

COURS DE METHODES DE CONDUITE DES PROJETS INFORMATIQUES

Raphael Yende

► To cite this version:

Raphael Yende. COURS DE METHODES DE CONDUITE DES PROJETS INFORMATIQUES.
Licence. Congo-Kinshasa. 2019, 90p. cel-02004689

HAL Id: cel-02004689

<https://hal.archives-ouvertes.fr/cel-02004689>

Submitted on 2 Feb 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Copyright

COURS DE METHODES DE CONDUITE DES PROJETS INFORMATIQUES



YENDE RAPHAEL Grevisse, Ph.D.

Docteur en Télécoms et Réseaux Inf.

Dr. Raphael Grevisse Ph.D.

**Cours dispensé à l'Institut Supérieur de Commerce
en deuxième licence Gestion Informatique.**

©YENDE R.G., 2019

AVERTISSEMENTS

Le support de cours de «*Méthodes de conduite des projets Informatiques* », demande avant tout, un certain entendement des systèmes d'information et des connaissances de base des réseaux informatiques et principalement une prédisposition d'analyse inéluctable et cartésienne. Vu que l'apport de ce cours, met l'accent sur la conception, la conduite et la gestion des projets informatiques en particulier et des systèmes d'information en général reposant sur une compréhension technique approfondie de la gouvernance et de développement des produits logiciels dans les entreprises industrielles et leurs modes de communication courants. Le cours de Méthodes de conduite des projets informatiques se veut pour objectif primordial de renforcer la compétence de l'étudiant de deuxième licence en Gestion informatique, dans le domaine de la conception et de la conduite de projets des systèmes d'information afin de fournir des savoir-faire opérationnels pour toutes les étapes allant de la conception à la finalisation d'un projet informatique.

Ce support de cours est soumis aux droits d'auteur et n'appartient donc pas au domaine public. Sa reproduction est cependant autorisée à condition de respecter les conditions suivantes :

- * Si ce document est reproduit pour les besoins personnels du reproducteur, toute forme de reproduction (*totale ou partielle*) est autorisée à la condition de citer l'auteur.
- * Si ce document est reproduit dans le but d'être distribué à des tierces personnes, il devra être reproduit dans son intégralité sans aucune modification. Cette notice de copyright devra donc être présente. De plus, il ne devra pas être vendu.
- * Cependant, dans le seul cas d'un enseignement gratuit, une participation aux frais de reproduction pourra être demandée, mais elle ne pourra être supérieure au prix du papier et de l'encre composant le document.

Copyright © 2019 Dr. YENDE RAPHAEL; all rights reserved. Toute reproduction sortant du cadre précisé est prohibée.



TABLE DES MATIERES

AVERTISSEMENTS	1
TABLE DES MATIERES.....	2
BIBLIOGRAPHIE	5
INTRODUCTION	6
DEFINITION DES CONCEPTS CLES	7
OBJECTS DU COURS	8
 PREMIER CHAPITRE - GENERALITES SUR LES PROJETS INFORMATIQUES	 9
I.1. DEFINITION ET CONTEXTE D'ETUDES	9
I.2. OBJECTIFS DES PROJETS INFORMATIQUES	12
I.3. LES CARACTERISTIQUES DES PROJETS INFORMATIQUES	15
 DEUXIEME CHAPITRE - LA CONDUITE DU PROCESSUS DES PROJETS INFORMATIQUES.....	 19
II.1. LES ACTIVITÉS DE GESTION DE PROJET	19
II.2. PRINCIPAUX ACTEURS DANS UN PROJET INFORMATIQUE.....	22
II.2.1. LE MAITRE D'OUVRAGE.....	22
II.2.2. LE MAITRE D'OEUVRE	23
II.2.3. L'EQUIPE DU PROJET	23
II.2.4. LES UTILISATEURS.....	24
II.3. NIVEAUX DE RESPONSABILITÉ DANS LE PROJET INFORMATIQUE	26
II.4. ROLES DE CHEF DE PROJET.....	28
II.5. L'ADMINISTRATION DU CYCLE DE PROJETS INFORMATIQUES.....	30
 TROISIEME CHAPITRE - DYNAMIQUE DU PROCESSUS DE DEVELOPPEMENT DE PROJETS INFORMATIQUES.....	 33
III.1. DEFINITION ET CONTEXTE D'ETUDES	33
III.2. DECOUPAGE DE PROJETS INFORMATIQUES.....	33
III.3. LES MODELES DE DEVELOPPEMENT	36
III.3.1. LE MODELE DU CODE-AND-FIX.....	36
III.3.2. LE MODELE DE TRANSFORMATION AUTOMATIQUE.....	37
III.3.3. LE MODELE DE LA CASCADE	37
III.3.4. LE MODELE EN V	38

III.3.5. LE MODELE EN W	39
III.3.6. LE MODEL EVOLUTIF	39
III.3.6. LE MODEL EN SPIRALE	40
III.4. LA GESTION D'UNE PHASE D'UN PROJET INFORMATIQUE.....	41
III.5. LA CAPITALISATION DE L'EXPERIENCE D'UN PROJET INFORMATIQUE.....	42
III.6. LES PRINCIPALES MODIFICATIONS DE CONDUITE DE PROJETS INFORMATIQUES.....	44

QUATRIEME CHAPITRE - LES OUTILS DE GESTION DE PROJETS INFORMATIQUES..... 48

IV.1. LES METHODES DE DEVELOPPEMENT	48
IV.1.1. LES METHODES FONCTIONNELLES.....	49
IV.1.2. LES METHODES OBJET	50
IV.1.3. LES METHODES ADAPTATIVES	51
IV.1.3.1. LES METHODES PREDICTIVES.....	51
IV.1.3.2. LES METHODES AGILES	51
IV.1.3.2.1. LA METHODE RAD.....	52
IV.1.3.2.2. LA METHODE DSDM	53
IV.1.3.2.3. LE MODELE XP	54
IV.1.3.2.4. LA METHODE SCRUM.....	55
IV.1.4. LES METHODES SPECIFIQUES	56
V.1.4.1. LE CYCLE ERP	56
V.1.4.2. LE MODELE RUP	57

IV.2. LES TECHNIQUES DE GESTION.....	58
IV.2.1. LES TECHNIQUES DE PLANIFICATION OU DE L'ORDONNANCEMENT	58
IV.2.1.1. LE DIAGRAMME DE GANTT	59
IV.2.1.2. LES RESEAUX PERT	62
IV.2.1.3. LA MPM (METHODE DE POTENTIELS METRA).....	69
IV.2.2. LES TECHNIQUES D'ESTIMATION DE PROJETS INFORMATIQUES..	72
IV.2.2.1. ESTIMATION DE CHARGES DE PROJETS INFORMATIQUES	72
A. ESTIMATION DE LA TAILLE	73
B. ESTIMATION DE LA CHARGE	73
C. ESTIMATION DES DELAIS.....	74

D. ESTIMATION DU COUT	75
E. ESTIMER A PARTIR DES DELAIS IMPOSES	76
IV.2.2.2. EXACTITUDE ET PRECISION D'UNE ESTIMATION	77
IV.2.2.3. COMPRENDRE LES ARBITRAGES.....	78
IV.2.2.4. LES DIFFICULTES DES ESTIMATIONS	81
IV.2.2.5. PROJETS DE MAINTENANCE ET D'EVOLUTION, COMPARES AU NOUVEAU DEVELOPPEMENT.....	82
IV.2.2.6. ESTIMATION DES PETITS PROJETS.....	83
IV.2.2.7. ESTIMATION DES PROJETS DANS DE NOUVEAUX DOMAINES	83
IV.2.2.8. CONSEILS PRATIQUES POUR L'ESTIMATION DES PROJETS INFORMATIQUES.....	84
IV.2.2.9. LES OUTILS D'AIDE A L'ESTIMATION DES PROJETS INFORMATIQUES.....	85
CONCLUSION	89

BIBLIOGRAPHIE

- **AFITEP**(2000), « *Dictionnaire de management de projet* », 4^e édition, Afnor.
- **AFNOR** (1998), « *Management de projet* », éd. AFNOR, 854p.
- **BENNATAN E.M.** (1995), « *Management des projets informatiques*», Afnor.
- **CADLES James et YEATES Donald**,(2001), « *Project Management for Information System* », 3^e édition, Prentice Hall.
- **Christophe MIDLER**(1994), « *L'auto qui n'existait pas* », éd. InterEditions.
- **GOETHER Wolfhart B., BAILEY Elizabeth K., BUSBY Mary B.**, *Software Effort and Schedule Measurment,: A framework for counting Staff-hours and reporting Schedule Information*, CMU/SEI-92-TR-021, 1992,
- **HP MADERS, ET GAUTHIER** (1998), « *Conduire un projet d'organisation* » éd. Editions d'Organisation, 301p.
- **HUMPHREY Watts**, *A Discipline for Software Engineering*, Addison-Wesley, 1995
- **KEZSBOM D.S., SCHILLING D. Let EDWARD K. A**(1989),« *Dynamic Project Management. A practical Guide for Managers and Engineers*», John Wiley & Sons.
- **M. Gedin, H. Tardieu, A. Rochfeld et R. Coletti**; « *MCP, Méthode de conduite des projets informatiques* », éd. Organization, 1983
- **MCCONNELL Steve**, *Rapid Development - Taming Wild Software Schedules*, Microsoft Press, 1996
- **MORLEY Chantal**(2004), « *Management d'un projet système d'information* », 4^e édition, Dunod.
- **R. PARK**, *Software Size Measurement: A framework for counying source statements*, CMU/SEI-92-TR-020, 1992,
- **SYMONS Charles**, *Software Sizing and Estimating: Mark II Function Point Analysis*, John Wiley, 1991
- **Thierry PICQ**(1999), « *Manager une équipe projet* », éd. DUNOD, 235p.
- **VERMA V.K.**(2003), « *Guide du référentiel des connaissances en gestion de projet (Guide PMBOK)* », PMI.

INTRODUCTION

Le monde du travail s'est vigoureusement réformé ces dernières années. Citons pour faire court trois dispositions lourdes : d'abord, **la propagation du fonctionnement en mode projet** (forme d'organisation dans laquelle chaque salarié devient un « entrepreneur » qui se voit assigner un objectif et la responsabilité de l'atteinte de cet objectif) ; ensuite, **le lieu de l'exigence d'instantanéité** (la réponse à une question du client ou du supérieur hiérarchique qui ne peut être qu'immédiate) ; Enfin **l'intrusion de l'informatique et de l'internet** qui, bien utilisés, permettent cette instantanéité.

Dans cette vie, Nous avons tous des projets : Qu'ils soient d'ordre privé ou professionnel, ils donnent du sens à notre vie, et nous projettent vers un futur que nous voulons meilleur ... Un projet, au-delà de la part de rêve qu'il contient, appelle à la réalisation, à la concrétisation de l'idée de départ. Mais comme le dit cet adage bien connu « *l'intention ne vaut pas l'action* ». Il faut donc se donner les moyens de mener à bien une démarche plus ou moins compliquée pour atteindre l'objectif du projet.

Les projets sont partout (*la vie quotidienne, société, etc.*) et nous sommes tous des chefs des projets qui s'ignorent, cependant la conduite d'un projet exige plusieurs facteurs :

- Une bonne dose d'imagination et de créativité (*osez*) ;
- Une réelle ouverture d'esprit et beaucoup d'écoute.
- De l'audace et du réalisme (*une prise de risque raisonnée*) ;
- Une détermination sans faille (*croyez en vous et en votre projet*) ;
- Une patience à toute épreuve ;
- Du travail, encore du travail ;
- Un questionnement critique permanent ;
- Ce qu'il faut de méthode et d'organisation.

DEFINITION DES CONCEPTS CLES

- ✎ **Le maître d'ouvrage** : est la personne physique ou morale qui est propriétaire de l'ouvrage (Projet). Il fixe les objectifs, l'enveloppe budgétaire et les délais souhaités pour le projet. C'est le bailleur de fonds.
- ✎ **Le maître d'œuvre** : est la personne physique ou morale qui reçoit la mission du maître d'ouvrage pour assurer la conception et le contrôle de la réalisation d'un ouvrage conformément au programme préétabli. C'est l'organisation ou l'entreprise qui exécute le projet.
- ✎ **Le Projet** : ensemble d'activités regroupant les trois caractéristiques suivantes : Il est entrepris pour satisfaire un besoin spécifique, sa durée et les moyens accordés sont limités et enfin il aboutit à un résultat unique censé satisfaire le besoin.
- ✎ **Un système** est un ensemble d'*éléments* interagissant entre eux suivant un certains nombres de principes et de *règles* dans le but de réaliser un *objectif*.
- ✎ **Un modèle** : est une représentation schématique de la réalité.
- ✎ **Une base de Données**: ensemble des données (*de l'organisation*) structurées et liées entre elles : stocké sur support à accès direct (*disque magnétique*) ; géré par un SGBD (*Système de Gestion de Bases de Données*), et accessible par un ensemble d'applications.
- ✎ **Une analyse** : c'est un processus d'examen de l'existant
- ✎ Une **Conception** : est un processus de définition de la future application informatique.
- ✎ **Un système d'Information** : ensemble des moyens (*humains et matériels*) et des méthodes se rapportant au traitement de l'information d'une organisation.

OBJECTS DU COURS

Ce cours a pour objectif principal, d'acquérir et de renforcer la compétence dans le domaine de la conception et de la conduite de projets des systèmes d'information afin de fournir des savoir-faire opérationnels pour toutes les étapes allant de la conception à la finalisation d'un projet informatique.

Et, D'une façon spécifique ce cours vise à :

- Comprendre et concevoir un projet informatique dans un environnement économique et stratégique complexe ;
- Quantifier les ressources en temps, en matériels, en moyens humains et en financements qui sont nécessaires à un projet informatique;
- Anticiper les évolutions technologiques et de gérer des difficultés techniques ou managériales imprévues ainsi que la qualité et la sécurité dans un projet informatique ;
- Analyser la structure d'une organisation et le rôle de chaque acteurs du projet afin de se positionner et de mettre en œuvre des dynamiques collaboratives complexes (*comprendre et être capable de mettre en place un système de confiance et de coopération entre les différents acteurs d'un projet*) ;
- Négocier et de conduire les processus de changement et d'innovation ;
- Analyser et de négocier la gestion des SI dans l'ensemble des stratégies de l'entreprise.

YENDE RAPHAEL Grevisse, PhD.
Professeur associé

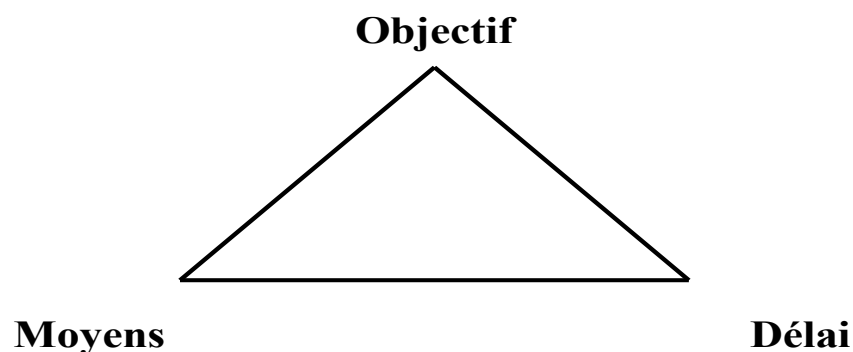
PREMIER CHAPITRE - GENERALITES SUR LES PROJETS INFORMATIQUES

I.1. DEFINITION ET CONTEXTE D'ETUDES

Du mot latin « *projectum* » de « *projicere* », en résumé, Ainsi, le mot « *projet* » voulait initialement dire « *quelque chose qui vient avant que le reste ne soit fait* » ou « *jeter quelque chose vers l'avant* » ; c'est ainsi, le projet peut être défini comme étant un ensemble d'activités qui sont prises en charge, dans un délai donné et dans les limites de ressources imparties, par des personnes qui y sont affectées dans le but d'atteindre des objectifs définis. Cependant, le projet peut être appréhendé de plusieurs et selon différentes organisations :

- Selon la norme ISO 10006 (*version 2003*) « un projet est un processus unique qui consiste en un **ensemble d'activités coordonnées** et maîtrisées, comportant des dates de début et de fin, entrepris dans le but d'atteindre un **objectif** conforme à des exigences spécifiques, incluant des **contraintes** de *délais*, de *coûts* et de *ressources* ». La définition d'un projet comporte deux notions clés : ***le projet est unique et le projet est temporaire.***
- Selon Le PMBOK, référentiel du PMI, considère un projet comme une « *entreprise temporaire décidée pour obtenir un produit ou un service unique* ». L'unicité du produit entraîne l'unité des activités à mettre en œuvre.
- Selon L'AFITEP définit 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* » ; et introduit une distinction entre les projets d'ingénierie qui visent l'obtention d'un résultat pour un client, et les projets produit débouchant sur un modèle qui fera ensuite l'objet d'une fabrication répétitive.

