposant qui reflète le caractère non linéaire de cette relation. Selon des études menées sur ce sujet, la valeur de b n'est pas bien définie. Si  $b > 1$  le coût du KLOC augmente avec la taille du projet → pas d'économie d'échelle. Si  $b < 1$  il y a économie d'échelle.

c: représente le coût fixe du projet, sa valeur peut être positive ou nulle.

### 6.3.3. Le modèle COCOMO

COCOMO (Constructive Cost Model) est la formule classique d'estimation du coût fondée sur le nombre de lignes de code. Introduite en 1981 par Barry Boehm, il utilise le millier d'instructions source livrées (KDSI: Kilo Delivered Source Instruction) comme unité de taille ce qui est équivalent au KLOC et Mois-Homme comme unité de coût. Ce modèle existe en trois versions: simple, intermédiaire et détaillé. **Trois types de projet** ont été identifiés:

- **Projets de mode organique** : ces projets sont réalisés par une équipe relativement petite, dans un environnement familier et dans un domaine d'application connu de l'équipe.
- **Projets de mode semi-détaché** : ce mode représente un intermédiaire entre le mode organique et le mode embarqué. L'équipe peut être composée de programmeurs de divers niveaux d'expérience. Les membres de l'équipe ont une expérience limitée.
- **Les projets de mode embarqué** : la caractéristique principale d'un tel projet est que le système doit fonctionner sous des contraintes particulièrement fortes. Le système à développer est une partie d'un système complexe et fortement connecté de matériels, de logiciels, de normes et de procédures opérationnelles.

Pour chaque catégorie, il attribue différentes valeurs aux paramètres de la formule :

- Mode organique :  $MH = 2.4 \times KDSI^{1.05}$
- Mode semi-détaché :  $MH = 3.0 \times KDSI^{1.12}$
- Mode embarqué :  $MH = 3.6 \times KDSI^{1.20}$

Boehm a également proposé, à partir de ses données, un temps de développement standard en fonction des ressources de personnel disponible et du type de projet. Les formules pour calculer le **Temps de Développement (TDEV)** sont:

- Q11. Mode organique :  $TDEV = 2.5 \times MH^{0.38}$
- Q12. Mode semi-détaché :  $TDEV = 2.5 \times MH^{0.35}$
- Q13. Mode embarqué :  $TDEV = 2.5 \times MH^{0.32}$

Relativement au modèle COCOMO simple, vu ci-dessus, le modèle intermédiaire prend en considération d'autres facteurs, autre que la taille et le type de projet. Les facteurs introduits sont au nombre de 15 répartis en **4 classes**: les attributs du produit, les attributs de l'environnement matériel et logiciel, les attributs du projet et les attributs liés au personnel.

- **Les attributs du produit** : sont fiabilité requise du logiciel (FIAB), taille de la base de données (DONN), complexité du produit (CPLX).
- **Les attributs de l'environnement matériel et logiciel** : sont contraintes de temps d'exécution (TEMP), contraintes d'espace mémoire (ESPA), volatilité de la machine virtuelle (VIRT), contraintes du système de développement (CSYS).
- **Les attributs du personnel** : sont aptitude à l'analyse (APTA), l'expérience dans le domaine d'application (EXPA), expérience de la machine virtuelle (EXPV), aptitude à la programmation (APTP), expérience du langage de programmation (EXPL).



- **Les attributs du projet** : sont méthodes de programmation modernes (PMOD) outils logiciels (OLOG), durée requise du développement (DREQ).

Les multiplicateurs associés à ces attributs sont décrits dans le tableau suivant avec **TB** signifie: très bas, **B**: bas, **M**: moyen, **E**: élevé, **TE**: très élevé et **TTE**: très très élevé.

Attributs	TB	B	M	E	TE	TTE
FIAB	0.75	0.88	1.00	1.15	1.4	
DONN		0.94	1.00	1.08	1.16	
PLX	0.70	0.85	1.00	1.15	1.30	1.65
TEMP			1.00	1.11	1.30	1.66
ESPA			1.00	1.06	1.21	1.56
VIRT		0.87	1.00	1.15	1.30	
CSYS		0.87	1.00	1.07	1.15	
APTA	1.46	1.19	1.00	0.86	0.71	
EXPA	1.29	1.13	1.00	0.91	0.82	
APTP	1.42	1.17	1.00	0.86	0.70	
EXPV	1.21	1.10	1.00	0.90		
EXPL	1.14	1.07	100	0.95		
PMOD	1.24	1.10	1.00	0.91	0.82	
OLOG	1.24	1.10	1.00	0.91	0.83	
DREQ	1.23	1.08	1.00	1.04	1.10	

## 6.4. Analyse des points de fonction

Le principe des points de fonction consiste à identifier et à quantifier les fonctionnalités réclamées par le projet, en comptant les éléments extérieurs qui devront être traités. Les éléments les plus couramment pris en compte sont les suivants:

- **Les interrogations**: ce sont les paires requête-réponse qui ne modifient pas les données internes. Par exemple: une requête pour connaître l'adresse d'un employé.
- **Les entrées** : ce sont les données fournies au programme. Par exemple: l'entrée du nom, du prénom d'un employé.
- **Les sorties** : les données qu'affiche le programme. Par exemple: afficher une liste des employés de l'entreprise, l'ensemble de ces lignes sera compté comme un élément unique.
- **Les fichiers internes** : ce sont les fichiers que le client comprend que le système doit gérer. Par exemple: un fichier qui comprend 1000 entrées de données Personnelles sera compté comme un seul fichier mais, s'il contient d'une part les données personnelles et d'autre part des données relatives au département, il sera compté comme deux fichiers différents.
- **Les interfaces externes** : ce sont les données partagées avec d'autres programmes. Par exemple : le fichier personnel qui peut être utilisé par le service paie pour calculer les payes sera considéré comme une interface dans les deux systèmes.

### 6.4.1. Comptage des points de fonctions bruts

Les points de fonction recensés sont classés en trois catégories: simples, moyens ou complexes. Les coefficients (voir tableau ci-dessous) sont ensuite appliqués aux différents éléments et la somme de ces produits s'appelle les points de fonctions bruts.



	<b>Simple</b>	<b>Moyen</b>	<b>Complex</b>
Sorties	4	5	7
Interrogations	3	4	6
Entrées	3	4	6
Fichiers	7	10	15
Interfaces	5	7	10

Coefficients des points de fonction

Cas où le point de fonction correspond aux interrogations avec deux catégories simples et moyennes.

Type	Simple	Moyenne	Complexe	Total
interrogations	Requête par nom	Rendez-vous	-	7

#### 6.4.2. Analyse de la valeur acquise

Elle consiste à calculer ce qui a été accompli pour mesurer le progrès effectué.

Mesures de base :

- ✓ **CBT** (Coût Budgété du Travail) : est la quantité de travail estimée pour une tâche donnée.
- ✓ **CBA** (Coût Budgété à l'Achèvement): est la somme de tous les quantités de travail estimées.
- ✓ **CBTE** (Coût Budgété du Travail Effectué): est la somme de tous les quantités de travail estimées qui sont effectivement achevées.
- ✓ **CBTP** (Coût Budgété du Travail Prévu): est la somme de tous les quantités de travail estimées qui sont effectivement achevées à une date donnée.
- ✓ **CRTE** (Coût Réel du Travail Effectué): est la somme de tous les quantités de travail réelles qui sont effectivement achevées.
- ✓ **VP** (Valeur Prévue) : est la proportion de la quantité de travail totale estimée attribuée à une tache donnée **VP = CBT/CBA**.

#### 6.4.3. Indicateurs d'avancement

- ✓ **VA** (Valeur Acquise) ou **PA** (Pourcentage achevé): est la somme des VP pour les tâches achevées **VA = PA = CBTE/CBA**.
- ✓ **IPT** (Indicateur de Performance Temporel): **IPT = CBTE/CBTP**.
- ✓ **VE** (Variance à l'Echéancier): variance par rapport à l'échéancier **VE = CBTE – CBTP**.
- ✓ **IC** (Indicateur des Coûts): indicateur d'écart sur les coûts **IC = CBTE/CRTE**.
- ✓ **VC** (Variance des Coûts): variance par rapport aux coûts **VC = CBTE – CRTE**.

Tache	Travail estimé (jh)	Travail réel aujourd'hui	Date d'achèvement estimée	Date d'achèvement effective
1	5	10	25/01	01/02
2	25	20	15/02	15/02
3	120	80 (140 au problème 2)	15/05	(01/07 au problème 2)
4	40	50	15/04	15/04
5	60	50	01/07	
6	80	70	01/09	

**Problème 1:** calculer les indicateurs d'avancement au 01/04



Le CBA : somme des estimations des quantités de travail

- $CBA = 5+25+120+40+60+80 = 330jh$

Les tâches 1,2 et 4 sont achevées

- Le CBTE est la somme des CBT pour ces tâches
- $CBTE = 5+25+40 = 70jh$
- $VA = CBTE/CBA = 70/330 = 21,2\%$

Les tâches 1 et 2 devraient être achevées pour le 01/04, et pas 1,2 et 4 donc :

- $CBTP = 5+25 = 30jh$
- $IPT = CBTE/CBTP = 70/30 = 233\%$
- $VE = CBTE - CBTP = +40jh$

La CRTE est la somme des quantités de travail réelles pour les tâches 1,2 et 4

- $CRTE = 10+20+50 = 80jh$
- $IC = CBTE/CRTE = 70/80 = 87,5\%$
- $VC = CBTE - CRTE = 70 - 80 = -10jh$

**Problème 2:** que deviennent ces indicateurs, à supposer que la tâche 3 a également été achevée avec 140jh de travail et que nous sommes le 01/07 ?

- $CBTE = 190jh$
- $CBTP = 250jh$
- $CRTE = 220jh$
- $VA = 190/330 = 57.5\%$
- $IPT = 190/250 = 76\%$
- $VE = 190 - 250 = -60jh$

Seules les tâches 1–4 sont réalisées, au lieu de 1–5

- $IC = 190/220 = 86.6\%$
- $VC = 190 - 220 = -30jh$

## 6.5. Conclusion

- ✓ Les facteurs qui affectent la productivité incluent l'aptitude individuelle (facteur dominant), expérience du domaine, processus de développement, la taille du projet, le support outil et l'environnement de travail.
- ✓ Le modèle COCOMO est le modèle le mieux développé, il prend en considération les attributs du projet, du produit, du matériel et du personnel dans la formulation de l'estimation du coût.
- ✓ Le modèle COCOMO 81 suppose que le logiciel est développé selon un modèle de processus en cascade et utilisant un langage de programmation impératif (C ou FORTRAN).
- ✓ Cependant, les changements effectués dans le développement du logiciel à savoir les processus logiciels adoptés (processus de développement incrémental et itératif), le développement de logiciel par assemblage de composants réutilisables, la technique de réingénierie utilisée pour créer un nouveau logiciel à partir d'un logiciel existant, etc. ont contribué à donner lieu au modèle COCOMO II (COCOMO 2) qui prend en compte ces changements.
- ✓ Le temps nécessaire pour compléter un projet n'est pas simplement proportionnel à l'effort requis. L'ajout du personnel en fin de projet peut augmenter plutôt que diminuer le temps prévu pour terminer le projet. Enfin, l'analyse de la valeur acquise permet de mesurer les progrès du développement réalisés pour mieux contrôler le projet.



## **7. Gestion des ressources humaines**

Texte...

## **8. Gestion de la configuration**

Texte...

## **9. Métriques du logiciel**

Texte...

## **10. Récapitulation de la gestion des projets**

Texte...



## La partie : Méthodes

### 1. Méthode Planning de PERT

#### 1.1. Généralités

Dans le cas de fabrications non répétitives et très complexes, (prototypes, avions, grands ensembles en construction, etc .....), on utilisera la méthode dite du chemin critique dont la plus utilisée est la méthode P.E.R.T.

Cette méthode a été mise au point en 1957 aux Etats-Unis, lors du développement du missile POLARIS. Ce projet mobilisait 250 fournisseurs principaux et environ 9000 sous-traitants. Le délai initial prévu de 6 ans a pu être ramené à 2 années et demie.

**P.E.R.T** (Program Evaluation Review Technique) : technique d'élaboration et de mise à jour de programme

#### 1.2. BUT de la Méthode P.E.R.T

- ✓ Définir le délai total d'accomplissement de l'œuvre et éventuellement proposer des moyens pour le réduire.
- ✓ Connaitre les conséquences du changement de la durée d'une tâche partielle.
- ✓ Evaluer les moyens à mettre en œuvre.
- ✓ Etablir une relation entre les délais et les coûts.

#### 1.3. Domaine d'Application de la Méthode P.E.R.T

- ✓ Dans le Bâtiment (grands ensembles, hôpitaux, etc .....) )
- ✓ Dans les Travaux Public (routes, ponts, etc .....) )
- ✓ Pour l'Ordonnancement de prototypes.
- ✓ En Maintenance pour coordonner les tâches de plusieurs équipes de spécialités différentes

#### 1.4. Conditions de mise en œuvre de la Méthode P.E.R.T

- ✓ L'œuvre doit être divisée en tâches partielles.
- ✓ La durée de chaque tâche doit être connue.
- ✓ L'étude technique doit préciser si certaines tâches doivent être impérativement effectuées avant certaines autres tâches.

#### 1.5. Principe de la Méthode basée sur une Représentation Graphique

Sur le Graphique ou Diagramme apparaîtront les liaisons entre les différentes tâches de l'œuvre à réaliser.