L'unicité du processus projet doit être comprise de deux façons. D'une part, les activités qui permettront d'atteindre l'objectif sont définies de façon à prendre en compte les particularités de chaque projet, même si l'on réutilise des trames générales. D'autre part, ces activités ne seront exécutées qu'une seule fois. Il y a donc unicité au niveau du type et au niveau de l'instance. Ainsi, un projet peut être symbolisé par un triangle dit « *triangle projet* » : on est en mode projet lorsque l'on doit atteindre un objectif avec des moyens ad hoc et dans un délai donné :



L'AFNOR donne les deux définitions suivantes¹ :

- « *un projet est déterminé et mis en œuvre pour élaborer la réponse au besoin d'un utilisateur, d'un client ou d'une clientèle et il implique un objectif et des actions à entreprendre avec des ressources précises* »
- « *un projet est une démarche spécifique qui permet de structurer méthodiquement et progressivement une réalité à venir* »

A partir de celles-ci, nous voyons qu'un projet est :

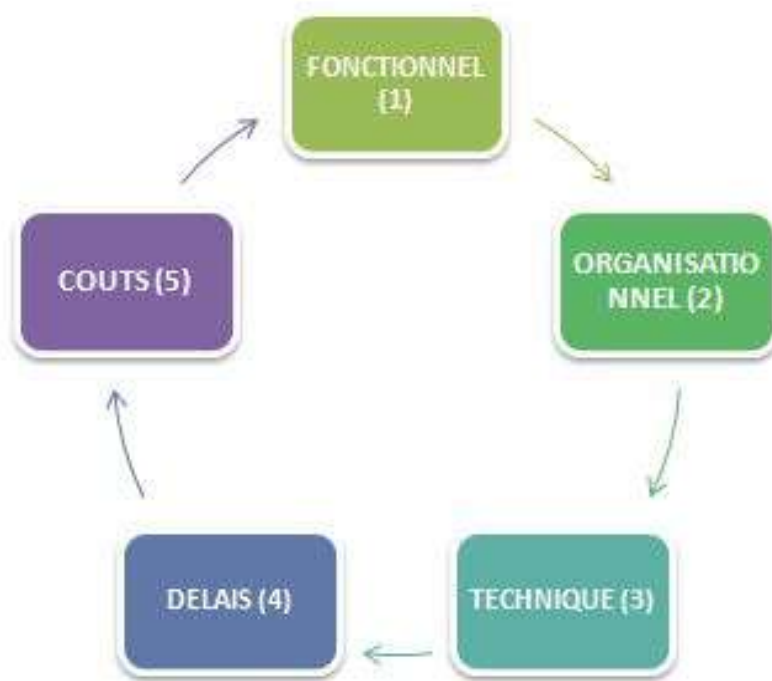
- ➔ Composé d'un **ensemble d'activités** : Il s'agit donc d'un processus qui doit être géré du point de vue de sa conduite, des moyens mis en œuvre et résultats attendus ;
- ➔ Mis en œuvre en vue d'**atteindre un objectif précis** : Celui-ci doit être clairement exprimé par le (système) client. Cet objectif présente un caractère novateur et n'est pas répétitif. Il est la plupart du temps matériel, mais il peut également porter sur des éléments culturels, sur des valeurs à faire évoluer ou à introduire.
- ➔ Réalisé dans un **délai donné** : Il est limité dans le temps avec des dates de début et de fin, « *un projet n'a pas d'avenir, il a une fin* » C. MIDLER ;
- ➔ Exécuté grâce à un ensemble de **moyens** : qui sont matériels et humains ; Pour P. ZARIFIAN, l'organisation en projet réunit « *une équipe multi-métiers autour d'un projet d'innovation avec des objectifs précis et une durée de vie bien spécifiée ... Les gens travaillent ensemble sur un projet précis et pour une durée limitée* »².

¹AFNOR, « *Management de projet : Qualité et efficacité des organisations* », éd AFNOR, 1995.

²P. ZARIFIAN, « *Objectif compétence* », éd. Liaison col. Entreprise et carrière, 1999

Tout projet informatique digne de ce nom doit répondre à cinq (5) prérogatives (atouts ou privilèges) principales :

- **Fonctionnel** : il doit répondre à un besoin défini par le client;
- **Technique** : il doit respecter des spécifications et des contraintes ;
- **Organisationnel** : il doit respecter un mode de fonctionnement (*rôles, culture, fonctions, résistance au changement*) ;
- **Délais** : il doit respecter des échéances (planning) ;
- **Coûts** : il doit respecter le budget prévu pour le projet.



Le projet est constitué des actions ou des interventions que fait quelqu'un dans la plupart de temps une personne morale pour atteindre des résultats utiles. *Le projet implique que ces personnes s'organisent pour le faire. L'organisation qu'elles mettent en place n'est pas le projet mais cette organisation fait le projet.* Le but du projet est de produire de l'utilité de façon économe.

I.2. OBJECTIFS DES PROJETS INFORMATIQUES

L'objectif visé par la réalisation d'un projet informatique est souvent appelé « *produit* », même lorsqu'il ne présente pas un caractère matériel. On distingue ainsi le contenu du produit (*les spécifications*) et le contenu du projet (*le travail à réaliser pour livrer le produit*). Le premier détermine le second.

Le contenu du produit est en général précisé au cours de l'avancement du projet. Il fait l'objet d'une première description, qui est ensuite traduite en « *produits livrables* ». Le cahier des charges contient la liste des produits livrables. Le contenu du projet dépend du contenu du produit. Il contient les tâches orientées produit, ainsi que les tâches de gestion de projet. On opère en général un regroupement des tâches en sous-ensembles aboutissant à la production des différents livrables. C'est ce qu'on appelle l'organigramme des tâches (*OT*), qui décrit la totalité du travail à effectuer : les tâches ne figurant pas sur l'organigramme des tâches se situent en dehors du contenu du projet. Les outils et méthodes utilisés pour réaliser le produit varient selon le domaine d'application, alors que ceux qui permettent la gestion du projet sont en grande partie partagés par tous les projets.

Comprendre les objectifs du projet et faire émerger des réponses adéquates est de la responsabilité du chef de projet. On rencontre souvent les grandes catégories suivantes, qui auront des conséquences sur le management du projet. Nous allons les esquisser :

- *Productivité administrative* : la rentabilité du capital investi est recherchée dans la diminution de main d'œuvre grâce à l'automatisation d'une partie des tâches. Le climat social sera tendu et la gestion du changement difficile à mener. La participation des utilisateurs peut conduire à un blocage du projet.
- *Aide au management* : l'objectif majeur du projet est l'amélioration des prises de décision au moyen d'un observatoire au service du management. On va bâtir 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, assurer une veille concurrentielle. La conception du système doit être très proche des gestionnaires, faute de quoi le système ne sera guère utilisé.
- *Efficacité opérationnelle* : on attend un meilleur fonctionnement opérationnel par un usage créatif des technologies de l'information et de la communication. L'analyse et la reconstruction des processus sont déterminantes, mais la gestion du changement est un enjeu essentiel.

- *Évolutivité* : on cherche à obtenir un système flexible pouvant être modifié rapidement en cas d'évolution des contraintes et/ou de la stratégie et sachant prendre en compte des adaptations ou des personnalisations non encore identifiées au moment du projet. Cet objectif s'inscrit dans une meilleure maîtrise des investissements informatiques. La compréhension du domaine et de son évolution est importante.
- *Utilisation d'une nouvelle technologie* : le but principal du projet est d'expérimenter une nouvelle technologie, pour voir ce que l'on peut en tirer ou pour obtenir un « effet vitrine » vis-à-vis de l'extérieur. Un délai court est un élément essentiel de la réussite du projet.

Le choix de démarrer un projet est fondé sur *le gain* que l'on en attend. Ce gain se mesure à l'aide du **ROI (Ret Our Investissement)**. Dans la langue de Shakespeare, on parle de « *Business Case* », en terme plus simple, ce sont les objectifs économiques. Les objectifs *définissent le périmètre du projet*. Les modifier revient à signer l'arrêt de mort du projet. Il convient de ne pas confondre les objectifs et le cahier des charges qui découlent des objectifs du projet. Ces objectifs sont :

- La définition des démarches d'amélioration fonctionnelle,
- la mise en route d'un nouveau produit,
- la croissance externe de l'entreprise,
- le nouveau besoin en termes d'organisation,
- la productivité attendue.

La vision entre le centre de coût ne permet toutefois pas bien d'appréhender tous les avantages d'un projet informatique. La productivité issue de la mise en œuvre d'une solution informatique est difficilement mesurable. C'est le paradoxe de Solow. Les objectifs doivent intégrer deux contraintes : « *celle du coût et celle du délai* ». Il faut aussi prendre en compte le risque dans la mesure où l'activité humaine s'appuie toujours sur un environnement incertain. Les projets informatiques supposent à :

- l'adhésion libre et volontaire d'un ensemble des cadres du projet ;
- La définition et la compréhension explicite des objectifs du projet (*chaque doit comprendre l'intérêt qu'il a d'y participer afin que l'objectif global soit atteint*);
- L'organisation responsable de la conduite du projet (*l'ensemble des opérateurs et bénéficiaires doivent s'assurer, en cours d'exécution, que le projet est constamment géré en fonction de la poursuite des objectifs acceptés*) ;

- L'appréciation et la mesure des avantages d'un projet (*la valeur du produit du projet ne peut pas se mesurer par un prix sur un marché*) ;
- Du fait que le projet représente un ensemble d'opérateurs investisseurs de leur propre efforts, son analyse en terme de comparaison des avantages se fait de façon double : du point de vue du projet dans son ensemble et du point de vue des participants. Ce n'est que dans la mesure où la mesure où l'ensemble projet est lui-même rentable qui sera intéressant de l'entreprendre du point de vue des participants, surtout dans la mobilisation des ressources ;
- La technique de comparaison la plus usuelle entre les coûts et les avantages d'un projet est celle du taux de rentabilité interne, financier ou économique (TRI) ;
- Tout projet informatique qu'il soit marchand ou non doit être rentable. C'est-à-dire ce qu'il rapporte doit être supérieur à ce qu'il coûte.

L'objectif doit être précisé de façon claire, chiffrée et datée. Le résultat doit être conforme à des normes de qualité et de performances prédéfinies, pour le moindre coût et dans le meilleur délai possible :

- ✗ **C**adré (spécifique, précis, défini)
- ✗ **A**pprouvé, consensuel
- ✗ **M**esurable, critère de validation donné
- ✗ **É**chéance précise dans le temps
- ✗ **R**éaliste et faisable...
- ✗ **A**mbitieux.

Le projet est un objectif «extraordinaire»(*au sens littéral du mot*) qui combine quatre aspects :

I.3. LES CARACTERISTIQUES DES PROJETS INFORMATIQUES

Un projet se caractérise par un ensemble de déterminants. Ils sont de nature à préciser :

1. La nature du projet³ : Un système d'information présente souvent un degré élevé de complexité, dans la mesure où c'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* »⁴. De plus, son caractère largement immatériel augmente la difficulté quand on veut en obtenir une description a priori. Cela n'est pas sans conséquence sur les projets, qui doivent prendre en compte ses diverses dimensions.

Le triplet (*objectif, moyens, délai*) présente ainsi, dans le domaine système d'information, trois caractéristiques particulièrement marquées. D'abord, il y a interaction entre l'objectif d'une part et les moyens/délais d'autre part. L'identification de l'objectif conduit à évaluer la charge de travail. Cela permet de décider d'une échéance cible théorique et des moyens à affecter. Si d'autres contraintes obligent à limiter le délai ou le budget, on ajuste l'objectif, par exemple selon le principe du « *design-to-cost* » (*conception contrainte par le budget disponible*) ou du « *time-boxing* » (*conception et développement contraints par l'enveloppe temps*).

Ensuite, l'objectif du projet n'est parfaitement défini qu'à la fin. La plasticité des technologies de l'information ouvre la porte à une multitude de choix, aussi bien pour le logiciel que pour l'organisation qui l'accompagne. Une description exhaustive des fonctions et des rôles est longue et coûteuse. Les modèles n'en donnent qu'une vue partielle, ce qui explique qu'au fur et à mesure du projet, des options puissent être modifiées. Enfin, le développement d'un système d'information se déroule en général dans une Organisation, certains acteurs du projet étant directement concernés par le futur système d'information. La répartition du pouvoir et des ressources, la division du travail, les modes de coordination, les procédures opératoires, les statuts... peuvent être modifiés. Les acteurs ne forment pas un groupe uni vers la réalisation d'un même objectif et certains peuvent développer des stratégies individuelles ou de sous-groupes, pour soutenir ou contrer le projet. Ces caractéristiques font que les projets de système d'information sont particulièrement risqués.

³La nature d'un projet fait appel à la complexité de plusieurs éléments de ce dernier : (**ressources, moyens et compétences**) qui ne sont pas généralement placées sous la responsabilité d'une même autorité et qu'il va falloir coordonner afin que les actions concourent ensemble à l'atteinte d'un même objectif.

⁴R.Reix, « Systèmes d'information et management des organisations », 4^e éd., Vuibert, 2002.

2. La délimitation d'un projet : Tout projet doit forcément avoir un début et une fin, ce qui est déterminé par l'expression du besoin normalement formalisée par un cahier des charges et/ou un contrat.

3. Les risques d'un projet⁵ : Le risque est défini comme « *la possibilité qu'un projet ne s'exécute pas conformément aux prévisions de dates d'achèvement, de coût et de spécifications, ces écarts par rapport aux prévisions étant considérés comme difficilement acceptables voire inacceptables* »⁶. La réalisation des risques peut porter sur les trois sommets du triangle projet : **objectif non atteint, délai non respecté, surconsommation de moyens**. On a longtemps cru que le respect de dispositions méthodologiques a priori permettrait de mener les projets informatiques de gestion avec succès.

C'était mal comprendre les particularités de ces projets. Les trois principales sources d'échec sont **la définition des besoins, l'estimation des charges et la possibilité de nombreux aléas** dans le déroulement du projet. En effet, les objectifs peuvent être imprécis et le pilote du projet doit veiller à une clarification progressive. Par ailleurs, l'estimation des charges n'est pas une science exacte : les écarts de productivité individuelle sont parfois importants et de nombreux facteurs peuvent conduire à un dépassement de charge. Divers imprévus peuvent également avoir une incidence sur le déroulement du projet. Pour repérer et prévenir ces difficultés, deux approches d'analyse des risques ont été proposées. L'une est générale à tous les projets, l'autre est spécifique des projets système d'information. L'idée sous-jacente à l'approche généralisée est qu'en général le/la chef de projet connaît bien les risques, mais qu'il/elle a besoin d'un cadre rigoureux pour s'obliger à les gérer. La démarche proposée comprend cinq étapes :

- Identification des risques ;
- Évaluation d'impact sur les coûts et le délai ;
- Définition d'actions de réduction des risques inacceptables ;
- Suivi des actions ;
- Capitalisation d'expérience.

Des techniques classiques de stimulation et d'organisation des idées sont utilisées au cours des trois premières étapes, telles que les remue-méninges, le diagramme causes-effets dit d'Ishikawa, la matrice dite de Pareto permettant un classement en quatre catégories depuis le risque « mineur » (*probabilité faible, conséquences peu importantes pour le projet*) jusqu'au risque « inacceptable » (*probabilité élevée, conséquences graves*).

⁵Le risque d'un projet fait appel au **non-conformisme**(unicité) de ce dernier, c'est-à-dire qu'il n'y aura jamais deux projets identiques, bien qu'ils peuvent se ressembler du point de vue l'objectif à atteindre, tout comme l'objet et le résultat du projet, mais il y aura toujours une différence plus ou moins importante du point de vue de l'environnement, des bénéficiaires avec leur culture,...

⁶AFITEP, 2000.

4. La gestion du changement⁷ : La gestion du changement implique la planification et la réalisation d'un ensemble d'activités, On identifie classiquement : « *la communication, la formation, la documentation, l'organisation des sites, la migration, l'expérimentation* ». Le recours à de telles activités est en partie lié aux réticences anticipées. Le concept de résistance au changement a parfois été présenté comme un facteur explicatif de l'échec de certains systèmes d'information (*projets informatiques*), les utilisateurs préférant la stabilité et refusant a priori toute évolution. Il est préférable de considérer la résistance comme un symptôme, pouvant se manifester dès le début du projet jusque dans le fonctionnement opérationnel du nouveau système, et de s'interroger sur ses origines, qui relèvent souvent de plusieurs facteurs :

- ***La résistance peut être due au système*** : il s'agit d'une problématique de « *besoins* ». Le système, en particulier le système informatique, apparaît insatisfaisant en termes techniques (*temps de réponses, bugs...*), fonctionnels (*fonctions inadéquates, informations manquantes, erreurs, tâches lourdes...*) ou ergonomiques (*interfaces inadaptées au contexte métier, apprentissage difficile...*). Pour éviter ce type de résistance, des actions sont mises en œuvre tout au long du projet pour s'assurer de l'adéquation du futur système : participation des utilisateurs à l'analyse et à la conception, expérimentation sur des sites pilotes, planification de la migration de l'ancien vers le nouveau système.
- ***La résistance peut être liée aux acteurs eux-mêmes*** : il s'agit d'une problématique d'attitude individuelle. Deux interprétations ont été données. La première considère que, face au changement, il existe des comportements différents selon le profil psychologique (*innovateur ou conservateur*). Il s'agit alors de rassurer et préparer ces utilisateurs, par des actions de communication, de formation et d'accompagnement. La seconde estime que les différences d'attitude sont contingentes à une situation donnée, c'est-à-dire que la résistance ou l'engagement dépendent de la perception par l'individu du changement proposé, notamment la facilité d'utilisation et l'utilité du futur système. Il s'agit de faire évoluer les perceptions, par la formation et la documentation pour agir sur la facilité perçue, et par la communication et la participation à la conception pour développer la perception de l'utilité.

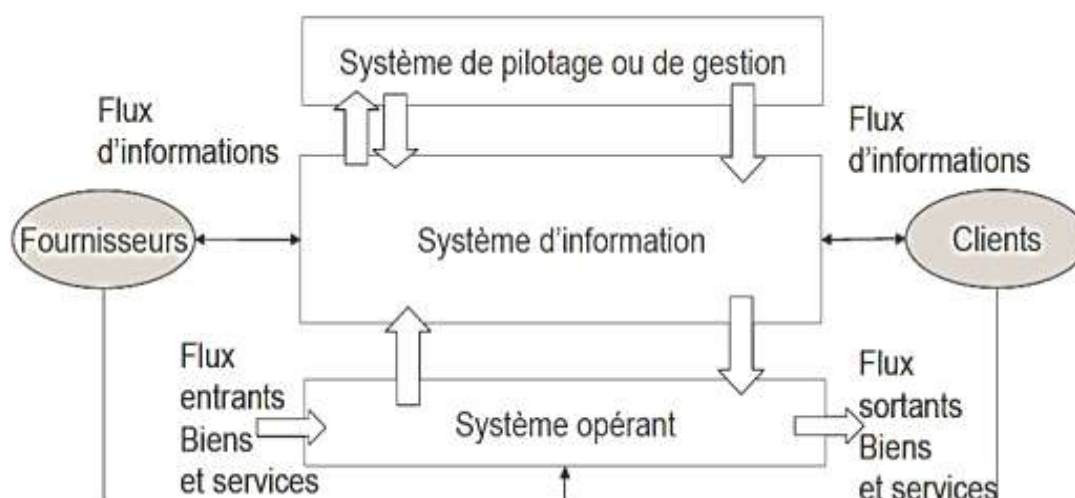
⁷**La non répétitivité du projet.** Ceci signifie qu'il est nécessaire de mettre en place une organisation spécifique, éventuellement distincte de la structure de l'entreprise, spécialement adaptée et temporaire, voire évolutive au cours du déroulement du projet. Cette caractéristique fait référence au fait que le projet doit devenir autonome et non toujours dépendant du donateur.

- **La résistance peut être liée à l'organisation:** il s'agit d'une problématique impact collectif. Le nouveau système modifie la division du travail, la coordination, la collaboration ou coopération requise, les activités de contrôle... ce qui peut entraîner des oppositions. Deux approches explicatives ont été proposées :

- **Selon la perspective dite « sociotechnique » :** certains acteurs manifestent une résistance parce qu'ils pensent que le nouveau système dégradera leurs conditions de travail. Il s'agit, si l'on veut prévenir une telle résistance, de s'appuyer lors l'analyse des processus sur des principes d'ergonomie (*par exemple, appréciation du stress causé par certaines tâches*) et d'organisation des postes (*par exemple, degré de responsabilité et d'autonomie*), et de favoriser une participation à l'analyse des processus et à l'organisation des sites.
- **Selon la perspective dite « politique » :** certains acteurs s'opposent parce qu'ils pensent que le nouveau système leur enlèvera du pouvoir. L'opposition peut provenir de différents niveaux hiérarchiques. Un cadre peut voir diminuer son pouvoir de décision et d'action, ou la taille du groupe placé sous sa responsabilité. Un opérationnel peut voir réduire sa « zone d'incertitude », liée à la détention de son savoir-faire ou d'informations clés. Certains projets système d'information conduisent, pour atteindre des objectifs d'efficacité, à remettre en question des territoires. Cette situation doit être prise en compte, en particulier dans l'analyse des risques et l'élaboration d'une stratégie adéquate. La construction d'une vision d'un « *bien commun* », partagée par les acteurs en conflit, peut débloquer la situation : un engagement de la direction de l'entreprise et/ou la référence à des valeurs collectives sont des moyens d'y parvenir.

DEUXIEME CHAPITRE - LA CONDUITE DU PROCESSUS DES PROJETS INFORMATIQUES

L'administration d'un projet informatique en mutation requiert un ensemble d'opérations impliquant des exécutions fortuites. C'est pourquoi l'IPMA précise que *« le conduite de projet consiste à planifier, organiser, suivre et maîtriser tous les aspects d'un 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 »*.



II.1. LES ACTIVITÉS DE GESTION DE PROJET

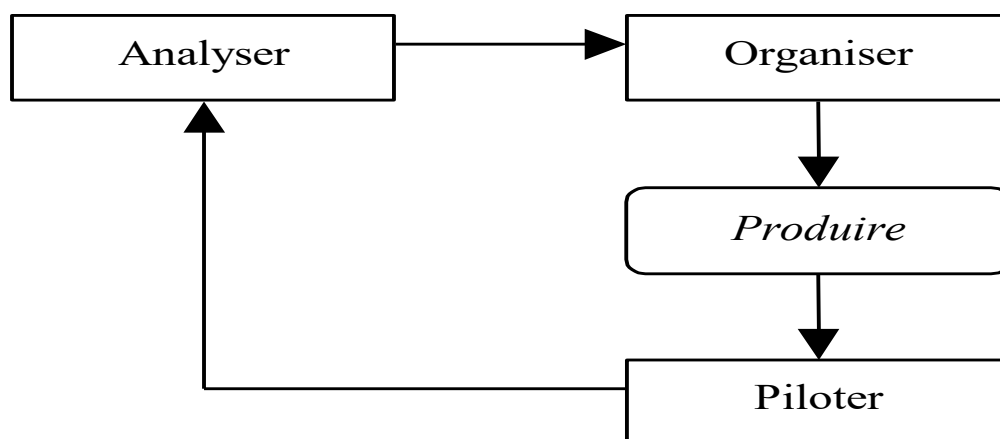
Pour le PMBOK, la gestion de projet est *« l'application de connaissances, de compétences, d'outils et de méthodes aux activités d'un projet afin de répondre à ses besoins »*. Il convient toutefois de distinguer deux catégories d'activités dans un projet :

- **les activités de production** : sont celles qui sont nécessaires pour réaliser le produit ou service visé par l'objectif du projet ;
- **les activités de gestion de projet** : sont celles qui consistent à planifier le travail, l'organiser et suivre l'avancement des activités de production.

Le champ de la conduite de projets informatiques est calé sur les trois aspects représentés par le triangle Projet. Ainsi :

- *Le délai* donne lieu à une *gestion du temps* dont le rôle est de définir le parcours et de le jalonner, 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... 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. Cela implique une *gestion de la production*, qui a pour but de suivre et diriger l'avancement vers l'objectif tout au long du projet. On parle parfois de « *faire converger le projet* » : cela signifie qu'il faut sans cesse s'assurer que l'on se rapproche du but et que l'on ne part pas dans des directions remettant en cause un avancement consolidé.

On peut décomposer l'activité de la conduite des projets informatiques en trois activités principales autour de la production proprement dite :



Les activités de gestion de projet

- **Analyser** : consiste à déterminer le chemin que l'on va emprunter pour avancer vers l'objectif. Pour cela, on étudie les caractéristiques du projet, son contexte, les risques qui le menacent et l'état de son avancement. Cela conduit à un découpage du projet en actions à entreprendre et à une estimation de l'effort nécessaire. La maille de cette analyse varie selon le moment du projet auquel on se place.

- **Organiser** : signifie repérer les contraintes d'enchaînement entre les tâches afin de les ordonnancer. Cela permet d'établir un calendrier. L'organisation recouvre aussi la constitution d'une équipe, c'est-à-dire des personnes qui sont affectées et imputées au projet, en déterminant les bons profils. Les relations avec tous les partenaires nécessaires sont également prises en compte. Dès que la charge est importante, on répartit le travail entre plusieurs personnes, voire plusieurs équipes, ce qui conduit à mettre en place des moyens de partage d'informations pour éviter les incohérences.
- **Produire** : consiste à assurer les actions diverses (*cultures, fabrications, etc.*) qui concourent à fournir sur le marché commercial certains biens ou services.
- **Piloter** : comprend le suivi de l'avancement du projet, en quantité et en qualité, ainsi que l'analyse et le traitement des écarts avec ce qui était prévu, les orientations et les décisions à prendre ou à faire prendre. Le pilotage inclut également le management de l'équipe et la gestion des conflits.

Pour la norme ISO10006: 2003, les activités de gestion de projet sont organisées en processus, et réparties en quatre familles :

- **La première famille** ne comprend qu'un seul processus, appelé processus stratégique, qui regroupe les activités relevant de la responsabilité de la direction de l'entreprise, telle que la nomination des chefs de projet.
- **La deuxième famille** comprend des processus qui touchent à la gestion des ressources, en particulier le personnel.
- **La troisième famille** structure les activités de gestion de projet orientées vers la réalisation du produit, c'est-à-dire la production du bien ou service visé par l'objectif. Elle comprend la coordination des activités, la maîtrise des tâches de production, la maîtrise des délais et des coûts, la gestion de la communication, la maîtrise des risques et celle des achats pour le projet.
- **La quatrième famille** de processus réunit les activités de gestion de projet, qui visent l'amélioration, c'est-à-dire le management des connaissances issues des projets achevés.

II.2. PRINCIPAUX ACTEURS DANS UN PROJET INFORMATIQUE

Un acteur est quelqu'un qui joue un *rôle* dans le déroulement du projet. La notion de rôle est importante dans un univers projet. C'est une fonction temporaire, déconnectée de la place qu'occupe la personne dans l'organigramme de l'entreprise ; on attache au rôle des activités à effectuer et une responsabilité. Quand on organise un projet, on commence par déterminer les rôles nécessaires. On distingue trois types d'acteurs participant à un projet de développement de système d'information :

- le maître d'œuvre et
- le maître d'ouvrage ;
- l'équipe de projet ;
- les utilisateurs.



Les intervenants du projet informatique doivent être identifiés avec leurs rôles et les fonctions qu'ils doivent remplir dans quel ordre et dans quelles conditions. Cette notion de responsable à chaque niveau d'intervention est essentielle à la bonne marche du projet. *La première démarche* consiste à définir les conditions d'intervention des différents partenaires et *la seconde* à préciser les liens qui peuvent exister entre eux.

II.2.1. LE MAITRE D'OUVRAGE

Selon l'AFNOR, le maître d'ouvrage est « *la personne physique ou, le plus souvent, personne morale qui sera le propriétaire de l'ouvrage. Il fixe les objectifs, l'enveloppe budgétaire et les délais souhaités pour le projet* ». À ce titre, il « *assure le paiement des dépenses liées à la réalisation* ».

Les rôles de *maître d'œuvre* et *maître d'ouvrage*⁸ sont issus des grands projets industriels. Ils sont liés par une relation contractuelle. *Le maître d'ouvrage* représente « *le client* ». Partant d'une demande des futurs utilisateurs du système, il établit un cahier des charges qui peut servir de base à un appel d'offres. Après discussions sur une ou plusieurs propositions reçues, il passe contrat avec un fournisseur qui jouera le rôle de maître d'œuvre. Celui-ci est responsable de la conduite du projet. *Le maître d'ouvrage assure un suivi de l'avancement du projet, selon des modalités contractuellement définies*. À chaque fourniture contractuelle par le maître d'œuvre, il procède à la recette et prononce l'acceptation ou le refus. Il pilote la mise en œuvre du projet. Strictement parlant, *l'ouvrage est le résultat concret d'un projet : construction, matériel, progiciel*. Bref, c'est le « *bailleur de fonds* ».

II.2.2. LE MAITRE D'OEUVRE

Le maître d'œuvre est « *la personne physique ou morale qui réalise l'ouvrage pour le compte du maître d'ouvrage et qui assure la responsabilité globale de la qualité technique, du délai et du coût* ». L'œuvre est définie comme « *le processus de réalisation de l'ouvrage, c'est-à-dire la mise en place des moyens nécessaires à cette réalisation et à leur conduite* ». Par conséquent, elle « *est constituée de l'ensemble des tâches, regroupées ou non en lots de travaux* ».

II.2.3. L'EQUIPE DU PROJET

L'*équipe de projet* rassemble différents acteurs, qui sont affectés au projet⁹. D'après, l'AFNOR, c'est « *l'ensemble des personnes placées sous l'autorité directe (et quelquefois indirecte) du chef de projet* ». Mais, on peut parfois considérer que l'équipe de projet s'étend à *toutes les personnes participant à la réalisation du projet*. On distingue ainsi trois différents types d'acteurs dans l'équipe de projet :

- Le *chef de projet*, est responsable devant le maître d'œuvre de l'avancement du projet.
- Le *concepteur*, qui peut être tenu par un informaticien, et qui joue le rôle d'un organisateur ou un gestionnaire selon le stade d'avancement : sa responsabilité est de concevoir le futur système aux étapes étude préalable et étude détaillée.

Le *développeur*, qui est tenu par un informaticien : sa responsabilité est d'écrire les programmes ou de réaliser un prototype. Pour certains développements réalisés en langage de 4e génération, le rôle peut être tenu par un gestionnaire.

⁸ La transposition de ces deux rôles quand il s'agit de système d'information renvoie davantage à la répartition des responsabilités qu'à la notion de propriété. En effet, même si l'on parle parfois du « *propriétaire d'une application informatique* », il est difficile d'étendre ce droit de propriété aux processus et aux acteurs.

⁹ Les autres différents acteurs, qui sont affectés au projet peuvent aussi être *les consultants, analystes, développeurs ...*).

La performance d'une équipe dépend de l'équilibre de ces positionnements, appelés « rôles », parmi les membres. Au-delà de la répartition des activités, chacun endosse en général un « rôle », et des rôles complémentaires favoriseront la synergie, alors que des « rôles » non endossés manqueront dans la dynamique du groupe. C'est ainsi **Meredith Belbin**¹⁰ a identifié neuf types de « rôles » regroupés en trois catégories, selon l'orientation majeure : *vers les personnes*, *vers l'action* et *vers la production intellectuelle* :

- **Les rôles orientés vers les personnes sont les suivants :**
 - *Le coordinateur* a tendance à clarifier les buts et favoriser la prise de décision dans le groupe ;
 - *Le coéquipier* est attentif aux autres et les encourage ;
 - *Le pourvoyeur de ressources*, dispose d'un carnet d'adresses qu'il est prêt à ouvrir pour les membres du groupe.
- **Les rôles orientés vers l'action sont les suivants :**
 - *L'organisateur* a tendance à traduire les idées en tâches concrètes ;
 - *L'entraîneur* aide l'équipe à conserver le cap vers les objectifs et maintient la pression ;
 - *Le contrôleur* vérifie les détails, il a le souci du respect du calendrier et des engagements.
- **Les rôles orientés vers l'intellect sont les suivants :**
 - *L'innovateur* propose de nouvelles idées, mais il peut être irréaliste et peu communicateur.
 - *L'évaluateur* est au contraire réaliste et il aime analyser les problèmes complexes.
 - *Le spécialiste* possède des connaissances précises, utiles à l'équipe.

II.2.4. LES UTILISATEURS

Les utilisateurs ont pour rôle la mise en correspondance avec les fonctions permanentes du développement du produit réalisé. Ces derniers peuvent représenter différents niveaux : *décideurs*, *gestionnaires* ou *utilisateurs finals*. Quand on évoque l'intervention d'utilisateurs dans un projet, on emploie souvent le terme générique de « *participation* ». En réalité, la nature et les modalités de participation peuvent présenter une grande variété et servir différents buts. Dans une perspective de pilotage de projet, on considère que la participation peut avoir un effet sur la détermination des besoins, sur le processus de décision ou sur une évolution organisationnelle.

¹⁰ Professeur à l'université de Cambridge.

On peut alors distinguer 3 catégories des utilisateurs :

- L'*utilisateur final* est celui qui va utiliser les transactions et les éditions du futur système dans son travail au quotidien. C'est, par exemple, l'employé de l'agence bancaire qui se sert de son terminal pour effectuer les opérations avec la clientèle. La responsabilité attachée à ce rôle est d'exprimer des besoins et des contraintes liées au travail courant.
- L'*utilisateur gestionnaire* est un opérationnel qui a des responsabilités d'encadrement. Il utilisera le futur système pour obtenir des informations permettant de prendre des décisions de gestion. Par exemple, un responsable produit attendra des informations permettant de suivre une vente promotionnelle. La responsabilité attachée à ce rôle est d'exprimer des besoins en informations favorisant la gestion à moyen terme de l'activité.
- L'*utilisateur décideur* a le pouvoir de modifier les règles du système de gestion, pour utiliser au mieux les possibilités technologiques. Sa responsabilité est de prendre des décisions sur l'évolution des règles de gestion.