##### 1.5.1. Définitions et Conventions de Représentation

**Tâche :** Une tâche fait évoluer l'œuvre vers son état final, elle consomme donc du temps, de l'énergie, de la matière et de ce fait coûte. Chaque tâche est représentée par une flèche (segment orienté dans le sens de l'écoulement du temps) dont la longueur est indépendante de la durée de la tâche.

**Symbole :**



- A = Identification de la tâche
- 5 = Durée de la tâche

**Tâche fictive :** Une tâche fictive représente une contrainte entre tâches non indépendantes. Chaque tâche fictive est représentée par une flèche pointillée, sa durée est nulle, elle ne consomme aucune ressource, elle ne coûte donc rien.

**Symbole :**

**Etape :** On appelle étape le début ou la fin d'une tâche. Une étape sera représentée par un cercle ou une autre figure géométrique. Une étape est de durée nulle, elle ne coûte donc rien.

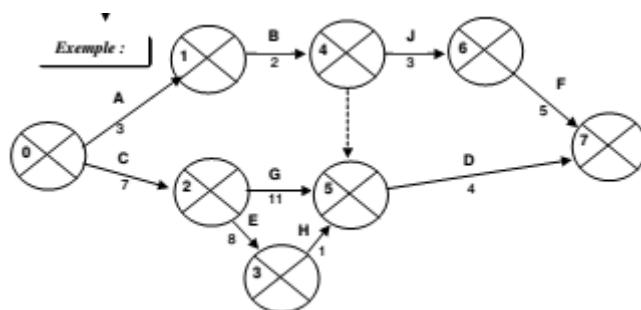
**Symbole :**



- 9 = Identification de l'étape
- 34 = Date au plus tôt
- 38 = Date au plus tard
- 4 = Marge totale

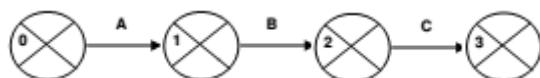
**Réseau :** c'est l'ensemble des tâches et des étapes qui représente l'œuvre. Le réseau met en évidence les relations entre les tâches et les étapes.

**Exemple :**



**Par Convention :**

- Des tâches **Consécutives** sont des tâches qui se suivent.
- Des tâches **Antérieures** sont des tâches qui, par rapport à une autre, doivent être réalisées avant.
- Les Antériorités immédiates sont appelées **Antécédentes**.



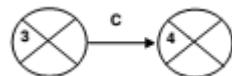
- A et B sont antériorités de C
- B est antécédente de C



### 1.5.2. Conventions de Constructions du Planning P.E.R.T

**1ère Convention :** toute tache a une étape début pour origine et une étape fin pour extrémité

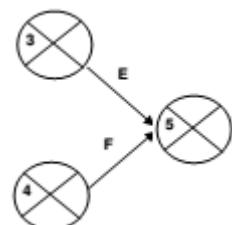
**Exemple :**



La tâche C commence à l'étape 3 et se termine à l'étape 4.

**2ème Convention :** une étape ne peut être atteinte que lorsque les taches qui la précédent sont toutes terminées.

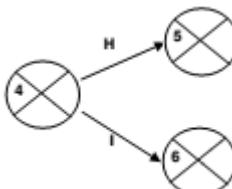
**Exemple :**



L'étape 5 ne sera atteinte que lorsque les tâches E et F seront terminées.

**3ème Convention :** Aucune tache ne peut commencer tant que l'étape située a son origine n'est pas atteinte.

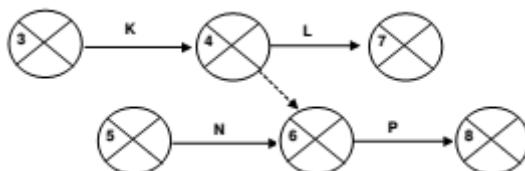
**Exemple :**



Les tâches H et I ne pourront commencer que lorsque l'étape 4 sera atteinte.

**4ème Convention :** l'étape située a l'extrémité d'une fictive ne peut être atteinte qu'après l'étape située a son origine.

**Exemple :**



- L'étape 6 ne peut être atteinte que : lorsque l'étape 4 est atteinte, et lorsque la tâche N est terminée.
- Lecture : La tâche L a pour antécédente K. La tâche P a pour antécédente N et K.

**Application n°1 :**

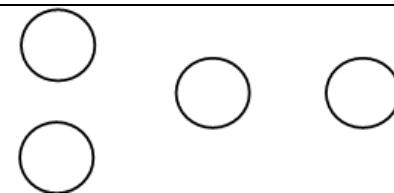
Positionner les tâches dans les différents exercices proposés ci-dessous.

**Exercice n°1 :**

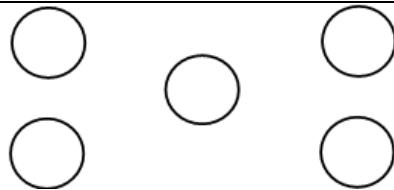
- A a pour antériorité B.
- B a pour antériorité Z.

**Exercice n°2 :**

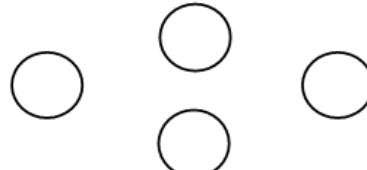
- La tâche B a pour antériorités A et Z

**Exercice n°3 :**

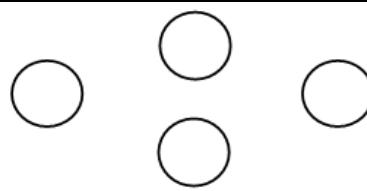
- Les tâches K et A ont pour antériorités L et B.

**Exercice n°4 :**

- La tâche Z a pour antériorités A et B.
- La tâche L a pour antériorité B.

**Exercice n°5 :**

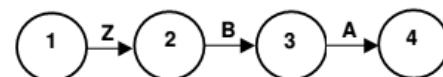
- La tâche Z a pour antériorité A.
- La tâche L a pour antériorités A et B.

**Corrigé n°1 :**

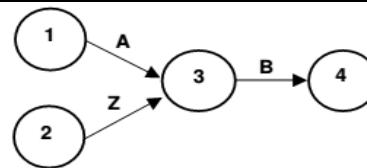
Positionner les tâches dans les différents exercices proposés ci-dessous.

**Exercice n°1 :**

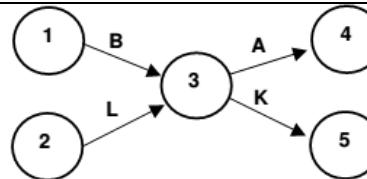
- A a pour antériorité B.
- B a pour antériorité Z.

**Exercice n°2 :**

- La tâche B a pour antériorités A et Z

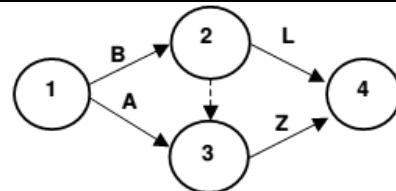
**Exercice n°3 :**

- Les tâches K et A ont pour antériorités L et B.

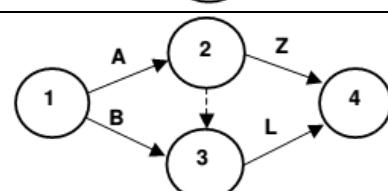


**Exercice n°4 :**

- La tâche Z a pour antériorités A et B.
- La tâche L a pour antériorité B.

**Exercice n°5 :**

- La tâche Z a pour antériorité A.
- La tâche L a pour antériorités A et B.

**1.5.3. Recherche des Rangs ou Niveaux**

- ➔ **Sont de RANG 1** : Les tâches qui n'ont pas de tâches antérieures.
- ➔ **Sont de RANG 2** : Les tâches qui ont pour antécéentes les tâches de rang 1.
- ➔ **Sont de RANG 3** : Les tâches qui ont pour antécéentes les tâches: de rang 2.

- ✓ La même logique est à appliquer jusqu'aux dernières tâches.
- ✓ Cas particulier : Lorsqu'une tâche a plusieurs antécéentes, on prend la tâche de rang le plus élevé.

**Application n°2 :**

Classer les tâches dans leurs différents rangs.

Antériorités	Opération	Rang
Rien	A	
A C	B	
Rien	C	
C	D	

1 <sup>er</sup> RANG	2 <sup>ème</sup> RANG
.....	.....

Antériorités	Opération	Rang
Rien	A	
A	B	
Rien	C	
A C	D	
A C	E	

1 <sup>er</sup> RANG	2 <sup>ème</sup> RANG
.....	.....

Antériorités	Opération	Rang
Rien	A	
A	B	
B E G	C	
Rien	D	
Rien	E	
E G	F	
A D	G	
E	H	

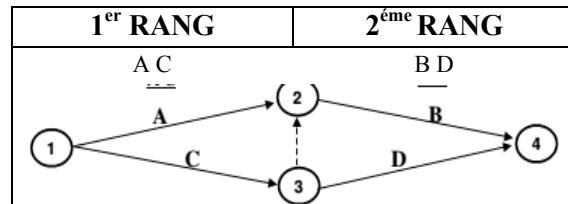
1 <sup>er</sup> RANG	2 <sup>ème</sup> RANG	3 <sup>ème</sup> RANG
.....	.....	.....



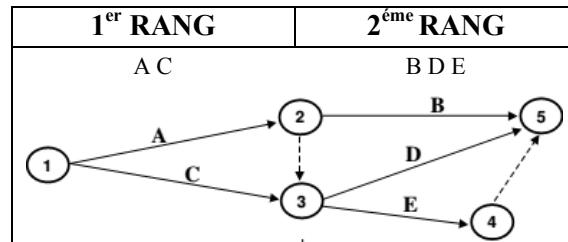
## Corrigé n°2 :

Classer les tâches dans leurs différents rangs.

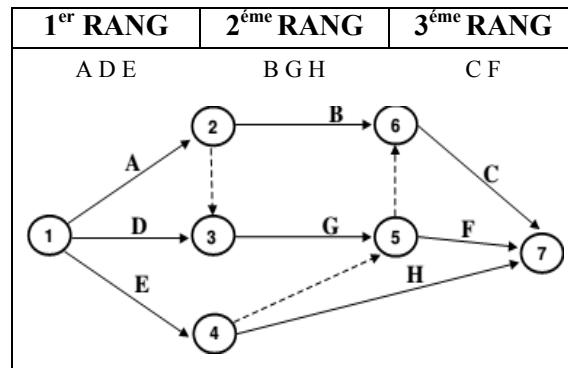
Antériorités	Opération	Rang
Rien	A	
A C	B	
Rien	C	
C	D	



Antériorités	Opération	Rang
Rien	A	
A	B	
Rien	C	
A C	D	
A C	E	



Antériorités	Opération	Rang
Rien	A	
A	B	
B E G	C	
Rien	D	
Rien	E	
E G	F	
A D	G	
E	H	

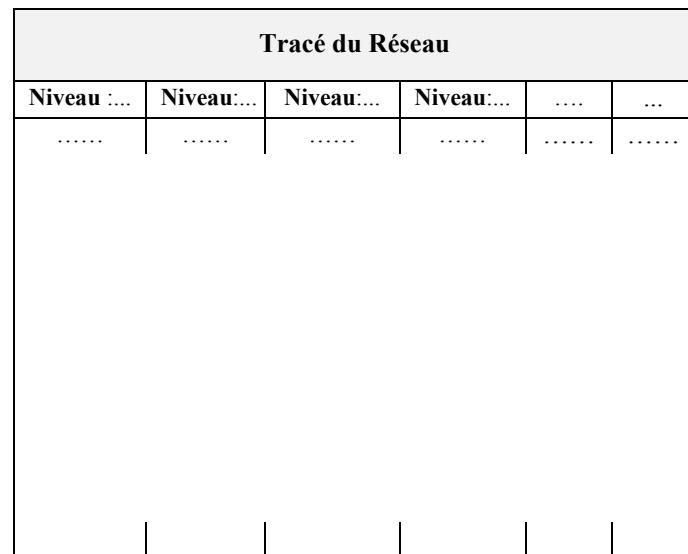


## 1.5.4. Construction du Réseau P.E.R.T

## Application N°3

1. Déterminer les niveaux des différentes tâches.
2. Tracer le réseau (méthode P.E.R.T)
3. Calculer les dates au plus tôt, les dates au plus tard.
4. Mettre en évidence le **chemin critique**.

REP Opération	Taches	Durée	Antériorités	Rang
A	X	4	Rien	
B	X	1	A	
C	X	6	A	
D	X	2	A	
E	X	7	A	
F	X	3	B	
G	X	4	C	
H	X	7	C	
I	X	3	D J	
J	X	4	E	
K	X	8	E	
L	X	1	F G	
M	X	2	H I K L	





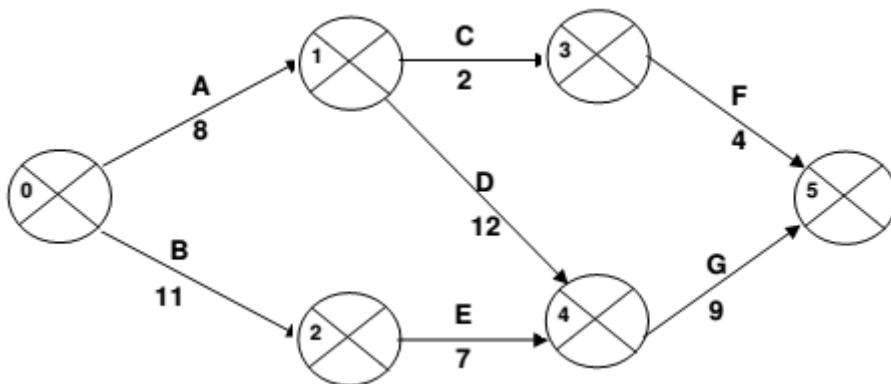
## Corrigé N°3

Trace du Réseau					
Niveau : 1	Niveau : 2	Niveau : 3	Niveau : 4	Niveau : 5	/
A	B C D E	F G H J K	I L	M	/
<b>Observations :</b> calcul des dates au plus tard avec détermination du chemin critique.					