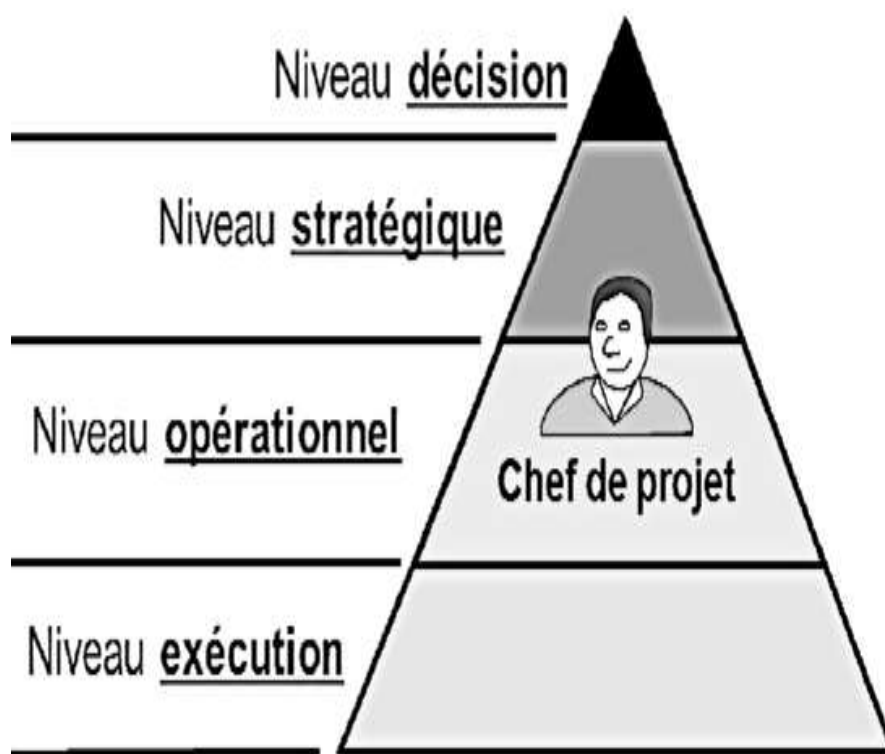
Ainsi la participation peut avoir pour objectif de transférer à l'équipe de projet une connaissance sur un domaine, un système de gestion, une organisation existante. La participation peut faciliter le processus de décision, en augmentant la visibilité des décideurs sur les enjeux et conséquences des décisions. Si les utilisateurs sont multiples, la prise de décisions passe par la construction d'un consensus. On peut citer à cet égard la technique JRP¹¹. Enfin, la participation peut rechercher des décisions d'orientation et de poursuite du projet. Les utilisateurs visés sont des décideurs, et la poursuite du projet pourra dépendre de leur appréciation et de leur engagement. Cette participation se fait généralement à travers un comité de pilotage, mais elle peut être plus informelle.

Prendre une décision, c'est souvent prendre un risque. Celui-ci est d'autant plus grand que les conséquences du choix se font sentir à moyen ou long terme. C'est pourquoi on observe parfois une réticence à s'engager. La participation permet de réduire cette crainte, en donnant aux décideurs une meilleure connaissance du projet et en offrant des possibilités d'échange et de discussion avec d'autres acteurs. La participation peut enfin faciliter l'utilisation du futur système par l'utilisateur final.

¹¹JRP (Joint Requirements Planning). Elle a pour objectif de donner un rôle actif à l'utilisateur lors de l'étape d'expression des besoins, afin de réduire la durée de cette étape et d'en augmenter la qualité. Les utilisateurs, qui sont également décideurs, jouent alors un rôle central : ils effectuent la collecte des informations nécessaires à l'expression des besoins ; ensuite, réunis en session avec les informaticiens, ils présentent leurs résultats qui sont confrontés, amendés, acceptés, pendant une période de temps courte mais dense.

II.3. NIVEAUX DE RESPONSABILITÉ DANS LE PROJET INFORMATIQUE

En matière de gestion de projets informatiques, le niveau de responsabilités, c'est un ensemble des étapes successives de l'analyse, hiérarchiquement subordonnées les unes aux autres à partir d'une unité supérieure, c'est-à-dire c'est un échelon (degré) d'un ensemble organisé des implications dans une hiérarchie quelconque. Comme dans la plupart des organisations humaines, la distribution des responsabilités et des tâches se fait sur le mode pyramidal du schéma ci-contre. Décrivons sommairement les quatre niveaux, en partant de la base de la pyramide.



- **Le niveau exécution** : est celui des entreprises en charge des travaux et des contributeurs internes, sollicités pour leur expertise métier. Le niveau exécution réalise les différents livrables du projet. Il applique les directives établies par le niveau opérationnel et lui rend compte des problèmes éventuels et de l'avancement. Le niveau exécution n'a pas à connaître du projet plus que les informations strictement nécessaire à la bonne réalisation des tâches qui lui sont confiées.
- **Le niveau opérationnel** : est celui du chef de projet et de son équipe. Il décline les exigences du cahier des charges en solutions techniques et en tâches à accomplir. Il pilote le projet au quotidien, détecte et corrige les écarts et rend compte au niveau stratégique.

- **Le niveau stratégique** : détermine l'objectif, nomme le chef de projet, définit le cadre général du projet et l'organisation, puis s'assure que la trajectoire du projet est conforme aux prévisions. Bien entendu il intervient en cas d'écart grave, sur demande du chef de projet ou de sa propre initiative.
- **Le niveau décision** : prend les grandes orientations, en principe sur proposition du niveau stratégique. Les solutions de gestion de projet diffèrent entre elles, en premier lieu par le service qu'elles rendent (*ou pas*) aux acteurs des différents niveaux. A minima, elles permettent au chef de projet de piloter les travaux. Les solutions les plus abouties fournissent aux acteurs de chaque niveau les informations qui leurs sont utiles : un exécutant aura la vision des tâches qu'il doit accomplir, le chef de projet aura accès à la totalité des données du projet. Quant aux acteurs du niveau stratégique, ils disposeront d'un tableau de bord synthétique les renseignant principalement sur la santé du projet. Bien entendu toute information saisie à un niveau est instantanément répercutée par le système aux acteurs concernés, c'est bien le moins que l'on attende d'un système informatique.

L'AFITEP appelle « *management de projet* », l'ensemble des activités de gestion de projet et introduit une distinction entre deux rôles. Le chef de projet est responsable de la « *direction de projet* » et s'appuie sur des assistants chefs de projet qui accomplissent des tâches rassemblées sous le terme « *gestion de projet* ». Ces deux rôles correspondent à deux niveaux de certification. La direction de projet, qui est toujours exercée par le chef de projet, a pour mission essentielle de définir la stratégie de conduite du projet, de coordonner l'ensemble des actions, de modifier le plan de marche si nécessaire, et d'optimiser l'utilisation des ressources.

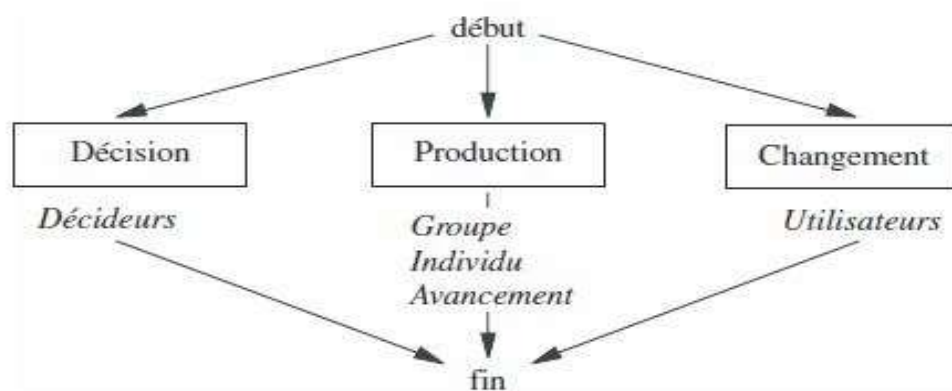
Pour accomplir la mission de direction de projet, le chef de projet a besoin d'analyser les risques, d'estimer la charge, d'organiser le travail, de le planifier, de le suivre. Pour cela, il doit effectuer, ou faire effectuer, un ensemble de tâches relevant de la gestion de projet, notamment la préparation des différents plans et l'élaboration de tableaux de bord de suivi. La distinction entre gestion et direction de projet n'est pas, à ce jour, reprise par les autres acteurs de la normalisation.

II.4. ROLES DE CHEF DE PROJET

Par définition, Le chef de projet est défini par l'AFNOR comme : « *la personne physique chargée par le maître d'œuvre, dans le cadre d'une mission définie, d'assurer la maîtrise du projet, c'est-à-dire de veiller à sa bonne réalisation dans les objectifs de technique, de coût et de délai* ». Cette définition est insuffisante pour les projets système d'information. Heureusement, la définition est élargie : « *Souvent le maître d'ouvrage identifie, en son sein, un interlocuteur au chef de projet du maître d'œuvre. Cet interlocuteur est aussi appelé chef de projet* »¹².

Le chef de projet doit faire en sorte que le projet réussisse. Pour cela, sa responsabilité est multiple : il a la charge d'une bonne gestion du groupe et des individus ; il pilote la production des livrables pour un achèvement en temps et en heure ; il participe à la gestion du changement, en particulier auprès des acteurs clés du projet ; il lui revient enfin de veiller à ce que les processus de décision ne soient pas bloqués. Un chef de projet doit au préalable développé les éléments suivants :

- les styles de management (conduite) d'équipe et leur adéquation ou inadéquation à différentes situations pour atteindre les objectifs du projet;
- la motivation des membres de l'équipe ;
- la gestion des conflits à l'extérieur du groupe de projet.



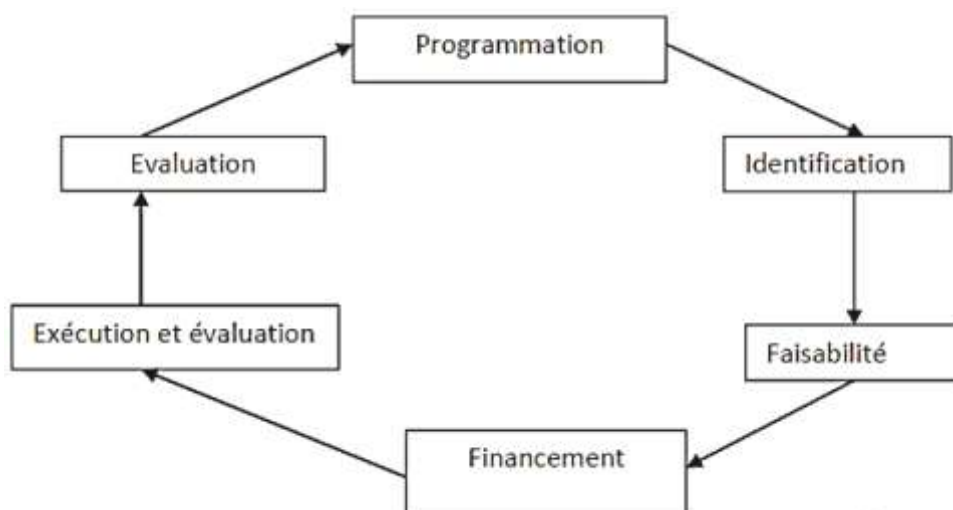
- **Il est responsable du groupe** - D'un ensemble d'individus, il doit constituer une équipe, l'animer et en maintenir la cohésion. L'équipe prend corps lorsque l'objectif (le projet) devient collectif. Pour obtenir cette adhésion et cette synergie, il faut que chacun soit responsable en propre d'une partie du travail ; il faut favoriser les échanges latéraux grâce notamment à des points réguliers de coordination. L'esprit de groupe se construit par des échanges directs d'individu à individu sur un but commun. C'est pourquoi il est souhaitable que la taille de l'équipe reste limitée. Au-delà de quinze à vingt personnes, le sentiment d'appartenance se dilue.

¹² AFITEP, 2000

- **Le chef de projet est responsable des individus membres de l'équipe**, qu'il doit valoriser et soutenir. Ce domaine est l'un de ceux où, par chance, le travail est souvent un moyen de se réaliser. Or, la satisfaction qu'un acteur retire de son activité sur le projet est le plus puissant des moteurs. Chercher à ce que chacun se sente gratifié par le travail accompli est une excellente approche managériale.
- **Le chef de projet doit prendre en compte les souhaits des individus et l'enrichissement apporté**, lorsqu'il attribue les tâches. L'estimation de la charge affectée doit être négociée avec celui qui va accomplir la tâche correspondante. S'il y avait désaccord dès le départ, le risque serait grand de dépasser le délai. L'avancement doit être suivi de près, pour ne pas laisser l'un ou l'autre piétiner. En cas de difficulté, le dialogue est indispensable. Un retard sur une tâche peut être dû à une cause exogène (*panne de la machine de développement*) ou organisationnelle (*demande de fonctionnalité supplémentaire*), qui est du ressort du chef de projet.
- **Le chef de projet est également responsable de l'avancement des travaux et doit convaincre l'équipe de la nécessité d'un suivi adéquat**. Le système d'information projet est indispensable à la réussite : il faut que les chiffres de base soient le plus juste possible. Pour cela, le chef de projet doit utiliser avec perspicacité les chiffres issus du bilan individuel. Les informations du compte rendu d'avancement doivent être traitées immédiatement.
- **Le chef de projet est un acteur du changement parmi les utilisateurs** - Il doit pour cela organiser une participation adaptée afin de créer un noyau qui portera le changement. Si aucune dynamique n'a été déclenchée parmi les utilisateurs, la mise en œuvre risque d'être difficile et le succès du projet compromis. Il appartient au chef de projet de repérer et d'associer au projet certains utilisateurs convaincus de son utilité et qui s'en feront les précurseurs.
- Le chef de projet est, à certains égards, **le pilote des décisions** touchant au contenu du projet. Toute décision à prendre, si elle n'est pas prise immédiatement, doit lui être signalée. Toutes celles qui sont en suspens doivent figurer sur son tableau de bord, pour qu'il en suive le traitement. Il doit parfois aider à choisir en apportant un éclairage aux décideurs ou en proposant des solutions flexibles si un choix définitif ne peut être fait.

II.5. L'ADMINISTRATION DU CYCLE DE PROJETS INFORMATIQUES

Du moment où le projet a un début et une fin, cette notion conduit à la notion de cycle de projet. Ce cycle peut se résumer dans le schéma suivant :



1. **La programmation** : Le processus de programmation est l'issus d'un document global stratégique pour un pays donné. Ce document détermine les différentes perceptions, profils et diagnostic sur l'ensemble d'un pays et il doit être en accord avec la législation de ce dernier pour l'exécution d'un projet informatique.
2. **L'Identification** : C'est une phase clé dans laquelle les analyses suivantes sont faites pour garantir la pertinence et la faisabilité d'une idée de projet :
 - Analyse du contexte politique et de programmation ;
 - Analyse des parties prenantes, y compris l'analyse de la capacité institutionnelle ;
 - Analyse des problèmes, y compris la prise en compte des questions transversales (*genre, gouvernance, environnement,...*) ;
 - Appréciation d'autres initiatives similaires en cours et prévus, et des enseignements tirés du passé ;
 - Analyse préliminaires des objectifs et des stratégies;
 - Une première appréciation des implications au niveau des moyens et des coûts;
 - Une première analyse des dispositifs de mise en œuvre, coordination et financement;
 - Analyse préliminaire de la viabilité économique, financière, environnementale, technique et sociale.

3. La Faisabilité : cette phase, il est question de voir si le projet répond aux besoins prioritaires (*pertinence*), le projet a été bien conçu et apporte des avantages tangibles et durables aux groupes cibles, la formulation du projet est bien gérée (*bonne gestion*) ... Cette étape a deux objectifs principaux :

- Démontrer, à l'institution qui va mener le projet, que celui-ci est faisable, viable et rentable;
- Servir de document de référence à toute requête de financement adressée à des bailleurs de fonds externes.

En outre dans

4. Le Financement : Le financement a comme objectifs de permettre l'appréciation du projet par l'organisme de financement et la négociation avec l'organisme de financement avant la passation d'accord de financement, quel qu'en soit le statut: subvention, prêts, apport en nature. Voici les étapes clés dans la phase de financement :

- Introduction de la demande de financement par le l'organisme promoteur du projet s'appuyant sur l'étude de faisabilité ;
- Appréciation de la demande par le bailleur de fonds (*comportant souvent une mission d'appréciation sur le terrain*) ;
- Négociation technique portant sur le contenu technique du projet ;
- Négociation portant sur les aspects financiers du projet: montant de la participation, termes et conditions (*taux d'intérêts, durée de remboursement*) et sur les conditions de financement (*procédure d'exécution, ...*) ;
- Signature de la convention de financement.

5. La mise en œuvre : La mise en œuvre a comme objectif de permettre au projet de produire les résultats, atteindre les buts et contribuer efficacement aux objectifs globaux du projet, gérer les ressources disponibles de façon efficiente, suivre le déroulement et en faire les comptes-rendus. Les principales phases de mise en œuvre du projet sont :

- *Le démarrage :* cette phase regroupe les éléments ci-après : les différentes terminaisons des conventions, la mobilisation des ressources, la définition des conditions et relations du travail, l'aménagement du démarrage, les auditions et révisions du schéma du projet et l'établissement des systèmes d'évaluation et de contrôle (*M&E*).

- *La mise en œuvre proprement dite* : celle-ci regroupe fréquemment les éléments suivants : la mise en place des ressources, la mise en œuvre des activités et obtention des résultats, le suivi et le contrôle de l'état de l'avancement du projet, la révision des plans opérationnels, et les rapports sur l'état de l'avancement du projet.
- *La phase finale* : celle-ci regroupe autant graduellement les éléments suivants : les différents transferts de toutes les responsabilités aux collaborateurs régionaux, la vérification de la mise en place des plans d'entretien, la vérification du transfert réel des compétences indispensables, la vérification de la couverture des dépenses récurrentes.

6. L'évaluation : L'évaluation a pour but d'effectuer une appréciation, aussi systématique et objective que possible, d'un projet, d'un programme ou d'une politique, en cours ou achevé, de sa conception, de sa mise en œuvre et de ses résultats. Elle se fixe à trois types de résultats :

- *La réalisation physique du projet* : voir si le projet est en train d'être exécuté physiquement ;
- *L'évaluation des effets* : vise à estimer dans quelle mesure les actions du projet ont eu les effets visés ;
- *L'évaluation des impacts* : mesure les améliorations des conditions de vie que les effets du projet ont pu entraîner.

Une bonne évaluation dans un projet informatique requiert Les principes ci-après :

- *L'Impartialité et indépendance*, du processus d'évaluation par rapport aux fonctions de programmation et de mise en œuvre ;
- *La Crédibilité*, de l'évaluation, grâce à l'intervention d'experts indépendants et compétents et à la transparence du processus ;
- *La Participation*, des parties prenantes au processus d'évaluation, afin de garantir que les différents aspects et points de vue soient pris en compte ;
- *L'Utilité*, des observations et recommandations de l'évaluation, grâce à la fourniture en temps utile aux décideurs d'une information pertinente, claire et concise.

TROISIEME CHAPITRE - DYNAMIQUE DU PROCESSUS DE DEVELOPPEMENT DE PROJETS INFORMATIQUES.

III.1. DEFINITION ET CONTEXTE D'ETUDES

Par définition, la dynamique du processus du développement de projets informatiques peut être considérée comme étant une démarche de l'évolution au l'intérieur d'une structure en développement.

Compte tenu de son unicité, tout projet est soumis à incertitude, car des événements imprévisibles ou non prévus peuvent survenir et entraver son déroulement. C'est pourquoi les méthodes et normes recommandent de découper le processus projet en « *phases*¹³ ». Les phases sont ordonnées dans le temps, selon la logique de construction du produit. Ce découpage introduit un jalonnement du projet, qui permet au commanditaire du projet de valider progressivement les livrables et d'orienter le projet. Il favorise une maîtrise accrue du projet. L'ensemble des phases du projet est appelé « *cycle de vie* » du projet. Le cycle de vie est unique pour chaque projet, mais il existe selon les domaines d'application des cycles de vie génériques qui guident le découpage du projet en phases. En informatique, on les appelle souvent des modèles de développement.

III.2. DECOUPAGE DE PROJETS INFORMATIQUES

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



¹³Une phase est un regroupement de tâches conduisant à un produit livrable principal et d'éventuels produits livrables secondaires.

- **L'étude préalable** - L'*objectif* d'une étude préalable est double. D'une part, il s'agit de faire des choix structurants pour la future application : évaluer l'adéquation de la solution aux objectifs, choisir éventuellement entre plusieurs solutions, évaluer l'investissement (*budget, temps*), ajuster la solution à l'enveloppe si cela est nécessaire. D'autre part, il s'agit de fournir une base de référence pour la suite du projet : le rapport d'étude préalable peut donc être considéré comme un cahier des charges pour l'étude détaillée. Une étude préalable comporte trois étapes : **observation, conception-organisation et appréciation** :

- L'*objectif* de l'**étape observation** est de donner une représentation pertinente du domaine étudié, suffisante pour porter un diagnostic et mettre en évidence des besoins. Le *résultat* comprend :
 - une structuration du domaine en processus, qui va ensuite guider un éventuel découpage structurel, pour établir un WBS par exemple ;
 - le choix d'un sous-ensemble représentatif (SER) : en effet, si le domaine est important, on va se limiter à une partie du domaine, en utilisant la notion de variante de procédure ;
 - une description du fonctionnement du SER ;
 - une description modélisée des données ;
 - un diagnostic.
- L'*objectif* de l'**étape conception-organisation** est de proposer une ou plusieurs solutions, aux niveaux conceptuel et organisationnel, sur tout ou partie du domaine. Le *résultat* comprend un modèle de données consolidé ou enrichi, ainsi qu'une description d'au moins une variante de chaque processus, avec les traitements et les règles de gestion.
- L'*objectif* de l'**étape appréciation** est d'une part de dresser un bilan des avantages attendus et des coûts prévisibles (*étude de rentabilité*), d'autre part d'élaborer un plan pour la poursuite du projet. On peut ainsi identifier différents sous projets qu'il faut ordonnancer. Le découpage en sous-projets repose sur un découpage structurel ; par exemple, on peut définir un sous-projet par processus. L'ordonnancement se fait selon :
 - une éventuelle priorité stratégique de certains processus ;
 - la périodicité (*traitements quotidiens, puis mensuels, puis annuels*) ;
 - les contraintes logistiques (*arrivée d'un matériel, mise en place d'un réseau*).

- **La conception détaillée** - L'*objectif* d'une étude détaillée est de concevoir et décrire de façon exhaustive la solution sur tout le champ de l'étude. Elle sera ensuite complétée par l'étude technique. Les spécifications ainsi obtenues doivent faire l'objet d'un consensus entre futurs utilisateurs et informaticiens. Elles représentent le cahier des charges pour la réalisation. Le *résultat* comprend toute la vision externe du système : l'interface homme-machine (maquettes d'écran, cinématique), la description des traitements à une maille suffisamment fine pour qu'il n'y ait plus d'ambiguïté fonctionnelle, ainsi que les sorties (maquettes d'état). On y ajoute souvent l'organisation à mettre en place et le planning détaillé.
- **Les études techniques** - L'*objectif* de cette phase, qui ne concerne que les informaticiens, est d'optimiser les structures physiques de données et de construire les traitements (dossier programmes) en essayant de préparer la réutilisation du code. Le *résultat* comprend des normes techniques, des dossiers de programme et la structure physique des données. Il complète le cahier des charges de réalisation.
- **La réalisation** - Cette phase est parfois appelée « développement ». L'*objectif* est de produire un logiciel testé. Elle comprend donc des tâches d'élaboration de jeu d'essai, de programmation et de test. Elle se termine par une procédure d'acceptation officielle est appelée *recette* : le client fournit un jeu d'essai et vérifie avec le fournisseur la conformité du logiciel à ce qu'il avait demandé. Dans la pratique, la recette fait souvent une étape séparée. On effectue parfois une recette provisoire après la réalisation et une recette définitive après la mise en œuvre. Dans une relation contractuelle donnant lieu à des flux financiers, la recette conditionne le paiement.
- **La mise en œuvre** - L'*objectif* est de préparer le démarrage effectif de la nouvelle application. Cette phase comprend notamment le paramétrage, la reprise ou l'alimentation des données, le développement d'interfaces, la formation des utilisateurs, l'installation d'environnement d'exploitation. La gestion du changement décrite au chapitre 5 relève de la mise en œuvre, qui souvent commence dès la fin de l'étude détaillée.
- **L'évaluation (qualification)** - L'*objectif* est de réaliser des tests dans l'environnement opérationnel et de tirer un bilan du système d'information installé, selon différents critères qualité. Ces derniers sont présentés au chapitre 8 qui regroupe tous les aspects de la qualité.

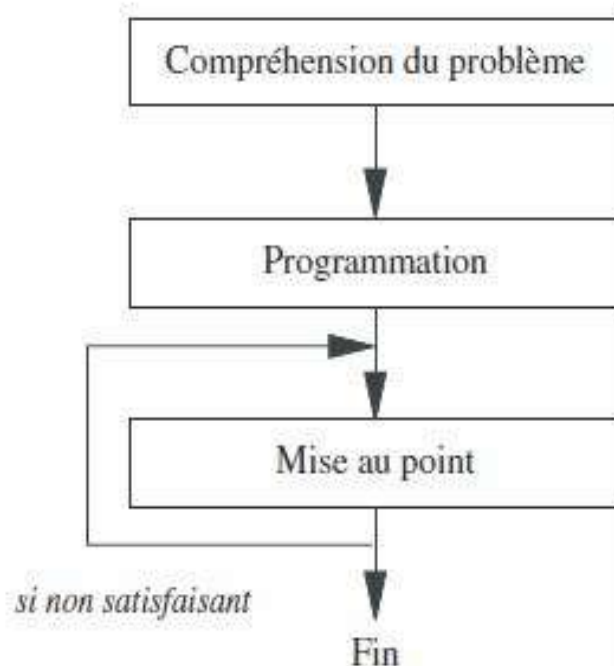
III.3. LES MODELES DE DEVELOPPEMENT

Il s'avère de nos jours, que nous ne pouvons plus avoir une démarche unique dans le développement de projets informatiques, mais qu'il faut construire le découpage temporel en fonction des caractéristiques de l'entreprise et du projet. On s'appuie pour cela sur des découpages temporels génériques, appelés *modèles de développement (process models)* ou modèles de cycle de vie d'un projet informatique. Les principaux modèles sont :

- 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.

III.3.1. LE MODELE DU CODE-AND-FIX

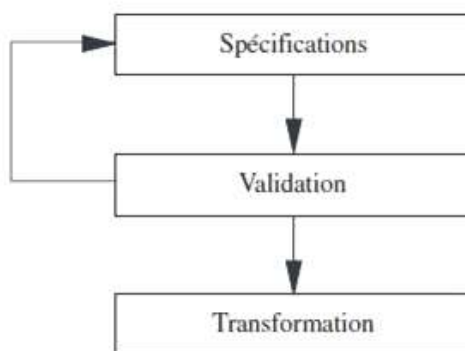
C'est un modèle qui repose sur la possibilité d'une détermination facile des besoins : une première phase de Compréhension du problème est suivie d'une phase de Programmation ; puis une phase de Mise au point, parfois en collaboration avec l'utilisateur du futur système, est répétée jusqu'à l'atteinte du résultat visé.



Le modèle du code-and-fix.

III.3.2. LE MODELE DE TRANSFORMATION AUTOMATIQUE

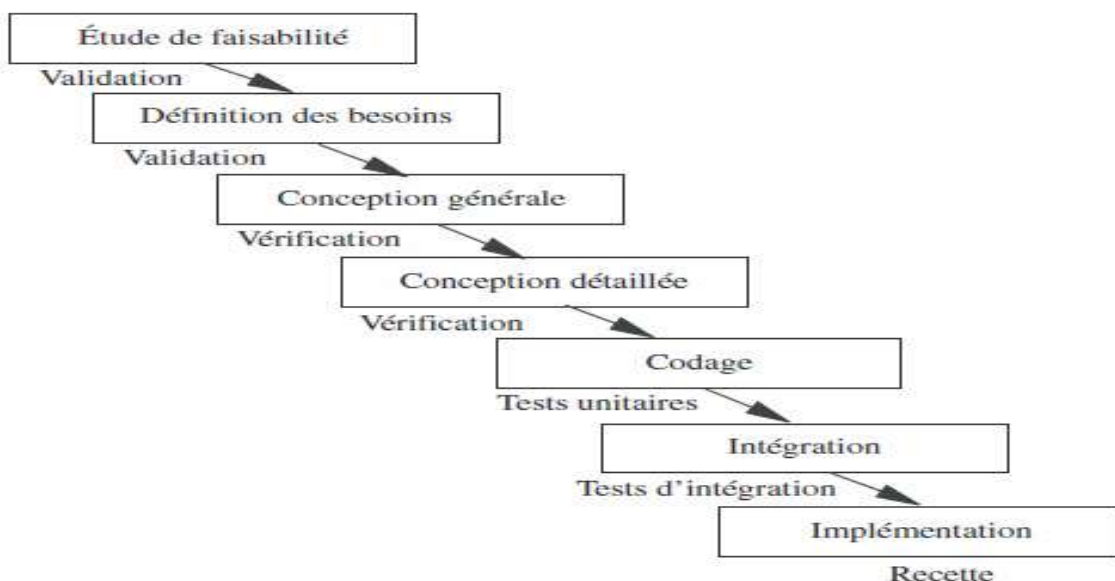
C'est un modèle basé sur la possibilité de transformer automatiquement des spécifications en programmes. L'essentiel de l'effort va donc porter sur une description exhaustive des spécifications qui devront être complètement validées. Une succession de cycles Phase de Spécifications – Phase de Validation s'achève par la phase de Transformation, où s'effectue la génération du code.



Le modèle de la transformation automatique

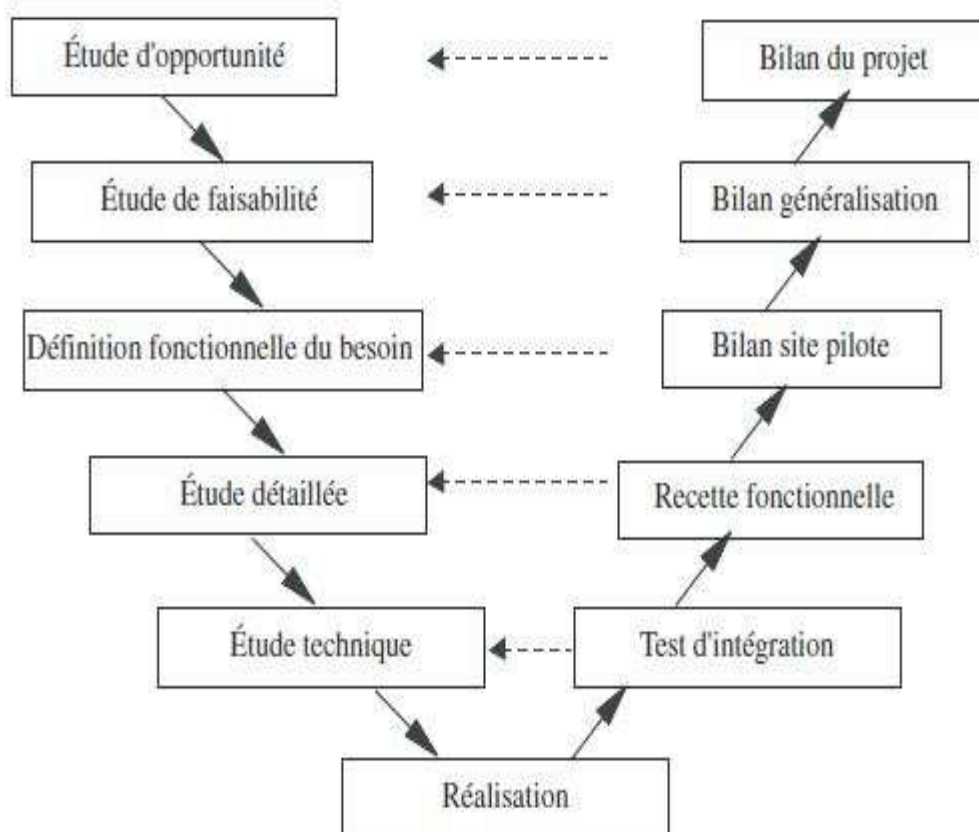
III.3.3. LE MODELE DE LA CASCADE

C'est un modèle qui a comme objectif majeur de jalonner (*présenter sobrement*) rigoureusement le processus de développement et de définir de façon précise les rôles respectifs du fournisseur qui produit un livrable et du client qui accepte ou refuse le résultat. Le découpage temporel se présente comme une succession de phases affinant celles du découpage classique : Étude de faisabilité, Définition des besoins, Conception générale, Conception détaillée, Programmation, Intégration, Mise en œuvre. Chaque phase donne lieu à une validation officielle. Si le résultat du contrôle n'est pas satisfaisant, on modifie le livrable. En revanche, il n'y a pas de retour possible sur les options validées à l'issue de phases antérieures.