## 1.5.5. Calcul des dates au plus tôt :

Date au plus tôt d'une étape =  $\max(\text{date au plus tôt de l'étape précédente} + \text{durée de la tâche comprise entre les 2 étapes})$ .

- ✓ Au dessous, à gauche du symbole de chaque étape, porter la date au plus tôt de l'étape; c'est la date à laquelle l'étape peut être atteinte au plus tôt.
- ✓ On procède dans l'ordre croissant des étapes.
- ✓ Quand il y a plusieurs tâches convergentes, on ne retient que la valeur la plus grande.
- ✓ La date de la dernière étape représente le temps normal d'exécution.

Exemple de calcul :

- ▲ Le calcul part de l'origine, étape 0, au temps 0.
- ▲ L'étape 1, n'est reliée à l'origine que par la tâche A, qui dure 8 jours.
- ▲ L'étape 1 est donc atteinte au plus tôt à :  $0 + 8 = 8$  jours.
- ▲ L'étape 2 est atteinte au plus tôt à :  $0 + 11 = 11$  jours.



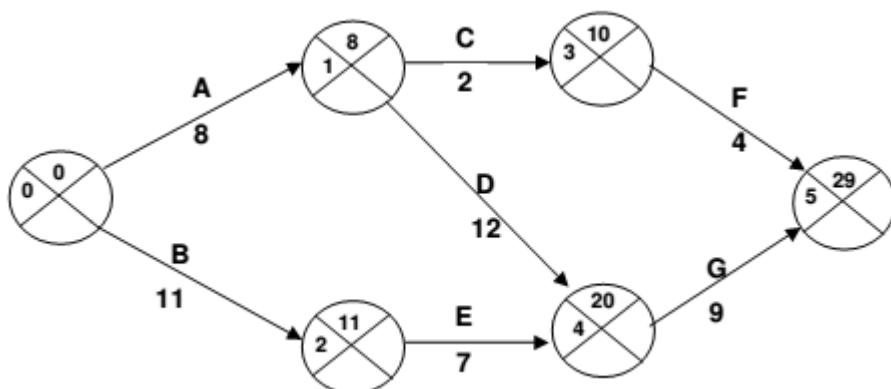
- ▲ L'étape 4 est atteinte par 2 " chemins " : A + D et B + E soit :  $8 + 12 = 20$  jours et  $11 + 7 = 18$  jours.
- ▲ Pour que l'étape 4 soit atteinte, il faut que D et E soient achevées ; on prend le maximum donc 20 jours.
- ▲ Le calcul se poursuit en utilisant cette même méthode.

### 1.5.6. Calcul des dates au plus tard :

Date au plus tard d'une étape = min (date au plus tard de l'étape suivante - durée de la tâche comprise entre les 2 étapes).

- ✓ Au dessous, à droite du symbole de chaque étape, porter la date au plus tard de l'étape ; c'est la date à laquelle l'étape peut être atteinte au plus tard.
- ✓ On procède dans l'ordre décroissant des étapes.
- ✓ Quand il y a plusieurs dates au plus tard à une étape, on ne retient que la valeur la plus petite.
- ✓ A l'étape 0 la date au plus tard doit être 0.

#### Exemple de calcul :

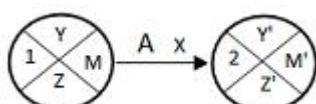


- ▲ Le calcul part de l'étape finale et remonte vers le début du projet.
- ▲ Pour l'étape 3 on a :  $29 - 4 = 25$  jours.
- ▲ Pour l'étape 4 on a :  $29 - 9 = 20$  jours.
- ▲ Pour l'étape 2 on a :  $20 - 7 = 13$  jours.
- ▲ Pour l'étape 1 on a :  $20 - 12 = 8$  jours.  $25 - 2 = 23$  jours. on retient le minimum donc 8 jours.
- ▲ Pour l'étape 0 on a :  $8 - 8 = 0$  jours.  $13 - 11 = 2$  jours. on retient le minimum donc 0 jours.

### 1.5.7. Calculs des différentes marges :

- ✓ **La marge totale** : C'est le retard maximum que peut prendre une tâche sans porter atteinte au plus tard de la tâche suivante (donc sans retarder la fin des travaux). C'est la différence entre la date au plus tard et la date au plus tôt.  $MT(t) = Dd+td(t) - Dd+tt(t) = Df+td(t) - Df+tt(t)$
- ✓ **La marge libre** : C'est le retard maximum que peut prendre une tâche sans porter atteinte au plus tôt de la tâche suivante.  $ML(t) = Dd+tt(t+1) - Dd+tt(t) - D(t)$

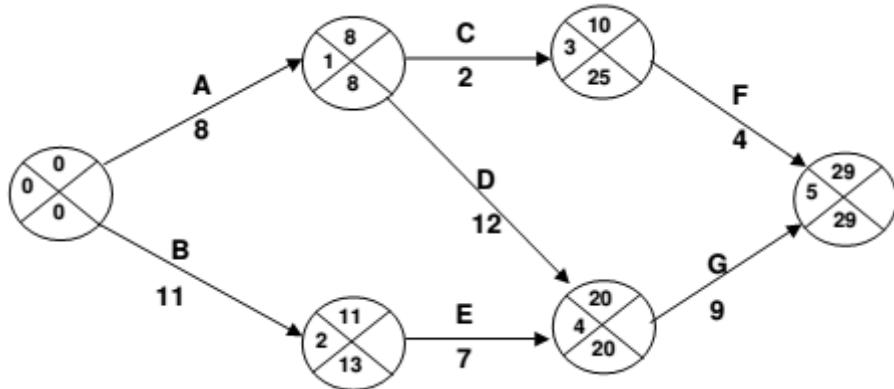
#### Exemple :



- ✓  $MT(A) = Z' - (X + Y)$
- ✓  $ML(A) = Y' - (X + Y)$
- ✓  $Y' = X + Y$  et  $Z = Z' - X$



### La marge totale :



- ▲ Le calcul de la marge totale de la tâche A sera :  $8 - (8 + 0) = 0$
- ▲ Le calcul de la marge totale de la tâche B sera :  $11 - (11 + 0) = 0$
- ▲ Le calcul de la marge totale de la tâche C sera :  $10 - (8 + 2) = 0$
- ▲ Le calcul de la marge totale de la tâche D sera :  $25 - (12 + 8) = 5$
- ▲ Le calcul de la marge totale de la tâche E sera :  $20 - (7 + 11) = 2$
- ▲ Le calcul de la marge totale de la tâche F sera :  $29 - (4 + 10) = 15$
- ▲ Le calcul de la marge totale de la tâche G sera :  $29 - (9 + 20) = 0$
- ▲ -----
- ▲ Le calcul de la marge libre de la tâche A sera :  $8 - (8 + 0) = 0$
- ▲ Le calcul de la marge libre de la tâche B sera :  $11 - (11 + 0) = 0$
- ▲ Le calcul de la marge libre de la tâche C sera :  $10 - (8 + 2) = 0$
- ▲ Le calcul de la marge libre de la tâche D sera :  $25 - (12 + 8) = 5$
- ▲ Le calcul de la marge libre de la tâche E sera :  $20 - (7 + 11) = 2$
- ▲ Le calcul de la marge libre de la tâche F sera :  $29 - (4 + 10) = 15$
- ▲ Le calcul de la marge libre de la tâche G sera :  $29 - (9 + 20) = 0$

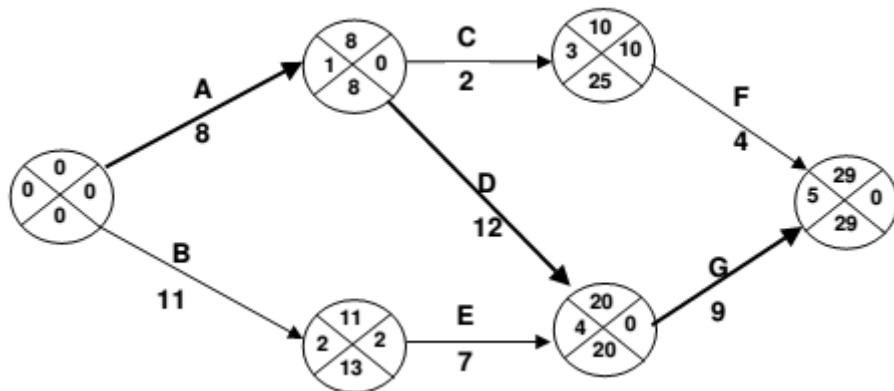
### **1.6. Exploitation du réseau :**

#### Constatations :

- ✓ On peut remarquer que certaines étapes présentent des dates au plus tôt **Différentes** des dates au plus tard, cela traduit une **Marge** qui autorise une certaine souplesse dans la réalisation des tâches.  
**Exemple :** L'étape 15 sera atteinte au plus tôt 10 jours après le début du projet, mais pourrait être atteinte au plus tard 25 jours après le début du projet.
- ✓ Quand la date au plus tôt est identique à la date au plus tard, la marge est NULLE et on dit que l'étape est critique.

#### Le chemin critique

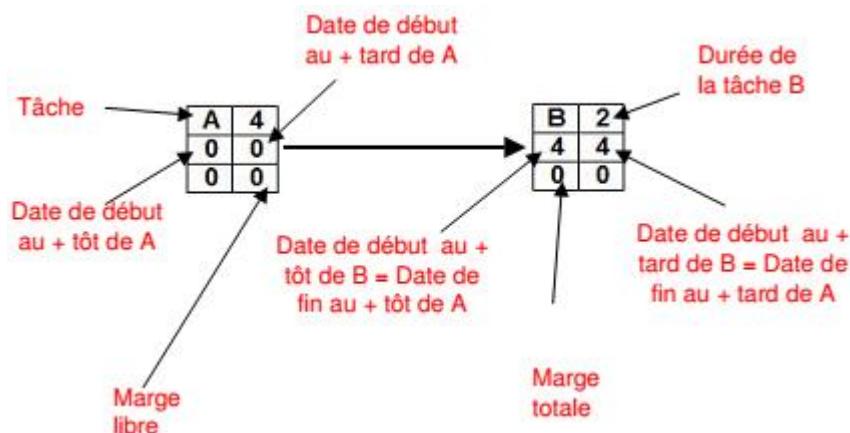
- ✓ C'est le chemin formé par les étapes de marge nulle.
- ✓ Pour un même projet, il peut y avoir plusieurs **chemins critiques**.
- ✓ Le chemin critique devra être tracé de l'étape 0 à l'étape finale du projet, en reliant toutes les **étapes critiques**.
- ✓ Par convention on le représente par une flèche orientée rouge barrée de 2 traits.



### 1.7. La méthode MPM :

- Le MPM (méthode des potentiels métra) est méthode proche du PERT qui sert également à déterminer et diminuer la durée de réalisation d'un projet.
- La différence entre ces deux méthodes sera un changement au niveau de la représentation du graphe.
- Ici, les tâches seront représentées par des sommets et les flèches définiront les liens d'antériorités.
- On notera également la disparition des tâches fictives.
- On trouvera les mêmes paramètres que pour le PERT avec les dates au plus tôt et au plus tard, les marges libres et totales et le chemin critique.

#### Symbolisation :





## 2. Méthode Planning de PERT Probabiliste

Cette technique s'appuie sur le graphe des antécédents; permet d'inclure dans la planification le risque et l'incertitude attachés à chaque tache et d'en déduire une durée du projet assortie d'un niveau de probabilité la durée de chaque tache peut être considérée comme une variable aléatoire.

C'est-à-dire que l'on peut faire plusieurs estimations ( $\pm$  probables) de la durée de la tache. Alors la durée de tout chemin dans le graphe PERT est également considérée une variable aléatoire; puisque c'est une somme de variables aléatoires. Sous réserves des conditions suivantes:

- ✓ un nombre suffisamment élevé de taches (min 4 taches)
- ✓ un ordre de grandeur semblable pour toutes les taches
- ✓ l'indépendance entre les durées des taches.

La durée probable du chemin obéit à une loi de distribution proche de loi normale (Laplace-gauss), dont la représentation graphique de la courbe en cloche dite de gauss.

Le calcul des paramètres du PERT probabiliste se fait en trois étapes:

### 2.1. La première étape

Consiste à déterminer la loi de probabilité attachée à chaque tache. En pratique on retient souvent une loi de distribution béta c'est dire qu'on est capable de donner trois estimations :

- $T_{opt}$  estimation optimiste de la durée; c'est le temps minimal possible, tout se déroule que prévu
- $T_{pes}$  estimation pessimiste de la durée, c'est le temps maximal possible, si tout déroule au plus mal (hors catastrophe)
- $T_{vrai}$  durée vraisemblable de la durée c'est que l'on donnerait si on devait n'en donner qu'une seule

### 2.2. La deuxième étape

Consiste à calculer trois valeurs pour chaque tache  $i$  :

- La durée probable  $t_{prob}(i)$ : c'est une moyenne statistique, c'est le temps moyen de la tache si elle est répétée un grand nombre de fois; Sa variance  $v(i)$  et son écart -type  $e(i)$  : plus les estimations optimistes et pessimistes sont éloignées; plus elles présentent d'incertitude. C'est la variance qui mesure cette incertitude. C'est la variance qui mesure cette incertitude. Si elle est faible, l'estimation de la durée probable de la tache sera plus précise.
  - $t_{prob}(i) = (T_{opt}(i) + 4 * t_{vrai}(i) + T_{pes}(i)) / 6$
  - $e(i) = (T_{pes}(i) - T_{opt}(i)) / 6$
  - $V(i) = e(i)^2$

### 2.3. La troisième étape

Consiste à calculer pour chaque chemin :

- Sa durée estimée  $D_{est}$
- Sa variance estimée  $V_{est}$
- Son écart-type estimé  $E_{est}$
- $D_{est} = \text{Somme } (t_{prob}(i))$  pour toutes les taches  $i$  du chemin
- $V_{est} = \text{somme}(e(i)^2)$  pour toutes les taches  $i$  du chemin
- $E_{est} = (V_{est})^{1/2}$



- Comme la durée du chemin obéit à une loi normale; on peut utiliser la table de gauss pour obtenir la durée  $D_p$ :  $D_p = D_{est} + G(p)E_{est}$

La durée estimée du projet est probable à 50% (car  $G(50) = 0$ )

Soit par exemple un chemin de durée estimée  $D_{est} = 100$  et d'écart-type estimé  $E_{est} = 15$ . On peut en déduire soit une durée probable pour une probabilité donnée; soit une probabilité d'achèvement dans un délai donné. Ainsi,

- ✓ La durée probable à 90 % est:  $D_{90} = 100 + G(90) * 15 = 100 + 1.28 * 15 = 119$  jours.
- ✓ La durée probable à 60 % est:  $D_{60} = 100 + G(60) * 15 = 100 + 0.26 * 15 = 104$  jours.
- ✓ La probabilité de terminer en 80 jours est:  $80 = 100 + G(P) * 15$  d'où  $G(p) = 1,33$
- ✓ La probabilité est comprise entre 0,09 et 0,1.

- On peut donner une mesure du ratio:  $R = (t_{pes} - t_{opt}) / t_{pes}$

- ✓  $R < 0,25$  le risque est faible
- ✓  $0,25 < R < 0,50$  le risque est moyen
- ✓  $R > 0,5$  le risque est fort

Le PERT probabiliste est intéressante pour les projets à forte incertitude, particulièrement sur le chemin critique, ainsi que sur les chemins ayant les plus forts risques,

Si l'incertitude est faible, la différence ( $t_{pes} - t_{opt}$ ) est faible par rapport à la durée estimée de la tâche, dans ce cas, les variances des tâches sont faibles de même que la variance du chemin, donc  $E_{est}$  on n'a pas donc besoin d'un PERT probabiliste. On peut se contenter d'un graphe de la méthode des antécédents.

## 2.4. Exercice Recette

Soit à estimer la durée d'une recette de gâteau au chocolat. Une incertitude pèse sur la durée de la préparation. Elle provient de différents facteurs: la qualité du four, la disponibilité de certains outillages (batteur électrique...) et la rapidité. Le tableau donne la liste des tâches, ainsi les estimations optimiste, pessimiste et vraisemblable de chaque tâche.

	Tâches	$t_{opt}$	$t_{pes}$	$t_{vrai}$
R1	Faire fondre le beurre et le chocolat	6	9	7,5
R2	Séparer les œufs en jaunes et blancs	1	4,5	3
R3	Ajuster les jaunes au mélange fondu et faire cuire en 2 mn	6	8	7
R4	Monter les blancs en neige	2	12	5
R5	Retirer le mélange du feu et incorporer les blancs	2	6	3
R6	Faire cuire au four	16	22	18

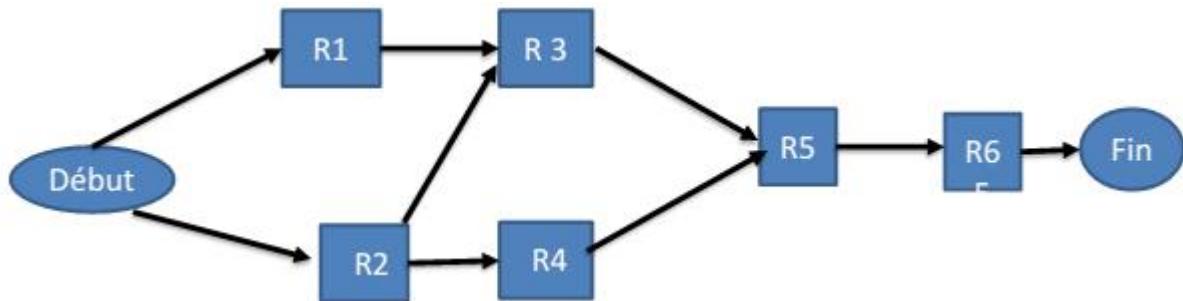
- Calculer le risque, la durée probable, l'écart-type et la variance de chaque tâche.
- Quel est le chemin critique.
- Quelle est la durée estimée de préparation du gâteau P=95% et P=90%,
- Quelle est la probabilité que l'on termine en 37 minutes.



## Solution

On considère le graphe des antécédents où les tâches sont les sommets au lieu des étapes.

1°) Le graphe



2°) Les paramètres probabilistes

Tâches	$T_{opt}$	$t_{pes}$	$t_{vrai}$	Risque	$T_{prob(i)}$	$E(i)$	$V(i)$
R1	6	9	7,5	0,33	7,50	1/2	0,25
R2	1	4,5	3	0,78	2,92	7/12	0,34
R3	6	8	7	0,25	7,00	1/3	0,11
R4	2	12	5	0,83	5,67	5/3	2,78
R5	2	6	3	0,67	3,33	2/3	0,44
R6	16	22	18	0,27	18,33	1	1,00

Chemins	$D_{est}$	$V_{est}$	$E_{est}$
R1R3R5R6	36,17	1,81	1,34
R2R3R5R6	31,58	1,90	1,38
R2R4R5R6	30,25	4,56	2,14

3°) Les durées probables des différents chemins sont données sur le tableau suivant

- ✓  $D_{95} = D_{est} + 1.65 * E_{est}$
- ✓  $D_{90} = D_{est} + 1,28 * E_{est}$

Chemins	$D_{95}$	$D_{90}$
R1R3R5R6	38,38	37,89
R2R3R5R6	33,86	33,35
R2R4R5R6	33,77	32,98

4°) La probabilité d'une durée de 37 minutes se calcule sur le chemin critique

- ✓  $D_p = D_{est} + G(p) * E_{est}$  soit  $37=36,17+G(p) * 1,34$
- ✓  $G(p) = 0,619$  d'où la probabilité est environ 0,72.



## 2.5. Fonction de répartition de la loi normale standard

$u_a$	0,00	0,01	0,02	0,03	0,04	0,05	0,06	0,07	0,08	0,09
0,0	0,5000	0,5040	0,5080	0,5120	0,5160	0,5199	0,5239	0,5279	0,5319	0,5359
0,1	0,5398	0,5438	0,5478	0,5517	0,5557	0,5596	0,5636	0,5675	0,5714	0,5753
0,2	0,5793	0,5832	0,5871	0,5910	0,5948	0,5987	0,6026	0,6064	0,6103	0,6141
0,3	0,6179	0,6217	0,6255	0,6293	0,6331	0,6368	0,6406	0,6443	0,6480	0,6517
0,4	0,6554	0,6591	0,6628	0,6664	0,6700	0,6736	0,6772	0,6808	0,6844	0,6879
0,5	0,6915	0,6950	0,6985	0,7019	0,7054	0,7088	0,7123	0,7157	0,7190	0,7224
0,6	0,7257	0,7291	0,7324	0,7357	0,7389	0,7422	0,7454	0,7486	0,7517	0,7549
0,7	0,7580	0,7611	0,7642	0,7673	0,7704	0,7734	0,7764	0,7794	0,7823	0,7852
0,8	0,7881	0,7910	0,7939	0,7967	0,7995	0,8023	0,8051	0,8078	0,8106	0,8133
0,9	0,8159	0,8186	0,8212	0,8238	0,8264	0,8289	0,8315	0,8340	0,8365	0,8389
1,0	0,8413	0,8438	0,8461	0,8485	0,8508	0,8531	0,8554	0,8577	0,8599	0,8621
1,1	0,8643	0,8665	0,8686	0,8708	0,8729	0,8749	0,8770	0,8790	0,8810	0,8830
1,2	0,8849	0,8869	0,8888	0,8907	0,8925	0,8944	0,8962	0,8980	0,8997	0,9015
1,3	0,9032	0,9049	0,9066	0,9082	0,9099	0,9115	0,9131	0,9147	0,9162	0,9177
1,4	0,9192	0,9207	0,9222	0,9236	0,9251	0,9265	0,9279	0,9292	0,9306	0,9319
1,5	0,9332	0,9345	0,9357	0,9370	0,9382	0,9394	0,9406	0,9418	0,9429	0,9441
1,6	0,9452	0,9463	0,9474	0,9484	0,9495	0,9505	0,9515	0,9525	0,9535	0,9545
1,7	0,9554	0,9564	0,9573	0,9582	0,9591	0,9599	0,9608	0,9616	0,9625	0,9633
1,8	0,9641	0,9649	0,9656	0,9664	0,9671	0,9678	0,9686	0,9693	0,9699	0,9706
1,9	0,9713	0,9719	0,9726	0,9732	0,9738	0,9744	0,9750	0,9756	0,9761	0,9767
2,0	0,9772	0,9778	0,9783	0,9788	0,9793	0,9798	0,9803	0,9808	0,9812	0,9817
2,1	0,9821	0,9826	0,9830	0,9834	0,9838	0,9842	0,9846	0,9850	0,9854	0,9857
2,2	0,9861	0,9864	0,9868	0,9871	0,9875	0,9878	0,9881	0,9884	0,9887	0,9890
2,3	0,9893	0,9896	0,9898	0,9901	0,9904	0,9906	0,9909	0,9911	0,9913	0,9916
2,4	0,9918	0,9920	0,9922	0,9925	0,9927	0,9929	0,9931	0,9932	0,9934	0,9936
2,5	0,9938	0,9940	0,9941	0,9943	0,9945	0,9946	0,9948	0,9949	0,9951	0,9952
2,6	0,9953	0,9955	0,9956	0,9957	0,9959	0,9960	0,9961	0,9962	0,9963	0,9964
2,7	0,9965	0,9966	0,9967	0,9968	0,9969	0,9970	0,9971	0,9972	0,9973	0,9974
2,8	0,9974	0,9975	0,9976	0,9977	0,9977	0,9978	0,9979	0,9979	0,9980	0,9981
2,9	0,9981	0,9982	0,9982	0,9983	0,9984	0,9984	0,9985	0,9985	0,9986	0,9986
3,0	0,9987	0,9987	0,9987	0,9988	0,9988	0,9989	0,9989	0,9989	0,9990	0,9990
3,1	0,9990	0,9991	0,9991	0,9991	0,9992	0,9992	0,9992	0,9992	0,9993	0,9993
3,2	0,9993	0,9993	0,9994	0,9994	0,9994	0,9994	0,9994	0,9995	0,9995	0,9995
3,3	0,9995	0,9995	0,9995	0,9996	0,9996	0,9996	0,9996	0,9996	0,9996	0,9997
3,4	0,9997	0,9997	0,9997	0,9997	0,9997	0,9997	0,9997	0,9997	0,9997	0,9998
3,5	0,9998	0,9998	0,9998	0,9998	0,9998	0,9998	0,9998	0,9998	0,9998	0,9998
3,6	0,9998	0,9998	0,9999	0,9999	0,9999	0,9999	0,9999	0,9999	0,9999	0,9999

$u_a$	0,8416	1,2816	1,6449	1,9600	2,0537	2,3263	2,5758	2,8070	3,0902	3,2905
$\alpha$	0,8000	0,9000	0,9500	0,9750	0,9800	0,9900	0,9950	0,9975	0,9990	0,9995

## 2.6. Inverse de fonction de répartition de la loi normale standard

$\alpha$	0,00	0,01	0,02	0,03	0,04	0,05	0,06	0,07	0,08	0,09
0,0	$\infty$	2,5758	2,3263	2,1701	2,0537	1,9600	1,8808	1,8119	1,7507	1,6954
0,1	1,6449	1,5982	1,5548	1,5141	1,4758	1,4395	1,4051	1,3722	1,3408	1,3106
0,2	1,2816	1,2536	1,2265	1,2004	1,1750	1,1503	1,1264	1,1031	1,0803	1,0581
0,3	1,0364	1,0152	0,9945	0,9741	0,9542	0,9346	0,9154	0,8965	0,8779	0,8596
0,4	0,8416	0,8239	0,8064	0,7892	0,7722	0,7554	0,7388	0,7225	0,7063	0,6903
0,5	0,6745	0,6588	0,6433	0,6280	0,6128	0,5978	0,5828	0,5681	0,5534	0,5388
0,6	0,5244	0,5101	0,4959	0,4817	0,4677	0,4538	0,4399	0,4261	0,4125	0,3989
0,7	0,3853	0,3719	0,3585	0,3451	0,3319	0,3186	0,3055	0,2924	0,2793	0,2663
0,8	0,2533	0,2404	0,2275	0,2147	0,2019	0,1891	0,1764	0,1637	0,1510	0,1383
0,9	0,1257	0,1130	0,1004	0,0878	0,0753	0,0627	0,0502	0,0376	0,0251	0,0125
1,0	0,0000	-	-	-	-	-	-	-	-	-

$\alpha$	0,001	0,0001	0,00001	0,000001	0,0000001	1,00E-08	1,00E-09	1,00E-10	1,00E-11	1,00E-12
$u_a$	3,2905	3,8906	4,4172	4,8916	5,3267	5,7307	6,1094	6,4670	6,8065	7,1305



### 3. Méthode Planning de GANTT

#### 3.1. Généralités

Le suivi de l'avancement de l'œuvre peut se faire directement sur le réseau P.E.R.T .Ce n'est pas toujours très facile car les longueurs des tâches ne sont pas proportionnelles à leurs durées

On peut avantageusement **transformer le réseau en un graphique de GANTT** qui lui tient compte de la durée des tâches. On l'appelle également **planning à bandes**

#### 1°) BUT de la Méthode GANTT

- Permet de déterminer la charge des postes de travail.
- Permet de répartir chronologiquement les postes de travail.
- Permet de suivre l'avancement de la fabrication, par pièce et par ensemble.
- Permet de vérifier la disponibilité de la main d'œuvre.
- Permet d'établir la situation des stocks.
- Permet d'établir le calendrier des opérations ou tâches à mettre en œuvre pour réaliser un projet.