III.3.4. LE MODELE EN V

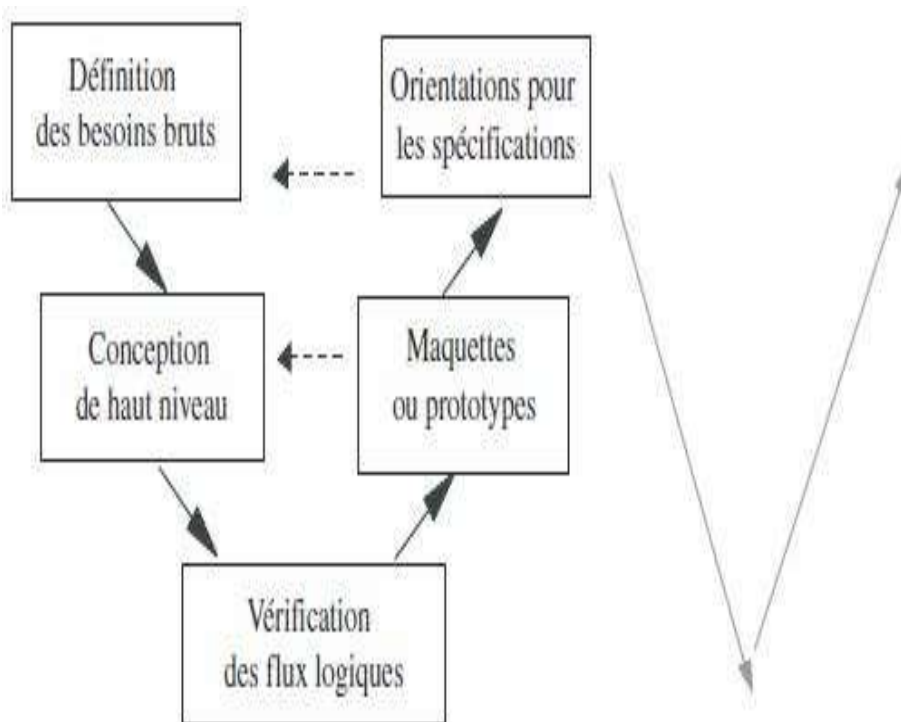
Le modèle en V est une amélioration du modèle de la cascade, qui vise à réduire ce que l'on a appelé l'« effet tunnel » : après la conception détaillée, le commanditaire du projet n'a plus de visibilité sur le projet, car les phases suivantes relèvent d'une validation technique. Ce n'est qu'au terme de la mise en œuvre que le projet ressort du tunnel. Or, les livrables ne sont pas toujours ceux attendus, non pas qu'ils ne soient pas conformes aux spécifications, mais parce celles-ci sont parfois impuissantes à décrire les attentes et la seule validation de documents est insuffisante. Le modèle en V propose de mettre en regard les phases de conception des phases de test. Dans chacune des phases de la première branche du V, on explicite les critères d'appréciation et d'acceptation du système aux étapes correspondantes de la deuxième branche du V. Ainsi, l'étude détaillée produira un jeu d'essai qui sera utilisé pour effectuer la recette fonctionnelle. L'installation sur un site pilote permettra de valider la définition fonctionnelle du besoin, d'après les critères exprimés dans cette étape. Le bilan global du projet vérifiera que les objectifs initiaux formulés dans l'étude d'opportunité ont bien été atteints.



Le modèle en V

III.3.5. LE MODELE EN W

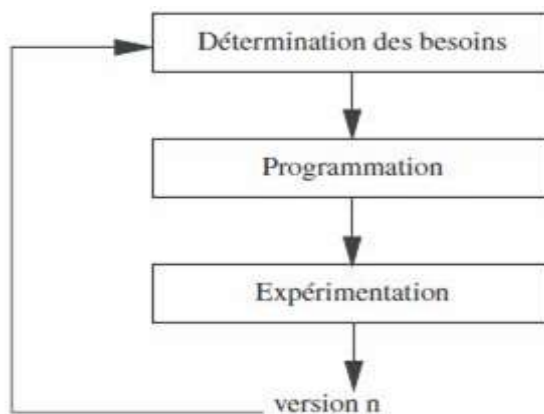
C'est un modèle qui enrichit le modèle en V dans le même esprit d'anticipation sur le livrable final. La première partie du W vise à dégager avec les clients des orientations pour la conception ou bien à explorer les possibilités d'une nouvelle technique. Le développement de maquettes ou prototypes permet une validation plus concrète des besoins, voire une expérimentation.



Le modèle en W

III.3.6. LE MODEL EVOLUTIF

Ce modèle est aussi appelé « *modèle de développement itératif* ». L'objectif de ce modèle est de construire progressivement le système de façon participative. Il repose sur l'idée que les besoins ne peuvent s'exprimer qu'après une expérimentation, même sur un système rudimentaire ou incomplet. Dans ce modèle, chaque cycle est composé de trois phases : « **Détermination des besoins – Programmation - Expérimentation** », aboutit à une nouvelle version du système : le projet s'arrête lorsque le commanditaire juge le système satisfaisant.

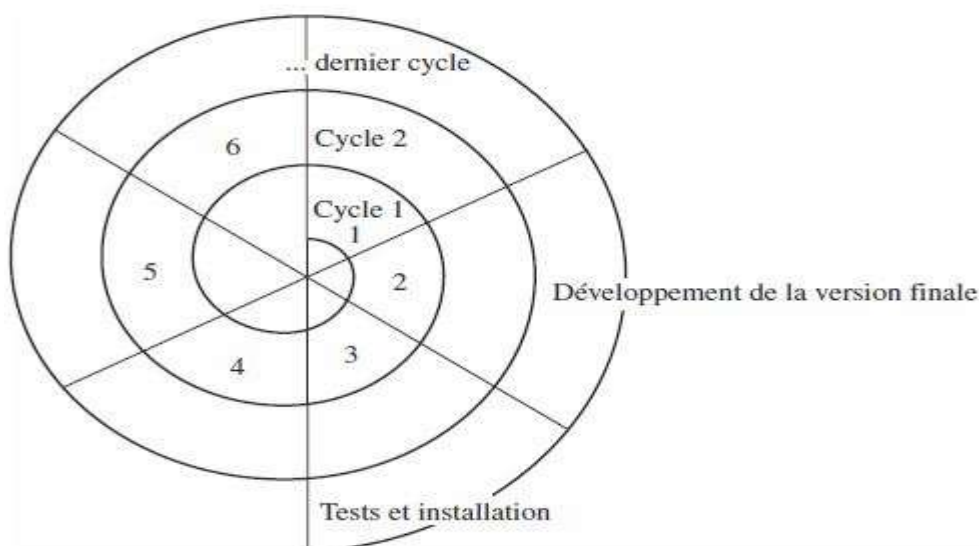


Le modèle évolutif

III.3.6. LE MODEL EN SPIRALE

Ce modèle repose sur le même principe que le modèle évolutif, mais il s'inscrit dans une relation contractuelle entre le client et le fournisseur. De ce fait les engagements et validations présentent un caractère formalisé. Chaque cycle donne lieu à une contractualisation préalable, s'appuyant sur les besoins exprimés lors du cycle précédent. Un cycle comporte six phases :

- Analyse du risque ;
- Développement d'un prototype (*modèle, archétype...*) ;
- 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 modèle en spirale

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

III.4. LA GESTION D'UNE PHASE D'UN PROJET INFORMATIQUE

D'après les normes (*ISO10006, PMBOK*), les activités de gestion de projet sont organisées comme des processus (*ensembles coordonnés d'activités aboutissant à un résultat*), classés en différents groupes, selon l'objectif : démarrage, planification, réalisation, contrôle et clôture. Pour le PMI, chaque phase fait l'objet d'une gestion complète qui met en œuvre les processus des cinq catégories à savoir :



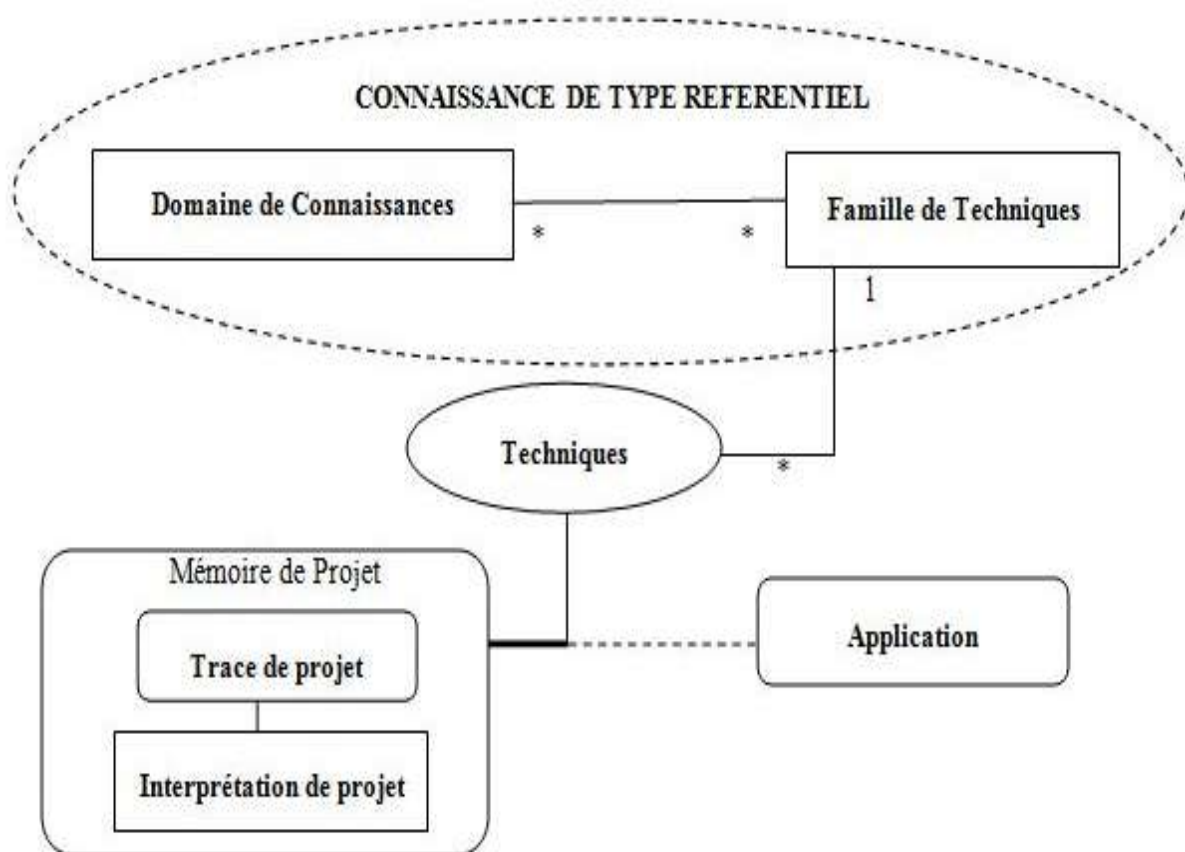
Organisation de la gestion d'une phase

1. **Les processus de démarrage** : ils ont pour but de lancer officiellement le démarrage du projet ou l'une de ses phases. Il s'agit de repérer certaines contraintes particulières (*financières, humaines, temporelles, techniques...*) et d'arrêter une stratégie générale de conduite de la phase. Lorsqu'il s'agit de la première phase du projet, le processus de démarrage comprend la désignation officielle du chef de projet.
2. **Les processus de planification** : ils visent à définir et à planifier de façon précise le travail à réaliser. Ils incluent une estimation des charges et une recherche de ressources, et ils conduisent notamment à la production d'un calendrier de travail, avec une répartition des tâches.
3. **Les processus de réalisation** : ils assurent la coordination du personnel et des autres ressources dans la réalisation du travail planifié. Ils assurent un suivi du travail réalisé et centralisent d'éventuelles demandes de changement par rapport à ce qui avait été planifié.
4. **Les processus de contrôle** : ils sont mis en œuvre périodiquement. Ils comprennent la mesure de l'avancement du projet par rapport à sa planification, la prise de mesures correctives et le traitement des demandes de changement. Ils organisent également l'acceptation officielle des livrables par le client. Une activité de contrôle peut ainsi conduire à revoir la planification ou déboucher sur la clôture de la phase.
5. **Les processus de clôture** : ils ont pour but d'officialiser l'achèvement d'une phase. Il est recommandé de formaliser l'expérience acquise sur la phase ou sur le projet.

III.5. LA CAPITALISATION DE L'EXPERIENCE D'UN PROJET INFORMATIQUE

L'expérience joue un rôle important dans la construction des connaissances sur la gestion des projets informatiques. On observe un intérêt croissant pour leur gestion à un niveau collectif. Un dispositif pour capitaliser les expériences autour des projets système d'information relève à la fois d'une mémoire de projet et d'une mémoire d'entreprise¹⁴. Les connaissances en jeu dans la gestion de projet ont trois sources :

1. *L'explicitation* : dans des ouvrages, dans des supports internes à l'entreprise, au travers une formation, par des échanges entre individus d'un ensemble de savoirs partageables par une communauté.
2. *La capture des événements* et activités lors du déroulement d'un projet.
3. *L'engagement personnel d'un individu (en général le chef de projet)* dans un projet et sa perception du déroulement et contexte du projet.



Structuration des connaissances projet

¹⁴ Voir par exemple : DIENG-KUNTZ R., CORBY O., GANDON F., GIBOIN A., GOLEBIOWSKA J., MATTA N., RIBIERE M. (2001) Méthodes et outils pour la gestion des connaissances. Une approche pluridisciplinaire du Knowledge Management, Dunod

Cela conduit à distinguer trois grands ensembles de connaissances : *des connaissances de type référentiel*, *des connaissances de type trace* et *des connaissances de type interprétation*. Les premières ont un champ plus large que le projet, les deux autres sont liées à un projet précis. Les trois ensembles de connaissances doivent être articulées. Les connaissances de type trace servent en partie à alimenter ou faire évoluer les connaissances de type référentiel, en général éclairées par les connaissances de type interprétation. À l'inverse, des connaissances de type référentiel peuvent parfois être illustrées par des connaissances de type trace, c'est-à-dire concrétisées dans des projets réels.

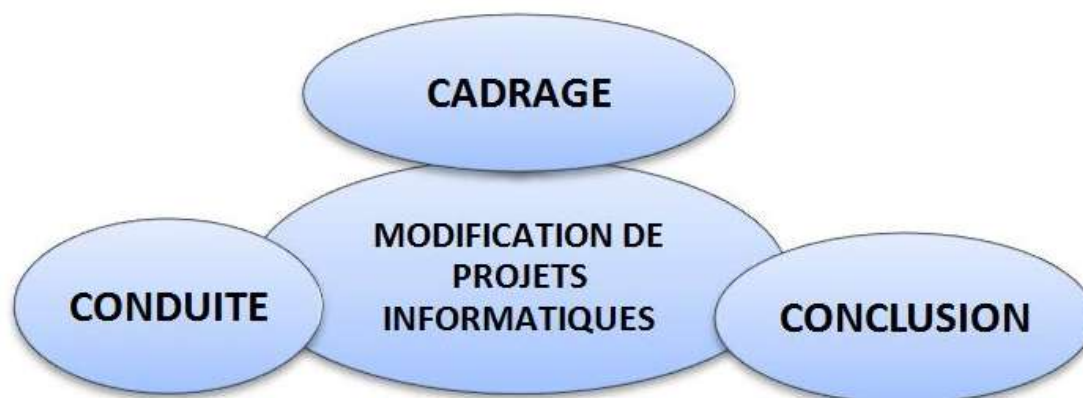
La capitalisation de l'expérience d'un projet informatique interprète 9 propriétés de connaissance pour sa bonne gestion :

- *Intégration* : Développer la charte du projet, la formulation du périmètre et du Plan. Diriger, Gérer, Suivre, Contrôler et Piloter les changements
- *Contenu* : Planification, Définition, Structure de Décomposition du Travail (*WBS*), Création, Vérification et Contrôle.
- *Délais* : Définition, Ordonnancement, Estimation de la Durée des tâches et des Ressources, Développement et Contrôle de la Planification.
- *Coût* : Planification des Ressources, Estimation des Coûts, Budgétisation et Contrôle.
- *Qualité* : Planification de la Qualité, Assurance Qualité et Contrôle Qualité.
- *Ressources Humaines* : Planification des RH, Recrutement, Développement et Gestion de l'équipe projet.
- *Communications* : Plan de Communications, Diffusion de l'information, Rapport d'Activité et de Performance, Gestion des Partenaires.
- *Risques* : Prévision et identification des Risques, Analyse des Risques (*méthodes qualitative et quantitative*), Prévision des Actions Correctrices Surveillance et Contrôle des Risques.
- *Approvisionnement* : Plan d'Acquisition et de Contractualisation, Réponses et Choix des Soumissionnaires, Administration et clôture des Pour chaque processus, activité, ou pratique est réalisé une description des produits en entrée, des outils et techniques ainsi que des produits.

III.6. LES PRINCIPALES MODIFICATIONS DE CONDUITE DE PROJETS INFORMATIQUES

Un projet se conduit en s'appuyant sur une démarche qui oscille (*tergiverse ou hésite*) régulièrement entre une approche globale située dans le sens, les objectifs et une approche analytique qui met l'accent sur la simplification afin de maîtriser tous les paramètres. Cependant, comme le simplifié peut conduire à une représentation trop réductrice du projet et donc à un résultat de mauvaise qualité, il faut absolument fonder la démarche sur des méthodes éprouvées. Certains domaines utilisent des méthodes (ex. : *Merise pour l'informatique*), des processus particuliers (ex. : *la conduite d'opération pour s'inscrire et respecter le cadre réglementaire, la programmation, les acteurs, les lots ...*).