#### 2°) Domaine d'Application de la Méthode GANTT

Il se construit dans les mêmes domaines que le graphique P.E.R.T.

#### 3°) Conditions de mise en œuvre de la Méthode GANTT

Son application est indispensable après le Graphique P.E.R.T si le projet est complexe, ou seule si le projet est simple.

#### 4°) Principe de la Méthode basée sur une Représentation Graphique

Les tâches sont représentées par des segments de droite dont la longueur est proportionnelle à la durée de la tâche .Chaque tâche utilise une ligne.

#### 5°) Construction du réseau GANTT avec calcul des marges libres et marges totales.

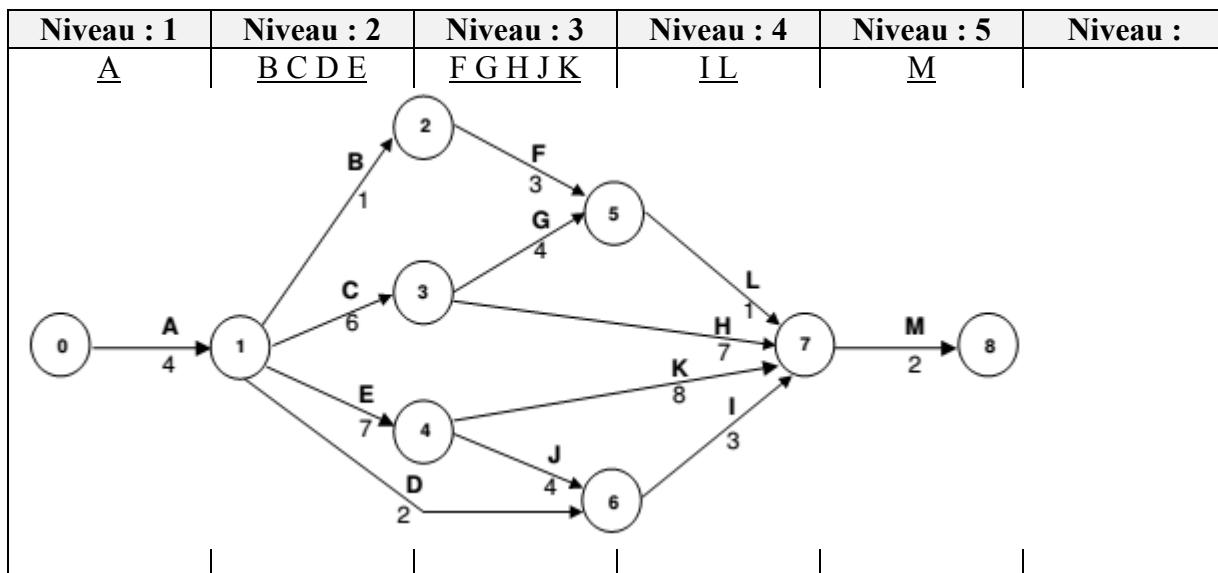
##### Application N°1

1. Déterminer les niveaux des différentes tâches.
2. Tracer le réseau (méthode GANTT)
3. Calculer les marges libres, les marges totales.
4. Proposer un ajustement des tâches au plus tard.

REP Opération	Taches	Durée	Antériorités
A	X	4	Rien
B	X	1	A
C	X	6	A
D	X	2	A
E	X	7	A
F	X	3	B
G	X	4	C
H	X	7	C
I	X	3	D J
J	X	4	E
K	X	8	E
L	X	1	F G
M	X	2	H I K L



### 1 ) Trace du réseau PERT :



**Observations** : calcul des dates au plus tard avec détermination du chemin critique

### 3.2. Utilisation de la matrice

1°) Remplir les 2 colonnes de gauche :

- ❖ Inscrire dans la 1ère colonne la nature de la tâche à effectuer. dans la 2ème colonne, le repère de la tâche à identifier. **Exemple** : Déchargement et stockage des matériaux pour colonne 1, A pour le repère de cette même tâche.

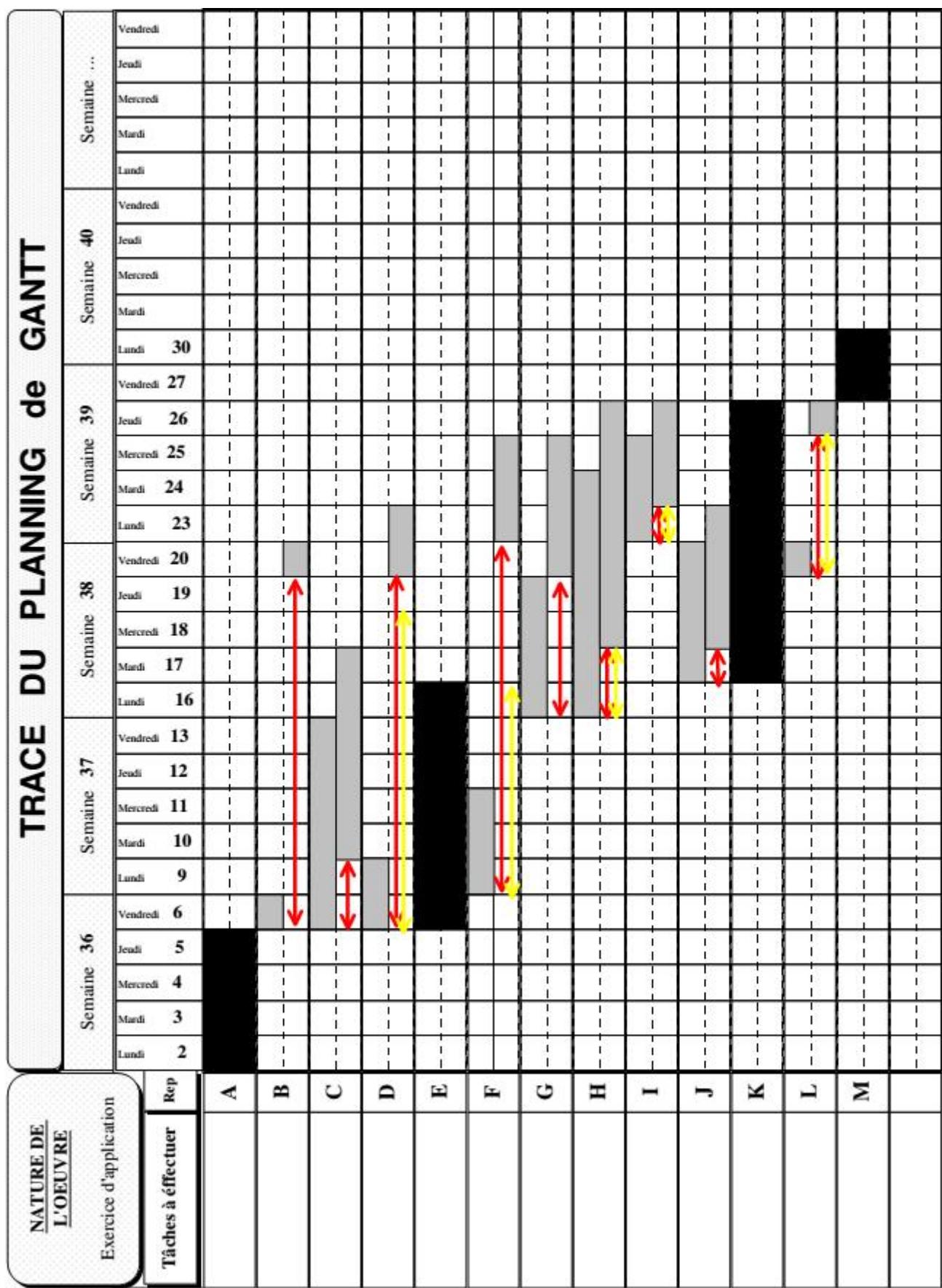
2°) Remplir les 2 lignes du haut :

- ❖ Pour la première ligne, mentionner la période ou date à laquelle le projet doit être réalisé.
- ❖ Pour la seconde ligne, inscrire la dénomination temps à laquelle l'œuvre est exécuté dans chacune des colonnes. **Exemple** : Heure, journée, semaine, etc...

3°) Remplir le tableau :

- ❖ Pour chaque tâche répertoriée, composée ainsi de 2 lignes, tracer **sur la ligne supérieure** un segment de longueur proportionnelle à la durée de la tâche, en se basant sur **les dates au plus tôt**.
- ❖ Chaque tâche devra être ainsi reportée sur le graphique en tenant compte des antériorités qui les lient entre elles. Le segment représentant une durée de tâche appartenant au **chemin critique** devra être symbolisé par un **trait fort**.
- ❖ Refaire pour chaque ligne la même démarche en tenant toujours compte des proportionnelles liées à la durée, des antériorités qui les lient et des liaisons qui les lient dans le déroulement de l'exécution de l'œuvre.
- ❖ **La seconde ligne** apparaissant pour chacune des tâches, est réservée pour la représentation des tâches par rapport aux dates au plus tard. Si le diagramme n'est pas composé de deux lignes pour chaque tâche, il faudra effectuer deux diagrammes de GANTT. **Un diagramme au plus tôt et un au plus tard**.

## 2 ) Trace du diagramme GANTT :





### 3.3. Calcul des marges totales

Une marge totale d'une tâche est un retard possible au démarrage de cette tâche par rapport à l'instant de départ au plus tôt qui n'entraîne aucun recul de la date d'achèvement de l'œuvre, mais qui imposera que les tâches suivantes soient commencées aux dates au plus tard.

$$\begin{aligned} \boxed{\text{Marge totale}} &= \boxed{\text{Date de fin au plus tôt de la tâche}} - \boxed{\text{Date de début au plus tôt de la tâche}} \\ &= \boxed{\text{Date de fin au plus tard de la tâche}} - \boxed{\text{Date de début au plus tard de la tâche}} - \boxed{\text{Temps nécessaire à l'exécution de cette même tâche}} \end{aligned}$$

### 3.4. Calcul des marges libres

Une marge libre d'une tâche est un retard toléré par cette tâche par rapport à l'instant de départ au plus tôt (ou un temps d'interruption en cours d'exécution, ou un allongement de la durée prévue) qui n'entraîne aucune modification du calendrier des tâches en aval et notamment aucun recul de la date d'achèvement de l'œuvre.

$$\boxed{\text{Marge libre}} = \boxed{\text{Date de début au plus tôt de la tâche suivante}} - \boxed{\text{Date de début au plus tôt de la tâche}} - \boxed{\text{Temps nécessaire à l'exécution de cette même tâche}}$$

### 3.5. Utilisation des marges

Selon les circonstances, elles pourront être utilisées :

- ✓ Pour conserver une sécurité de temps sur les tâches qui ne sont pas situées sur le chemin critique. Les tâches sont déclenchées au plus tôt.
- ✓ Pour réduire le coût des en-cours. Les tâches sont alors déclenchées au plus tard au risque de dépasser la durée impartie

#### → Utilisation du tableau

- Complétez la colonne des supériorités.
- Calculez les dates de début au plus tôt et au plus tard pour chaque tâche.
- Calculez les dates de fin au plus tard et au plus tôt pour chaque tâche.
- Déterminez les marges et le chemin critique.

**Tableau calcul des marges - PERT analytique**

Tâches	Antériorités	Supériorités	Durée	Début		Fin		Marges		Chemin Critique
				+Tôt	+Tard	+Tôt	+Tard	Total	Libre	
A	Rein	<b>BCDE</b>	4	<b>0</b>	0	<b>4</b>	4	<b>0</b>	0	X
B	A	<b>F</b>	1	<b>4</b>	14	<b>5</b>	15	<b>10</b>	0	
C	A	<b>GH</b>	6	<b>4</b>	6	<b>10</b>	12	<b>2</b>	0	
D	A	<b>I</b>	2	<b>4</b>	14	<b>6</b>	16	<b>10</b>	9	
E	A	<b>JK</b>	7	<b>4</b>	4	<b>11</b>	11	<b>0</b>	0	X
F	B	<b>L</b>	3	<b>5</b>	15	<b>8</b>	18	<b>10</b>	6	
G	C	<b>L</b>	4	<b>10</b>	14	<b>14</b>	18	<b>4</b>	0	
H	C	<b>M</b>	7	<b>10</b>	12	<b>17</b>	19	<b>2</b>	2	
I	D J	<b>M</b>	3	<b>15</b>	16	<b>18</b>	19	<b>1</b>	1	
J	E	<b>I</b>	4	<b>11</b>	12	<b>15</b>	16	<b>1</b>	0	
K	E	<b>M</b>	8	<b>11</b>	11	<b>19</b>	19	<b>0</b>	0	X
L	F G	<b>M</b>	1	<b>14</b>	18	<b>15</b>	19	<b>4</b>	4	
M	H I K L	<b>Rien</b>	2	<b>19</b>	19	<b>21</b>	21	<b>0</b>	0	X



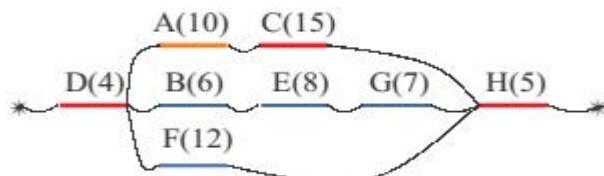
### 3.6. Diagramme de GANTT

Cette méthode, datant de 1918 et encore très répandue, consiste à déterminer la manière de positionner les différents tâches d'un projet à exécuter, sur une période déterminée. Chaque tâche est représentée par segment de droite dont la longueur est proportionnelle au temps.

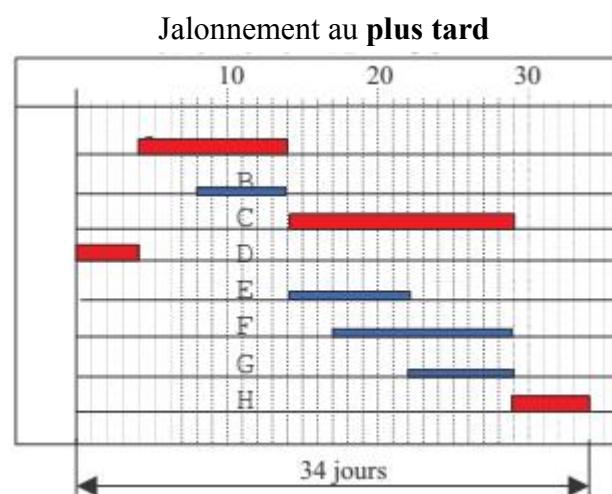
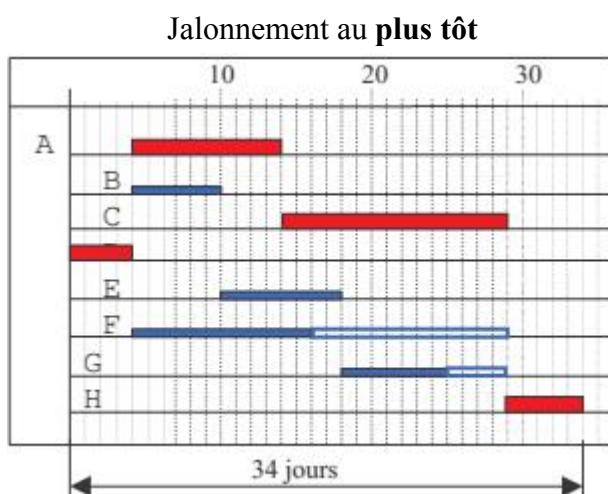
**Exemple 1 :** Soit un projet comprenant 8 tâches définies par le tableau suivant :

Tâche	Tâche antérieures	Durée (jours)
A	D	10
B	D	6
C	A	15
D	/	4
E	B	8
F	D	12
G	E	7
H	F, C, G	5

Réseau de tâches en fonction de contraintes



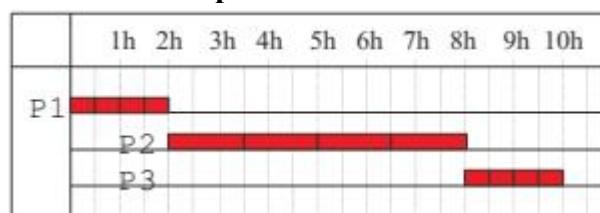
En rouge, le chemin critique est composé des tâches dites critiques pour lesquelles un retard éventuel de réalisation entraînerait une augmentation globale de la durée de projet (34 jours)



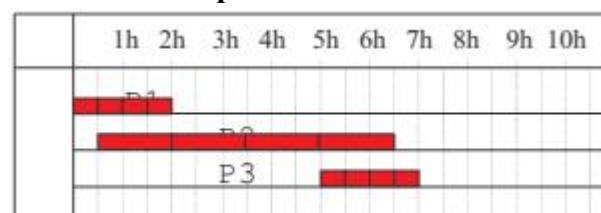
Flottement existant entre deux tâches correspond au retard que peut prendre une tâche particulière sans pour autant augmenter la durée globale de réalisation du projet.

**Exemple 2 :** Soit l'ordonnancement de la production de 100 pièces référencées ZCC et devant subir des opérations sur les postes P1, P2 et P3.

Jalonnement au plus tôt



Jalonnement au plus tard avec chevauchement



Dans le premier cas la production se termine au bout de 10heures. Si les lots sont fractionnés en 4, il est possible d'effectuer un chevauchement. Cela va se traduire par un transfert au poste suivant toutes les 25 pièces. La production se termine maintenant au bout de 7 heures, on a gagné 3 heures.

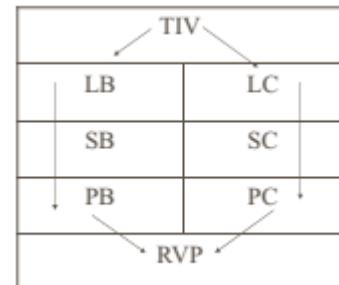


## Exercice:

Le centre d'accueil de St-Romuald Le centre d'accueil de St-Romuald a un seul employé affecté au service de la buanderie. Les équipements disponibles sont une laveuse, et une sécheuse. On vous demande d'analyser ce processus de production en utilisant le diagramme de Gantt. L'ordre des travaux est résumé dans le tableau suivant :

- Trouvez un ordre adéquat de déroulement de l'activité en prenant en considération que le linge blanc est lavé séparément du linge de couleur. (contrainte arbitraire: débutez avec le linge blanc)
- Complétez le graphique de Gantt plus loin.

- Sur ce graphe, ordonnez les activités dans l'ordre chronologique.
- Débutez chaque activité le plus tôt possible de façon à ce que le lavage soit terminé rapidement, tout en respectant les activités préalables.



- Quelle est la durée totale de l'activité faire la lessive? (réponse : 79 min.)
- Tracer en jaune le chemin critique.
- Pendant combien de temps fonctionne la machine à laver?
- Pendant combien de temps les deux appareils électriques fonctionnent-ils en même temps?
- Pendant combien de minutes l'employé travaille-t'il durant un cycle de lavage (de 79 minutes) ? Combien de temps libre a-t-il?
- Quelles activités peuvent être retardées sans allonger la durée du lavage (marge totale) ? De combien de minutes chacune?
- Faites le total durée de fonctionnement machine à laver \_\_\_\_ durée de fonctionnement sécheuse \_\_\_\_ temps de travail de l'employé \_\_\_\_ total \_\_\_\_\_. Pourquoi ce total est-il supérieur à 79 minutes?

Sur une feuille que vous intitulerez Graphique de Gantt 2, préparez un nouveau processus en débutant cette fois par le lavage de la couleur.

- Quelle est la durée totale du projet Faire la lessive?
- Tracer en jaune le chemin critique.
- On veut minimiser le plus possible la durée du cycle de lavage. Combien de temps sauverait-on en achetant une autre laveuse et une autre sécheuse. Un seul employé demeure affecté à cette tâche?

Activité	code	min.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35
Triage initial des vêtements	TIV	4																																				
Lavage- couleur	LC	10																																				
Séchage- couleur	SC	15																																				
Pliage- couleur	PC	17																																				
Lavage- blanc	LB	14																																				
Séchage- blanc	SB	20																																				
Pliage- blanc	PB	13																																				
Rangement des vêtements	RVP	9																																				



## La partie : Travaux Dirigés

### 1. TD N°1 : Ordonnancement

#### 1.1. Exercice 1

Q1.Déterminer la durée minimale du projet :

Tâche	A	B	C	D	E	F	G	H	I
Antériorités	/	/	A	B	A, B	A	C, D, E	B	F, H
Durée	7	5	6	8	3	5	6	5	4

Q2.Si la tâche F ne peut commencer que 4 jours après le début de A (c'est-à-dire l'antériorité de F est  $A^*+4$ ) et la tâche H ne peut commencer que 5 jours après le fin de B (c'est-à-dire l'antériorité de H est  $B+5$ ). Déterminer la durée minimale du projet.

#### 1.2. Exercice 2

Q1.Déterminer la durée minimale du projet. L'échéancier est le suivant :

Tâche	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
Antériorités	GM	H	EQN	/	H	BP	/	GM	D	AL	/	KO	I	AL	GM	ENQ	H
Durée	7	7	10	3	5	1	5	4	8	12	7	1	6	1	5	7	2

Q2.Si la tâche Q ne peut commencer que 2 jours après le début de H (c'est-à-dire l'antériorité de Q est  $H^*+2$ ). Déterminer la durée minimale du projet.

#### 1.3. Exercice 3

Q1.Déterminer la durée minimale du projet:

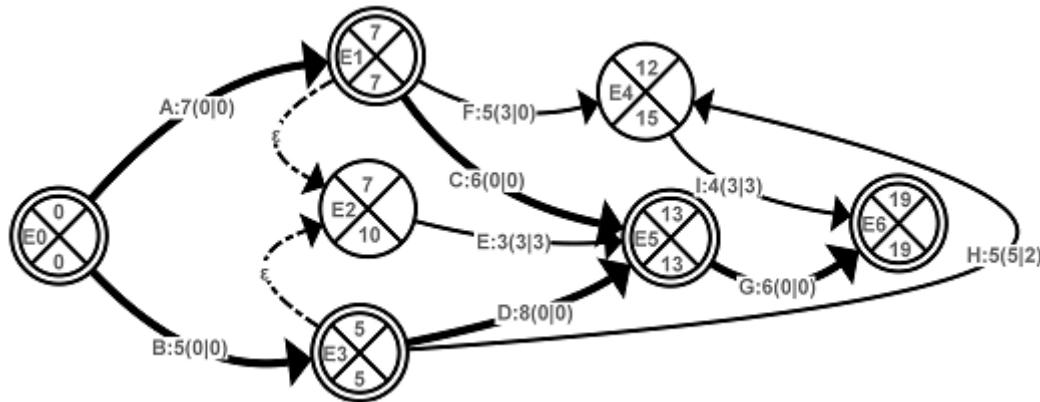
Tâche	A	B	C	D	E	F	G	H	I	J	K	L	M
Antériorités	GM	HI	B	/	H	BL	/	GM	D	AK	/	CE	D
Durée	7	7	2	3	5	1	5	4	8	12	7	7	6

Q2.Si la tâche C ne peut commencer que 2 jours après le début de B (c'est-à-dire l'antériorité de C est  $B^*+2$ ). Déterminer la durée minimale du projet.

## 2. TD N°1 : Ordonnancement Corrections

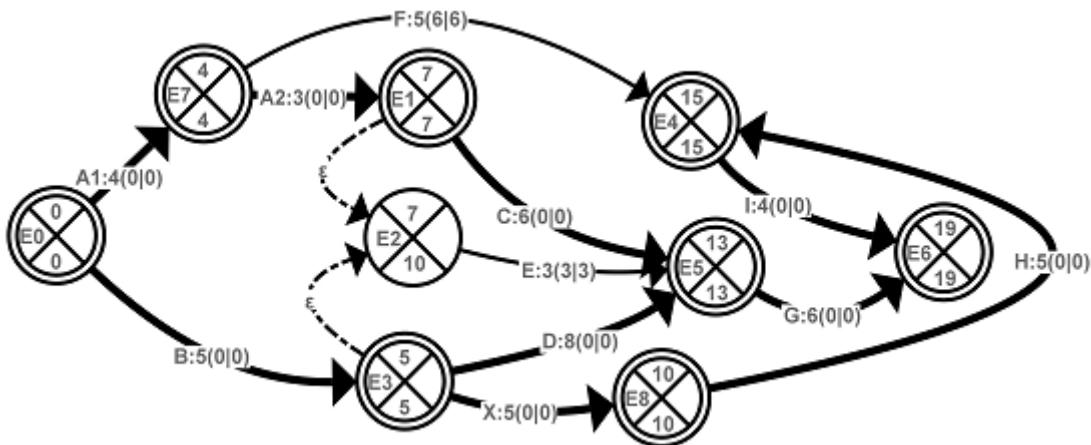
### 2.1. Exercice 1

Q1. Déterminer la durée minimale du projet :



Q2. Si la tâche F ne peut commencer que 4 jours après le début de A (c'est-à-dire l'antériorité de F est  $A^*+4$ ) et la tâche H ne peut commencer que 5 jours après le fin de B (c'est-à-dire l'antériorité de H est  $B+5$ ). Déterminer la durée minimale du projet.