Un projet se conduit en enchaînant des séquences, c'est un processus qui s'appuie sur un découpage en étapes, chacune se concluant par une production, un résultat livrable permettant au donneur d'ordre (DO) de : *décider de poursuivre ou non* ; et *vérifier l'adéquation entre les prévisions et les investissements*. Deux formes de découpage sont particulièrement connues et utilisées : celle de l'AFNOR qui retient quatre parties : Identification, Etude, Réalisation et Evaluation et celle dite des « **3C** » : **cadrage, conduite, conclusion**. C'est cette dernière que nous avons retenu pour sa simplicité d'utilisation :



1. le cadrage : c'est l'étape qui permet de définir les objectifs du projet, sa pertinence par rapport aux objectifs du service, ses composantes techniques, la qualité attendue, son coût et les délais. Le cadrage comporte trois phases :

- *la clarification de l'idée de départ et de l'intention* : Il s'agit là, de s'assurer de son intérêt, de son **opportunité** au regard des objectifs de service. A ce niveau le DO, après avoir validé ce document, peut rédiger la lettre de mission au chef de projet.

- *L'étude de faisabilité* : on fait d'abord ***l'Analyse fonctionnelle***¹⁵ qui détermine la démarche utile pour des projets de grande envergure, l'analyse fonctionnelle a pour but de définir plus précisément l'usage du produit, son fonctionnement et la motivation des utilisateurs. A cette fin, les éléments suivants seront déterminés, analysés, hiérarchisés :
 - les fonctions de service. Elles expriment le besoin éprouvé par le demandeur. Elles peuvent être décomposées en fonction d'un niveau inférieur en répondant à la question *comment* ?
 - les contraintes (*limites imposées au concepteur*) ;
 - l'aptitude du produit à satisfaire le client. Elle est évaluée par des indicateurs portant soit sur la fonction d'usage (*nature objective*) soit sur la fonction d'estime (*nature subjective*).

Après l'analyse fonctionnelle, c'est ***La recherche de plusieurs solutions (ou scénarios)*** : Jusqu'ici, le produit ou service, objet du projet, a été considéré dans sa globalité. Pour de gros projets, il peut devenir indispensable de réaliser un autre découpage du projet en sous-projets plus opérationnels. Les critères de ce découpage peuvent porter sur :

- les fonctionnalités du produit ;
- des sous-parties physiques ;
- des spécificités techniques ;
- les lieux de réalisation ...

Un chef de projet est alors nommé pour chaque sous-projet et la démarche reprend pour chacun au niveau de l'étude de faisabilité. La définition des scénarios doit comporter pour chacun l'analyse de leurs avantages et de leurs inconvénients, l'estimation des coûts et des délais. Dans un premier temps, la démarche s'appuie essentiellement sur des méthodes de créativité sans se préoccuper des risques, des contraintes : *quelles sont toutes les pistes envisageables ?*, Les contraintes sont prises en compte ensuite :

- *Le risque résulte-t-il du fait que le produit est nouveau, et/ou que le marché est nouveau et/ou que la technologie est nouvelle ?*
- *Quel est le meilleur plan ?*
- *Existe-t-il une (des) solutions de repli ?*
- *Quels sont les individus concernés (poseur, porteur, verrou, décideur, acheteur, payeur, ressource, voisin) ?*

¹⁵ A ce niveau, on produira un Cahier des Charges Fonctionnel (CdCF) dont la partie principale est l'énoncé fonctionnel du besoin. Ce document permet de contractualiser une logique de résultat.

Cette analyse¹⁶ permet au **DO** (*Donneur d'ordre*) de retenir la meilleure solution et de s'assurer de la faisabilité du projet. La nature du projet, son importance notamment au regard des objectifs du service permettront de définir des critères de choix qui seront alors orientés vers l'efficacité et/ou la pertinence et/ou la cohérence.

- *L'étude détaillée*¹⁷ : Cette phase vise à définir comment et par qui sera réalisé le projet. Elle décrit : les solutions techniques ; les contraintes ; la planification ; les moyens et les outils de suivi et de contrôle ; les rôles et les responsabilités des différents acteurs. La communication à réaliser autour du projet doit être définie à ce stade. Elle détaillera :
 - **la finalité, les buts de l'action de communication** : (partager les objectifs du projet ; construire une représentation partagée ; arriver à un accord, un consensus sur les différentes composantes du projet (méthodes, moyens, planning, rôles ...).
 - **une stratégie**¹⁸ : (l'identification de chaque récepteur (ou cible¹⁹) et de ses attentes ; l'adaptation du fond, de la forme des informations en fonction des cibles ; le choix du média ; une planification adaptée aux phases du projet).

Enfin, deux autres axes sont étudiés ici :

- **la maintenance** Elle peut mobiliser d'importantes ressources (*hommes, matériel, budget*) ; et
 - **le fonctionnement** en terme de coût nécessite de prévoir la durée du produit.
- 2. La conduite** : Le chef de projet constitue l'équipe projet. Jusqu'à cette étape, il a avancé dans son projet soit en s'appuyant sur des groupes de travail spécifique, soit seul. Les membres de cette équipe projet sont sollicités pour leurs compétences, leur motivation, l'entente ... Le chef de projet propose pour accord la liste des membres de cette équipe au DO. La réussite de cette étape repose sur trois voies d'action :
- *la gestion et l'organisation des ressources* : L'objectif que vise la gestion d'un projet est d'en suivre, d'en contrôler les coûts, les délais, et l'atteinte des objectifs partiels. Ces critères sont détaillés dans le plan d'action grâce à un découpage du planning (*cf. étude détaillée*) en tâches, à l'affectation des ressources, à la définition d'indicateurs et de jalons.

¹⁶ Avant le Projet Sommaire, nous avons répondu à deux questions : Quel est le but de ce projet ? Et Que réalise-t-on exactement ?

¹⁷ C'est la production du planning pour la conduite du projet

¹⁸ C'est la phase qui prépare le plan de la communication

¹⁹ Selon la cible, elle est opérationnelle, informative, promotionnelle.

- *La mise en œuvre du plan de communication* : Le plan de communication est un facteur clé de succès essentiel à la réussite du projet car il accompagne le changement induit par tout projet ; et il permet à chacun de se situer au regard des actions, des délais. La compréhension et l'adhésion au projet motivent et facilitent l'investissement en terme d'énergie et le projet devient un catalyseur d'échanges entre emplois, entre services. La diffusion de tableaux de bord, des CR, des plannings constituent un très bon vecteur d'information.
- *Le management de l'équipe* : Le chef de projet veille à entretenir la motivation, l'entente entre les membres de l'équipe ; et les compétences du collectif. Pour cela, il doit veiller à ce que :
 - la direction s'engage et démontre cet engagement ;
 - l'organisation dégage les moyens de fonctionnement, permette des délégations importantes ;
 - la culture puisse intégrer un mode de fonctionnement matriciel, une régulation basée sur l'ajustement mutuel ;
 - son management soit cohérent avec le mode projet (*donner des objectifs clairs et les faire partager, les évaluations sont collectives, les responsabilités partagées, les rôles et les procédures sont négociées ...*)

3. La conclusion : La livraison s'accompagne d'une évaluation qui porte sur la qualité, les coûts, les délais. Les indicateurs de coût et de délais ont été utilisés pour l'évaluation chemin faisant. En ce qui concerne les indicateurs de la qualité, ils ont été déterminés soit, parce que l'analyse fonctionnelle a été réalisée, soit pendant la pendant l'étude détaillée.

Le bilan final du projet doit être communiqué largement afin de porter à la connaissance de tous les agents du service le succès de l'action, de montrer le travail réalisé par les membres actifs du projet, de contribuer au changement culturel. Une capitalisation de l'expérience peut s'avérer utile pour les futurs projets. En effet, d'une part, elle met à la disposition des autres agents une source d'informations utiles et, d'autre part, cet exercice impose à l'auteur de réaliser une analyse basée sur une réflexion et une prise de recul. L'archivage du projet doit être effectué car il constitue la mémoire de ce dossier, source qui peut s'avérer indispensable en cas de litige.

QUATRIEME CHAPITRE - LES OUTILS DE GESTION DE PROJETS INFORMATIQUES

Les outils de gestion de projets sont les éléments actifs (*voies, moyens ou instruments*) qui permettent d'organiser, de piloter, de développer et de régir une activité déterminée d'un projet. Dans le cadre de ce cours, il ne sera question que de deux (2) grandes catégories d'outils de gestion de projets informatiques :

- Les méthodes de développement ;
- Les techniques de gestion.

IV.1. LES METHODES DE DEVELOPPEMENT

Une méthode définit une démarche en vue de produire des résultats. Par exemple, les cuisiniers utilisent des recettes de cuisine, les pilotes déroulent des check-lists avant de décoller, les architectes font des plans avant de superviser des chantiers. *Une méthode* permet d'assister une ou plusieurs étapes du cycle de vie du logiciel. Les méthodes d'analyse et de conception fournissent des notations standards et des conseils pratiques qui permettent d'aboutir à des conceptions « *raisonnables* », mais nous ferons toujours appel à la créativité du concepteur. Il existe différentes manières pour classer ces méthodes, dont :

- **La distinction compositionnelle / décompositionnelle** : met en opposition d'une part les méthodes ascendantes qui consistent à construire un logiciel par composition à partir de modules existants et, d'autre part, les méthodes descendantes qui décomposent récursivement le système jusqu'à arriver à des modules programmables « *simplement* ».
- **la distinction fonctionnel / orientée objet** : Dans *la stratégie fonctionnelle* un système est vu comme un ensemble d'unités en interaction, ayant chacune une fonction clairement définie. Les fonctions disposent d'un état local, mais le système a un état partagé, qui est centralisé et accessible par l'ensemble des fonctions. *Les stratégies orientées objet* considèrent qu'un système est un ensemble d'objets interagissant. Chaque objet dispose d'un ensemble d'attributs décrivant son état et l'état du système est décrit (*de façon décentralisé*) par l'état de l'ensemble. La décomposition fonctionnelle du haut vers le bas a été largement utilisée aussi bien dans de petits projets que dans de très grands, et dans divers domaines d'application. La méthode orientée objet a eu un développement plus récent. Elle encourage la production de systèmes divisés en composants indépendants, en interaction.

Dans le cadre de cours, nous distinguerons alors quatre (4) types des méthodes à savoir :

- *les méthodes fonctionnelles*, basées sur les fonctionnalités du logiciel ;
- *les méthodes objet*, basées sur différents modèles (*statiques, dynamiques et fonctionnels*) de développement logiciel.
- *Les méthodes adaptatives ou Agiles*, basées sur le changement des besoins ;
- *Les méthodes spécifiques*, basées sur les découpages temporels particuliers.

IV.1.1. LES METHODES FONCTIONNELLES

Les méthodes fonctionnelles ont pour origine la programmation structurée. Cette approche consiste à décomposer une fonctionnalité (*ou fonction*) du logiciel en plusieurs sous fonctions plus simples. Il s'agit d'une conception « top-down », basée sur le principe « *diviser pour mieux régner* ». L'architecture du système est le reflet de cette décomposition fonctionnelle. La programmation peut ensuite être réalisée soit à partir des fonctions de haut niveau (*développement « top-down »*), soit à partir des fonctions de bas niveau (développement « *bottom-up* »).

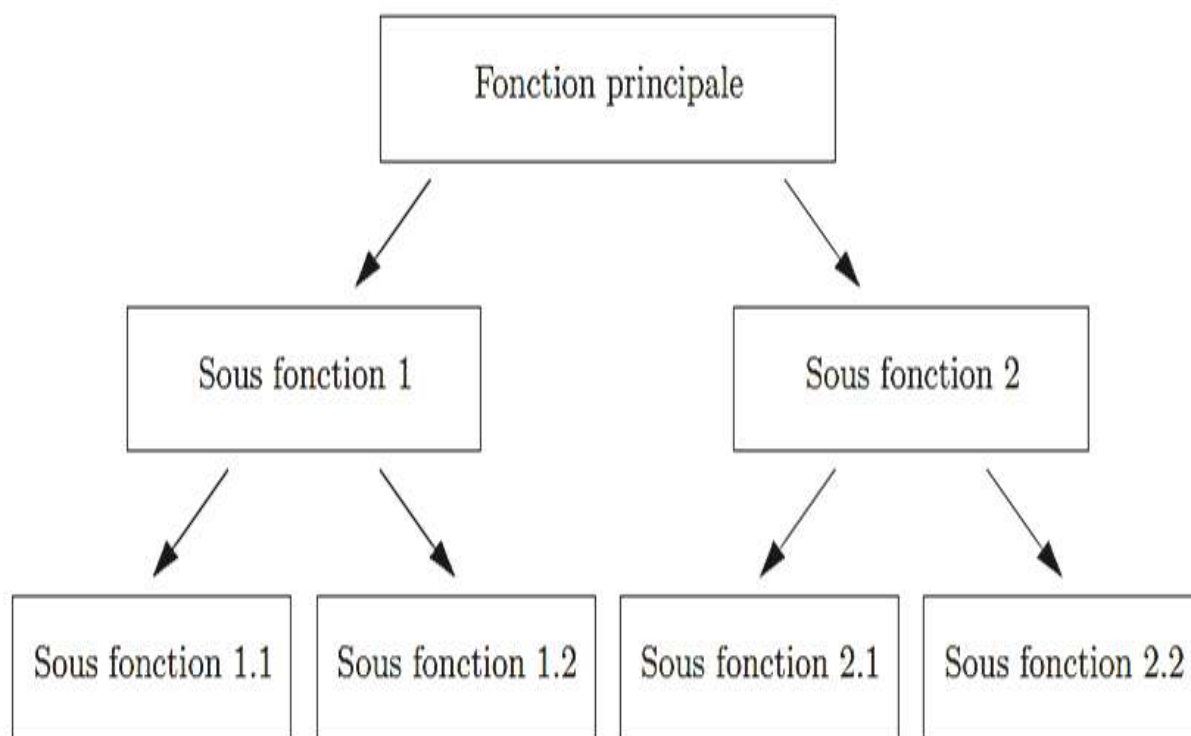


Schéma de la décomposition fonctionnelle

Cette méthode présente comme les inconvénients suivants :

- L'architecture étant basée sur la décomposition fonctionnelle, une évolution fonctionnelle peut remettre en cause l'architecture. Cette méthode supporte donc mal l'évolution des besoins.
- Cette méthode ne favorise pas la réutilisation de composants, car les composants de bas niveau sont souvent ad hoc et donc peu réutilisables.

IV.1.2. LES METHODES OBJET

Les approches objet sont basées sur une modélisation du domaine d'application. Les « *objets* » sont une abstraction des entités du monde réel. De façon générale, la modélisation permet de réduire la complexité et de communiquer avec les utilisateurs. Plus précisément un modèle :

- Aide à visualiser un système tel qu'il est ou tel qu'on le souhaite ;
- Permet de spécifier la structure ou le comportement d'un système ;
- Fournit un guide pour la construction du système ;
- Documente les décisions prises lors de la construction du système.

Ces modèles peuvent être comparés avec les plans d'un architecte : suivant la complexité du système on a besoin de plans plus ou moins précis. Pour construire une niche, on n'a pas besoin de plans, pour construire un chalet il faut un plan, pour construire un immeuble, on a besoin d'un ensemble de vues (*plans au sol, perspectives, maquettes*). Dans les méthodes objet, on distingue trois aspects :

- **Un aspect statique** : Dans lequel, on identifie les objets, leurs propriétés et leurs relations ;
- **Un aspect dynamique** : Dans lequel, on décrit les comportements des objets, en particuliers leurs états possibles et les événements qui déclenchent les changements d'état ;
- **Un aspect fonctionnel** : qui, à haut niveau, décrit les fonctionnalités du logiciel, ou, à plus bas niveau, décrit les fonctions réalisées par les objets par l'intermédiaire des méthodes.

Les intérêts des approches objet sont les suivants :

- Les approches objet sont souvent qualifiées de « *naturelles* » car elles sont basées sur le domaine d'application. Cela facilite en particulier la communication avec les utilisateurs.
- Ces approches supportent mieux l'évolution des besoins que les approches fonctionnelles car *la modélisation est plus stable, et les évolutions fonctionnelles ne remettent pas l'architecture du système en cause.*
- Les approches objet facilitent la réutilisation des composants (*qui sont moins spécifiques que lorsqu'on réalise une décomposition fonctionnelle*).

IV.1.3. LES METHODES ADAPTATIVES

Les méthodes dites « adaptatives » sont subdivisées en 2 parties notamment : *les méthodes prédictives et les méthodes agiles (adaptatives).*

IV.1.3.1. LES METHODES PREDICTIVES

Ce sont des méthodes qui correspondent à un cycle de vie du logiciel en cascade ou en V, sont basées sur une planification très précise et très détaillée, qui a pour but de réduire les incertitudes liées au développement du logiciel. Cette planification rigoureuse ne permet pas d'évolutions dans les besoins des utilisateurs, qui doivent donc être figés dès l'étape de définition des besoins.