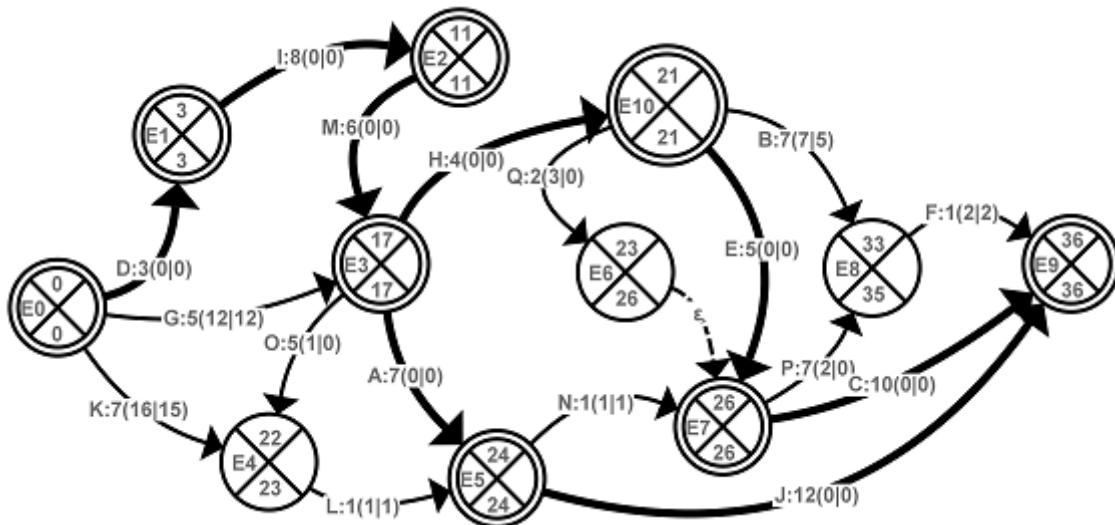
Pour résoudre le problème «  $A^*+4$  », il faut subdiviser la tâche A en 2 sous-tâches :  $A_1$  de durée 4 et  $A_2$  qui dure le reste, soit  $7-4 = 3$ .  $A_1$  sera antérieure à  $A_2$ , ce qui était postérieur à A sera maintenant postérieur à  $A_2$  et F sera postérieure à  $A_1$ . Pour résoudre le problème «  $B+5$  », il faut introduire une tâche X de durée 5 postérieure à B et antérieure à H. Pour résoudre le problème de E, il faut introduire 2 tâches fictives de la fin de  $A_2$  et de B vers un nouveau sommet.



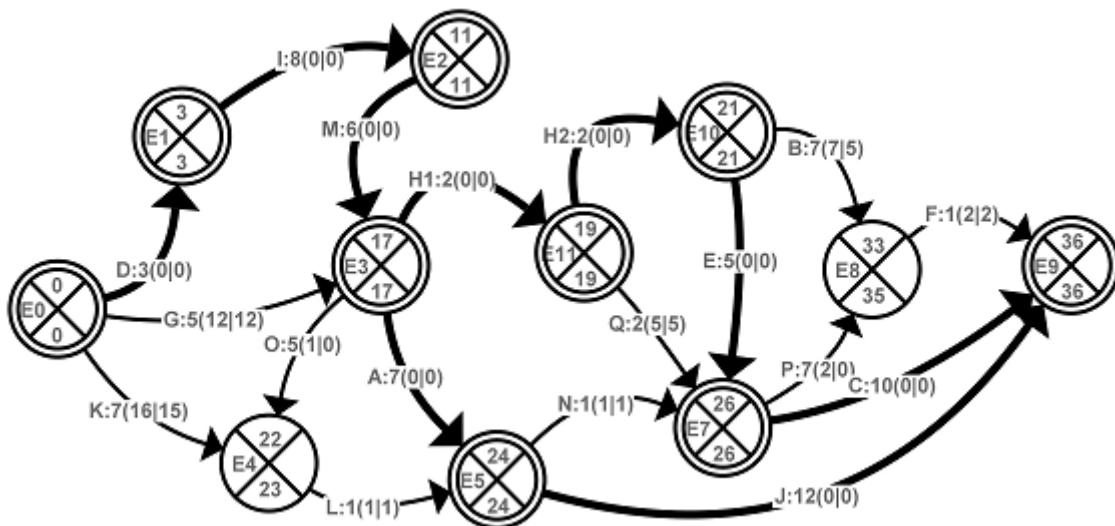
## 2.2. Exercice 2

Q1. Déterminer la durée minimale du projet. L'échéancier est le suivant :

- Les tâches commençantes sont D, G et K
- Les tâches finissantes sont C, F et J
- Les tâches convergentes sont : GM, ENQ, BP, AL, KO
- Les niveaux seront les suivants : DGK I M AHO BEQL JN PC F



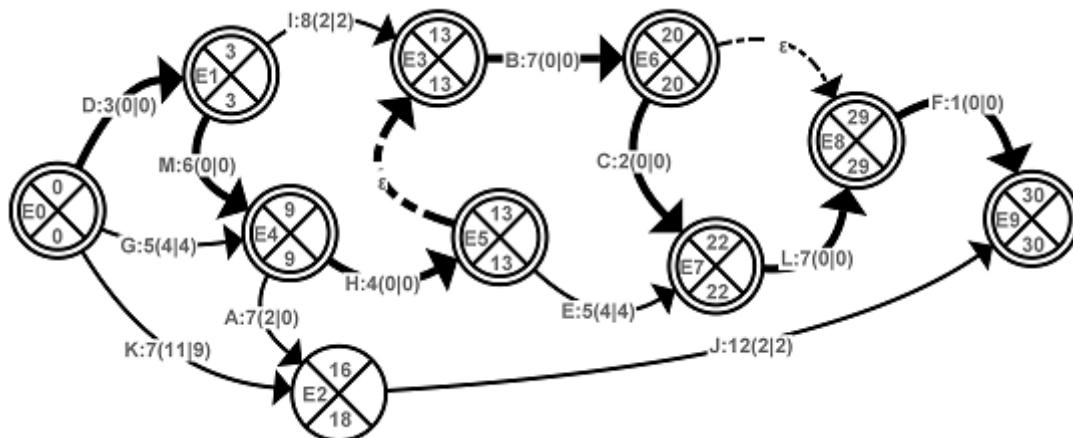
Q2. Si la tâche Q ne peut commencer que 2 jours après le début de H (c'est-à-dire l'antériorité de Q est  $H^*+2$ ). Déterminer la durée minimale du projet.



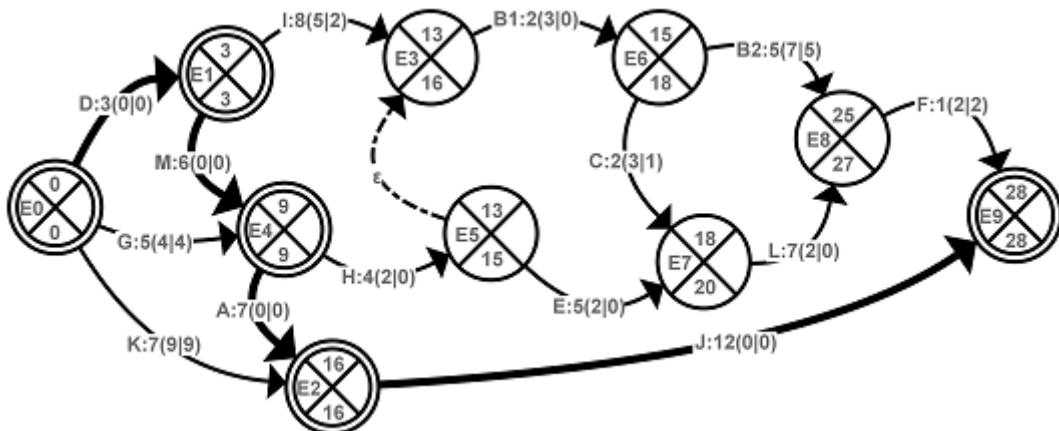
### 2.3. Exercice 3

Q1. Déterminer la durée minimale du projet:

- Les tâches commençantes sont D, G et K
- Les tâches finissantes sont F et J
- Les tâches convergentes sont : G et M, A et K, C et E
- Les niveaux seront les suivants : DGK IM AH EBJ C L F



Q2. Si la tâche C ne peut commencer que 2 jours après le début de B (c'est-à-dire l'antériorité de C est  $B^*+2$ ). Déterminer la durée minimale du projet.





## **La partie : Travaux Pratiques**

### **1. TP N°1**

Texte...

### **2. TP N°2**

Texte...



## Bibliographie

- [1] E Westney, Gestion de petits projets, Techniques de planification, d'estimation et de contrôle, Paris, afnor gestion, 1991.
- [2] Christophe Midler, "L'acteur projet, portrait d'un rôle d'influence", in Actes de la 8e convention de l'AFITEP : Direction et contrôle de projet", Paris, octobre 1992, pp 13-28.
- [3] L. Sayles, M. Chandler, "The project Manager : Organizational Metronome", in Managing Large Systems, New York, Harper & Row, 1971, pp 204-226.
- [4] P.J. Benghozi, Innovations et gestion de projets, Paris, Eyrolles, 1990.
- [5] Vincent Giard, "Les gestions du risque dans les projets", in Actes de la 8e convention de l'AFITEP : Direction et contrôle de projet", Paris, octobre 1992, pp 175-191.
- [6] Vincent Giard, Gestion de projet, Economica, 1992.
- [7] Jean-Louis Muller, "L'estimation : un métier ou un art ?", in Actes de la 8e convention de l'AFITEP : Direction et contrôle de projet", Paris, octobre 1992, pp 175-191.
- [8] Progèspace, "Guide méthodologique de gestion de projet", Support de formation 1990.
- [9] J.A. McCall, Quality factors, in Encyclopædia of Software Engineering, Vol 1, pp 958—969.
- [10] John Wiley& Sons, 1994 T. Force, Qualimétrie des systèmes complexes, mesure de la qualité du logiciel, Les éditions d'organisation.

### Questions (03pt)

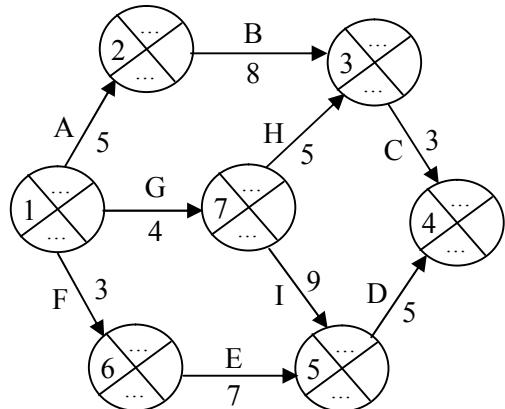
Q1. Quelles sont les enjeux de la gestion de projet ?

Q2. Citez les trois activités principales de la gestion de projet pour assurer une bonne production.

### Exercice 1 (08pt)

Soit le réseau PERT suivant :

1. Déterminez le niveau, l'antériorité et la supériorité de chaque tâche.
2. Calculez les dates au plus tôt et les dates au plus tard.
3. Calculez les marges totales et les marges libres.
4. Tracez le diagramme GANTT en mettant évidence le chemin critique.



### Exercice 2 (04pt)

Soit le tableau d'avancement de projet suivant :

Tâche	Travail estimé	Date de fin estimée	Travail réel	Date de fin effective
1	5	30/01	5	25/01
2	10	28/03	15	10/02
3	20	30/03	20	20/03
4	25	30/05	30	15/06
5	35	30/07		
6	40	30/08		

Q1. Calculez les indicateurs d'avancement au 30/03.

Q2. Si la tâche 5 a également été achevée avec 40 jours/homme le 15/08. Calculez ces indicateurs au 15/07.

### Exercice 3 (05pt)

Soit à estimer la durée de projet présenté par le tableau suivant :

- Calculez le risque, la durée probable, l'écart-type et la variance de chaque tâche.
- Quelle est la durée estimée de projet à la probabilité 0,95 ?
- Quelle est la probabilité d'achèvement de projet en 17 jours ?

Tâche	Antériorités	$T_{opt}$	$T_{pes}$	$T_{vrai}$
T1	/	4	8	6
T2	/	1	5	3
T3	T1	6	8	7
T4	T2	2	8	5
T5	T3	2	6	4
T6	T1, T4	5	7	6

### Questions (03pt)

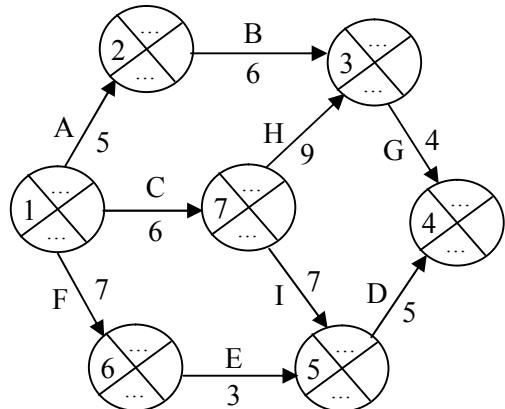
Q1. Quelles sont les enjeux de la gestion de projet ?

Q2. Citez les trois activités principales de la gestion de projet pour assurer une bonne production.

### Exercice 1 (08pt)

Soit le réseau PERT suivant :

5. Déterminez le niveau, l'antériorité et la supériorité de chaque tâche.
6. Calculez les dates au plus tôt et les dates au plus tard.
7. Calculez les marges totales et les marges libres.
8. Tracez le diagramme GANTT en mettant évidence le chemin critique.



### Exercice 2 (04pt)

Soit le tableau d'avancement de projet suivant :

Tâche	Travail estimé	Date de fin estimée	Travail réel	Date de fin effective
1	5	30/01	5	25/01
2	10	28/03	15	10/02
3	15	30/03	15	20/03
4	25	30/05	30	15/06
5	35	30/07		
6	40	30/08		

Q1. Calculez les indicateurs d'avancement au 30/03.

Q2. Si la tâche 5 a également été achevée avec 40 jours/homme le 15/08. Calculez ces indicateurs au 15/07.

### Exercice 3 (05pt)

Soit à estimer la durée de projet présenté par le tableau suivant :

- Calculez le risque, la durée probable, l'écart-type et la variance de chaque tâche.
- Quelle est la durée estimée de projet à la probabilité 0,95 ?
- Quelle est la probabilité d'achèvement de projet en 17 jours ?

Tâche	Antériorités	$T_{opt}$	$T_{pes}$	$T_{vrai}$
T1	/	1	5	3
T2	/	4	8	6
T3	T1	6	8	7
T4	T2	2	6	4
T5	T3	2	8	5
T6	T1, T4	5	7	6

**Questions (03pt)**

Q1. ....

.....

.....

Q2. ....

.....

.....

**Exercice 1 (08pt)**

Tâche	Durée	Rang	Antéri- orités	Supéri- orités	Marge totale	Marge libre
A						
B						
C						
D						
E						
F						
G						
H						
I						

Etape	Date plus tôt	Date plus tard
1		
2		
3		
4		
5		
6		
7		

Tracé du GANTT																				
Jours Taches	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
B	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
C	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
D	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
E	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
F	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
G	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
H	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
I	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

**Exercice 2 (04pt)**

Question	CBA	CBTE	CRTE	CBTP
Q1				
Q2				

**La formule  
de calcul**

Question	= VA	= IPT	= VE	= IC	= VC
Q1					
Q2					

**Exercice 3 (05pt)**

Tâche	Risque(i) = .....	T <sub>prob</sub> (i) = .....	V(i) = .....	E(i) = .....
T1				
T2				
T3				
T4				
T5				
T6				

Chemin	D <sub>est</sub> = .....	V <sub>est</sub> = .....	E <sub>est</sub> = .....	D <sub>95</sub> = .....	La probabilité d'une durée de 17 jours

### Questions (03pt)

Q1. Les enjeux de la gestion de projet sont:

- ✓ Améliorer la qualité ;
- ✓ Diminuer les coûts ;
- ✓ Maitriser les délais.

1,5 pt

Q2. Les trois activités principales de la gestion de projet sont :

Analyser → Organiser → produire → piloter

1,5 pt

↑ (la boucle de rétroaction)

### Exercice 1 (08pt)

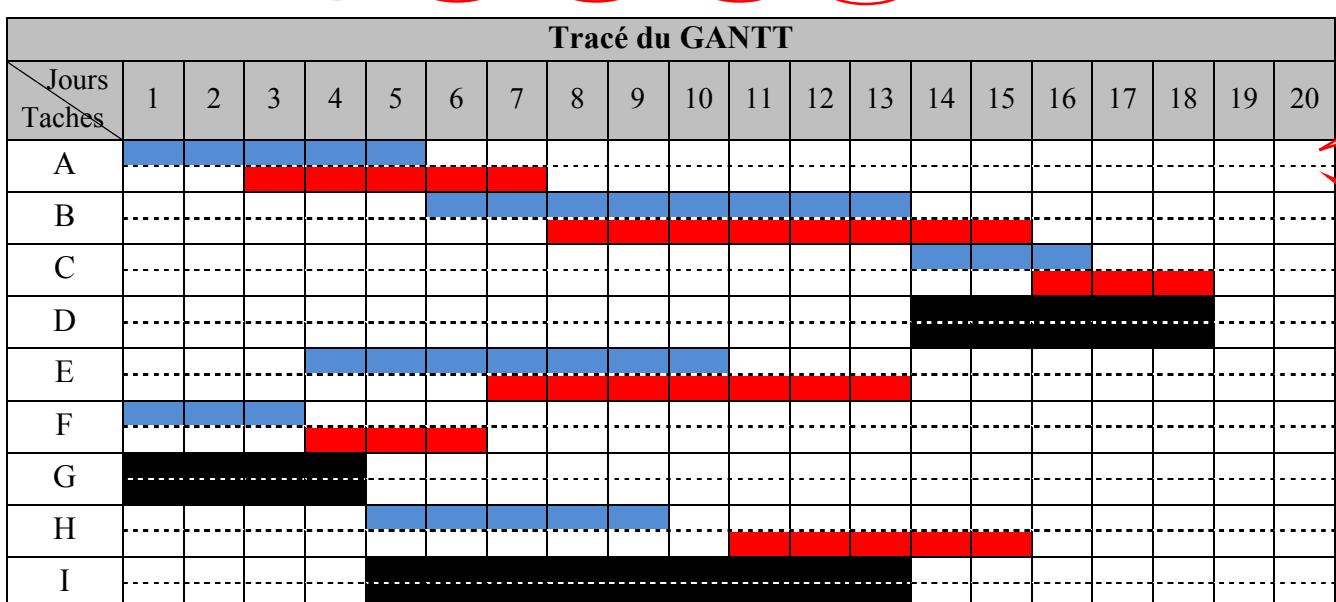
-0,25 pt chaque deux erreurs

Tâche	Durée	Rang	Antéri- orités	Supéri- orités	Marge totale	Marge libre
A	5	1	Rien	B	2	0
B	8	2	A	C	2	0
C	3	3	BH	Rien	2	2
D	5	3	EI	Rien	0	0
E	7	2	F	D	3	3
F	3	1	Rien	E	3	0
G	4	1	Rien	HI	0	0
H	5	2	G	C	6	4
I	9	2	G	D	0	0

Etape	Date plus tôt	Date plus tard
1	0	0
2	5	7
3	13	15
4	18	18
5	13	13
6	3	6
7	4	4

1 pt

1 pt



1 pt

1 pt

**NOTE**

...../20

Nom : .....

Prénom : .....

Groupe : .....

**Exercice 2 (04pt)**

-0,25 pt chaque erreur

Question	CBA	CBTE	CRTE	CBTP
Q1	135	60	70	35
Q2	135	95	110	60

1 pt  
1 pt

Question	VA = CBTE/ CBA	IPT = CBTE/ CBTP	VE = CBTE- CBTP	IC = CBTE/ CRTE	VC = CBTE- CRTE
Q1	44,44%	171,43%	25 jh	85,71%	-10 jh
Q2	70,37%	158,33%	35 jh	86,36%	-15 jh

**Exercice 3 (05pt)**

-0,25 pt chaque deux erreurs

Tâche	Risque(i) = $(T_{pes} - T_{opt}) / T_{pes}$	$T_{prob}(i)$ = $(T_{pes} + T_{opt} + 4 T_{vrai}) / 6$	V(i) = $E(i) * E(i)$	E(i) = $(T_{pes} - T_{opt}) / 6$
T1	0,50	6,00	0,44	0,67
T2	0,80	3,00	0,44	0,67
T3	0,25	7,00	0,11	0,33
T4	0,75	5,00	1,00	1,00
T5	0,67	4,00	0,44	0,67
T6	0,29	6,00	0,11	0,33

0,5 pt

0,5 pt

0,5 pt

0,5 pt

0,5 pt

0,5 pt

Chemin	$D_{est}$ $= \sum_{T_i \in \text{chemin}} T_{prob}(i)$	$V_{est}$ $= \sum_{T_i \in \text{chemin}} V(i)$	$E_{est}$ $= \sqrt{V(i)}$	$D_{95}$ $= D_{est} + 1.65 * E_{est}$	La probabilité d'une durée de 17 jours
T1T3T5	17,00	1,00	1,00	18,65	50,00%
T1T6	12,00	0,56	0,75	13,23	100,00%
T2T4T6	14,00	1,56	1,25	16,06	99,19%

0,5 pt

0,5 pt

0,5 pt

0,5 pt

### Questions (03pt)

Q1. Les enjeux de la gestion de projet sont:

- ✓ Améliorer la qualité ;
- ✓ Diminuer les coûts ;
- ✓ Maitriser les délais.

1,5 pt

Q2. Les trois activités principales de la gestion de projet sont :

Analyser → Organiser → produire → piloter

1,5 pt

↑ (la boucle de rétroaction)

### Exercice 1 (08pt)

-0,25 pt chaque deux erreurs

Tâche	Durée	Rang	Antéri- orités	Supéri- orités	Marge totale	Marge libre
A	5	1	Rien	B	4	0
B	6	2	A	G	4	4
C	6	1	Rien	HI	0	0
D	5	3	EI	Rien	1	1
E	3	2	F	D	4	3
F	7	1	Rien	E	4	0
G	4	3	BH	Rien	0	0
H	9	2	C	G	0	0
I	7	2	C	D	1	0

Etape	Date plus tôt	Date plus tard
1	0	0
2	5	9
3	15	15
4	19	19
5	13	14
6	7	11
7	6	6

1 pt

1 pt

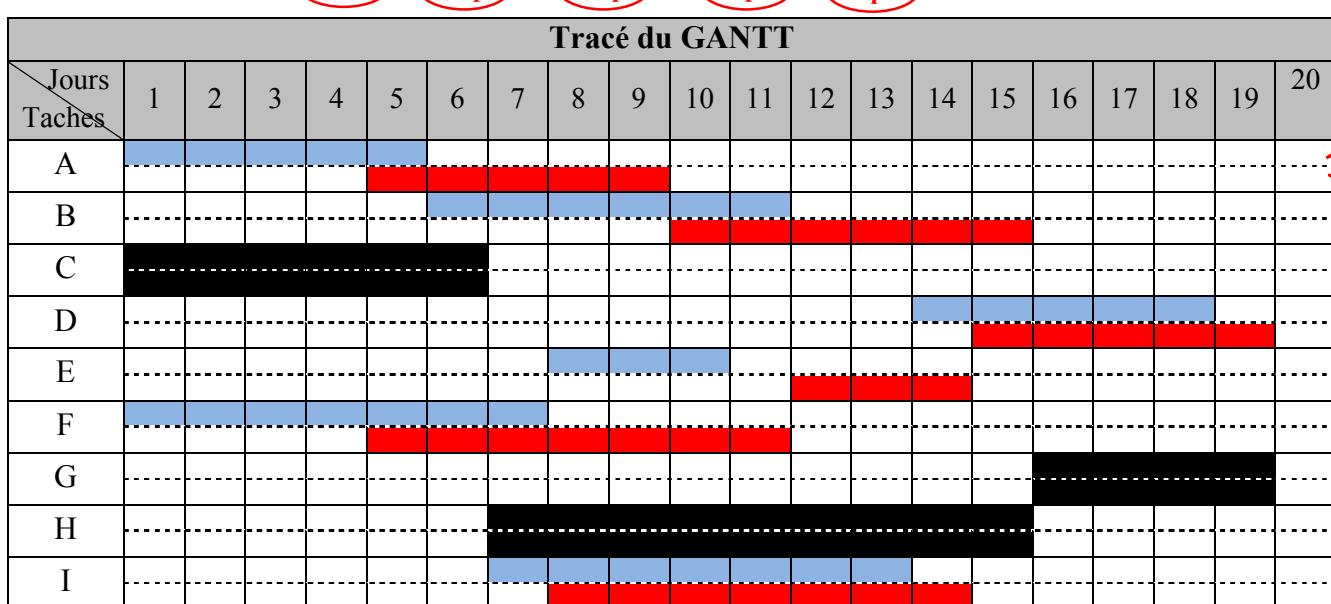
1 pt

0,5 pt

0,5 pt

1 pt

1 pt



1 pt

1 pt

Bon courage

**NOTE**

...../20

Nom : .....

Prénom : .....

Groupe : .....

**Exercice 2 (04pt)**

-0,25 pt chaque erreur

Question	CBA	CBTE	CRTE	CBTP
Q1	130	55	65	30
Q2	130	90	105	55

Question	VA = CBTE/ CBA	IPT = CBTE/ CBTP	VE = CBTE- CBTP	IC = CBTE/ CRTE	VC = CBTE- CRTE
Q1	42,31%	183,33	25 jh	84,62%	-10 jh
Q2	69,23%	163,64%	35 jh	85,71%	-15 jh

**Exercice 3 (05pt)**

-0,25 pt chaque deux erreurs

Tâche	Risque(i) = $(T_{pes} - T_{opt}) / T_{pes}$	$T_{prob}(i)$ = $(T_{pes} + T_{opt} + 4 T_{vrai}) / 6$	V(i) = E(i) * E(i)	E(i) = $(T_{pes} - T_{opt}) / 6$
T1	0,80	3,00	0,44	0,67
T2	0,50	6,00	0,44	0,67
T3	0,25	7,00	0,11	0,33
T4	0,67	4,00	0,44	0,67
T5	0,75	5,00	1,00	1,00
T6	0,29	6,00	0,11	0,33

Chemin	$D_{est}$ $= \sum_{T_i \in \text{chemin}} T_{prob}(i)$	$V_{est}$ $= \sum_{T_i \in \text{chemin}} V(i)$	$E_{est}$ $= \sqrt{V(i)}$	$D_{95}$ $= D_{est} + 1.65 * E_{est}$	La probabilité d'une durée de 17 jours
T1T3T5	15,00	1,56	1,25	17,06	94,56%
T1T6	9,00	0,56	0,75	10,23	100,00%
T2T4T6	16,00	1,00	1,00	17,65	84,13%

0,5 pt

0,5 pt

0,5 pt

0,5 pt

0,5 pt

## La table de loi normale standard