IV.1.3.2. LES METHODES AGILES

Ce sont des méthodes qui correspondent à un cycle de vie itératif, qui considèrent que les changements (*des besoins des utilisateurs, mais également de l'architecture, de la conception, de la technologie*) sont inévitables et doivent être pris en compte par les modèles de développement. Ces méthodes privilégient la livraison de fonctionnalités utiles au client à la production de documentation intermédiaire sans intérêt pour le client.

Ainsi, Toutes les méthodes agiles prennent en compte dans leur modèle de cycle de vie trois exigences :

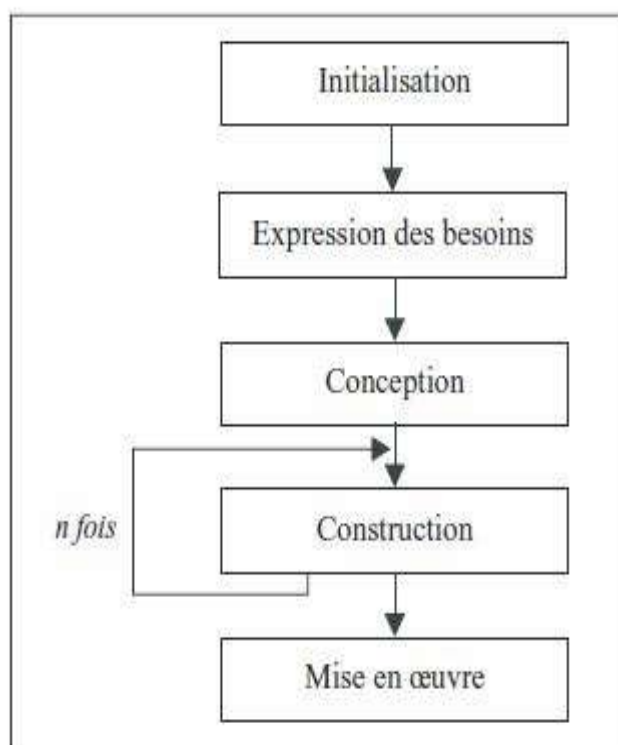
- Une forte participation entre développeurs et utilisateurs,
- Des livraisons fréquentes de logiciel et ;
- une prise en compte de possibles changements dans les besoins des utilisateurs au cours du projet.

C'est pourquoi toutes font appel, d'une façon ou d'une autre, à un modèle itératif et incrémental. De plus, elles préconisent en général des durées de cycle de vie des projets ne dépassant pas un an. Parmi les méthodes agiles, les plus usuelles ; on peut citer :

- La méthode RAD (*Rapid Application Development*) ;
- La méthode DSDM (*Dynamic Systems Development Method*);
- La méthode XP (*Programation eXtrême*);
- La méthode SCRUM.

IV.1.3.2.1. LA METHODE RAD

La méthode RAD, est un modèle linéaire (présentoir), structuré en cinq phases, et dont le modèle itératif intervient à la phase Construction du logiciel en vu de la séquencer en plusieurs modules Successivement livrés.

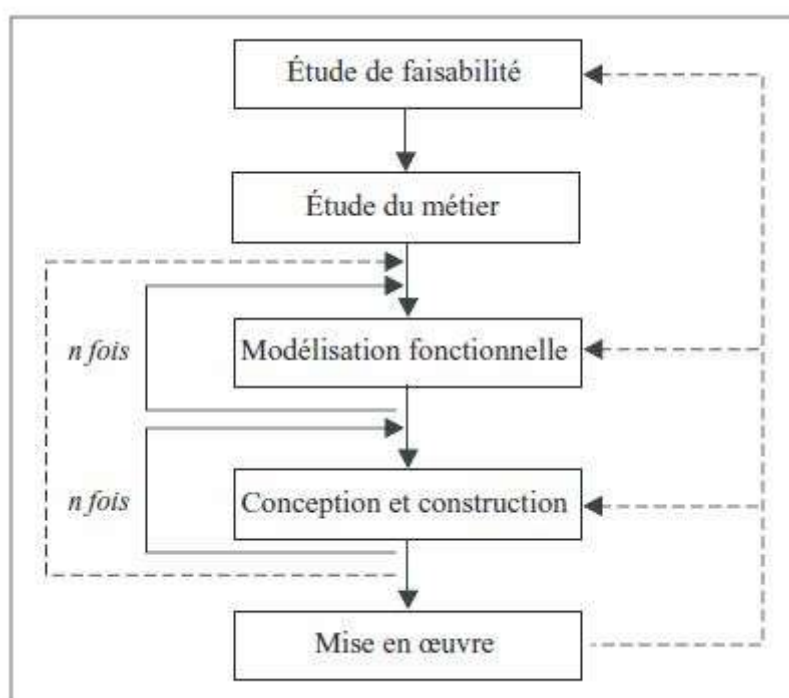


La participation des utilisateurs est placée au cœur du cycle. En effet, le déroulement d'une phase comprend une ou plusieurs sous-phases, et chaque sous-phase présente une structure à trois temps, dans laquelle la tenue d'une session participative joue un rôle central. Des travaux préparatoires rassemblent et construisent le matériau (modèle ou prototype) qui sera ensuite discuté par les différents acteurs et ajusté.

IV.1.3.2.2. LA METHODE DSDM

La méthode DSDM a évolué au cours des années. L'actuelle version distingue *le cycle de vie du système* et *le cycle de vie du projet*. Le premier comprend, outre les phases du projet lui-même, une phase de pré-projet qui doit conduire au lancement du projet et une phase post-projet qui recouvre l'exploitation et la maintenance de l'application.

Le cycle de vie du projet comprend cinq phases, dont deux sont cycliques. Les flèches pleines indiquent un déroulement normal. Les flèches en pointillé montrent des retours possibles à une phase antérieure, soit après la phase Conception et construction, soit après celle de Mise en œuvre.

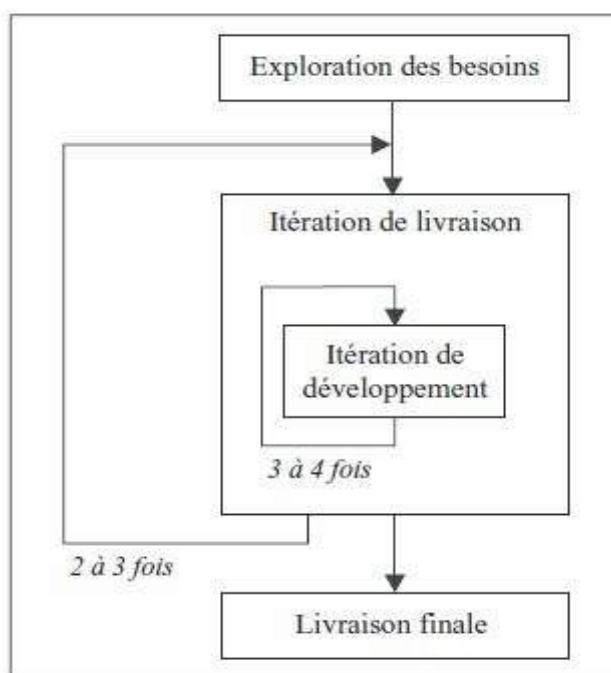


Après une Étude de faisabilité, la phase Étude du métier permet, à travers des ateliers (*workshops*) entre équipe de projet et managers, de définir le périmètre du projet, avec une liste d'exigences prioritaires et une architecture fonctionnelle et technique du futur système.

La phase Modélisation fonctionnelle est une suite de cycles. Chacun permet de définir précisément les fonctionnalités souhaitées et leur priorité. L'acceptation par toutes les parties prenantes d'un prototype fonctionnel, sur tout ou partie du périmètre, permet de passer à la phase Conception et construction. L'objectif de cette phase est de développer un logiciel testé, par des cycles successifs de développement/acceptation par les utilisateurs.

IV.1.3.2.3. LE MODELE XP

La méthode XP, focalisée sur la partie programmation du projet, propose un modèle itératif avec une structure à deux niveaux : d'abord des itérations de livraison (*release*), puis des itérations de développement. Les premières conduisent à livrer des fonctionnalités complètes pour le client, les secondes portent sur des éléments plus fins appelés scénarios qui contribuent à la définition d'une fonctionnalité.

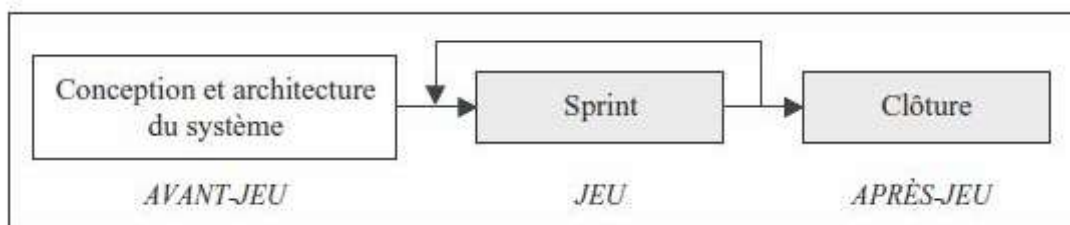


Après une phase initiale d'Exploration des besoins, un plan de livraison est défini avec le client. Chaque livraison, d'une durée de quelques mois, se termine par la fourniture d'une version opérationnelle du logiciel. Une itération de livraison est découpée en plusieurs itérations de développement de courte durée (deux semaines à un mois), chacune donnant lieu à la livraison d'une ou plusieurs fonctionnalités pouvant être testées, voire intégrées dans une version en cours.

De façon plus détaillée, chaque itération de développement commence par l'écriture de cas d'utilisation ou scénarios (*user stories*), c'est-à-dire des fonctions simples, concrètement décrites, avec les exigences associées, qui participent à la définition d'une fonctionnalité plus globale. Utilisateurs et développeurs déterminent ensemble ce qui doit être développé dans la prochaine itération. Une fonctionnalité est ainsi découpée en plusieurs tâches. Les plans de test sont écrits, les développeurs sont répartis en binôme ; ils codent les tâches qui leur sont affectées, puis effectuent avec les utilisateurs des tests d'acceptation. En cas d'échec, on revoit les scénarios et on reprend la boucle. Sinon, on continue jusqu'à avoir développé tous les scénarios retenus. Une version livrable est alors arrêtée et mise à disposition, ainsi que la documentation.

IV.1.3.2.4. LA METHODE SCRUM

La méthode SCRUM emprunte au vocabulaire du jeu le qualificatif des trois phases du cycle préconisé.



- La phase d'Avant-jeu (*pre-game*), Conception et architecture du système, se déroule de façon structurée, en général linéaire, et permet de déterminer le périmètre, la base du contenu du produit à développer et une analyse de haut niveau.
- La phase de Jeu (*game*) est itérative et qualifiée d'empirique, dans la mesure où le travail effectué ne fait pas l'objet d'une planification. Une itération, dont la durée oscille entre une et quatre semaines, est appelée un Sprint, en référence à ces poussées rapides et fortes que les joueurs de rugby peuvent effectuer sur le terrain. Un Sprint est découpé en trois sous-phases :
 - Développement (*develop*) : il s'agit de déterminer l'objectif visé au terme de l'itération, de le répartir en « paquets » de fonctions élémentaires, de développer et tester chaque paquet.
 - Emballage (*wrap*) : on referme les « paquets » et on les assemble pour faire une version exécutable. • Revue (*review*) : une revue élargie permet de faire le point sur les problèmes et l'avancement.
 - Ajustement (*adjust*) : ajusté le travail restant.
- La phase d'Après-Jeu (*postgame*), Clôture, vise à livrer un produit complet et documenté. Comme dans la première phase, on peut en planifier les tâches et les dérouler de façon linéaire.

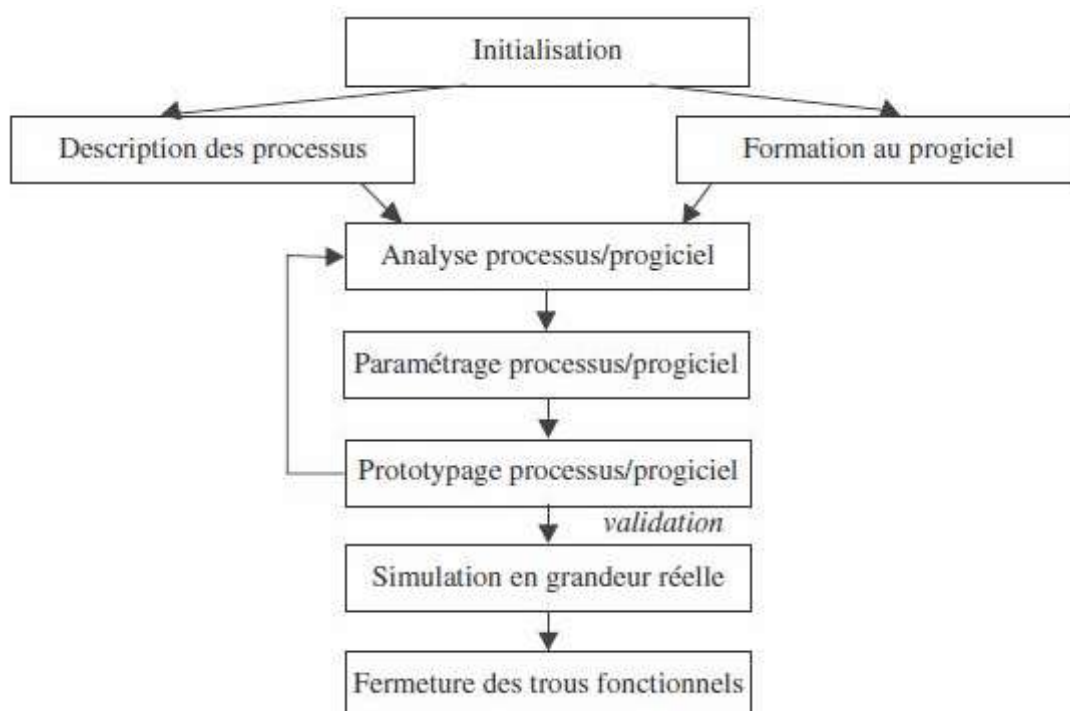
IV.1.4. LES METHODES SPECIFIQUES

Certains découpages temporels sont liés soit à une méthode, soit à un type de projet bien particulier. Nous en proposons deux exemples : le découpage préconisé pour mettre en place un progiciel intégré et le modèle RUP proposé par la société Rational Software. A ce stade, nous citerons les méthodes telles que :

- La méthode ERP ;
- La méthode RUP ;

V.1.4.1. LE CYCLE ERP

La mise en place d'un progiciel de gestion intégré, souvent appelé du terme anglo-saxon ERP (*Enterprise Resource Planning*) s'appuie sur un découpage spécifique.



En effet, il s'agit de construire, en tirant le meilleur parti du progiciel, un système améliorant la performance de l'entreprise. Deux étapes doivent donc être menées en parallèle : Description des processus et Formation au progiciel. Ensuite, il y a autant de *cycles d'analyse — paramétrage — prototypage* qu'il y a de processus. La validation par le Comité de pilotage permet une simulation en grandeur réelle. Il faut alors prendre en compte ce qui est resté en dehors du champ couvert par le progiciel.

V.1.4.2. LE MODELE RUP

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 (figure 2.17). Il peut être lu de la façon suivante :

1. Le cycle est constitué de quatre phases principales, que l'on retrouve globalement dans toutes les approches descendantes : étude préalable (opportunité), conception de la solution détaillée (élaboration), développement de la solution (*construction*) et mise en œuvre (transition).
2. Il existe six types de tâches qui, au lieu d'être affectées exclusivement à une phase, se retrouvent à des degrés divers dans chacune des phases. Par 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. L'implémentation (développement) a principalement lieu dans la phase de construction, mais on peut réaliser un prototype dès la première phase. Certaines tâches, comme la direction de projet, s'effectuent sur toute la durée du projet.
3. Certaines phases peuvent être menées de façon cyclique. Ainsi, l'élaboration se fait en deux cycles, conduisant par exemple à la production de spécifications externes (vision utilisateur) et spécifications techniques (vision développeur). La construction est itérative et incrémentale. De plus, l'ensemble du modèle représente un tour de spirale, dans le cas d'une approche globale en spirale.

IV.2. LES TECHNIQUES DE GESTION

Les techniques de gestion sont des procédés et de moyens pratiques à une activité du projet informatique en vu d'obtenir les résultats escomptés. Dans le cadre de cours, il ne sera question que :

- Les techniques de planification et ordonnancement
- Les techniques d'estimation de projets informatiques.

IV.2.1. LES TECHNIQUES DE PLANIFICATION OU DE L'ORDONNANCEMENT

La planification ou de l'ordonnancement, c'est l'organisation systématique des diverses phases d'un projet selon un plan, c'est-à-dire selon un ensemble des dispositions en vue de l'exécution d'un projet. Il s'agit d'un processus résolu *de fixation d'objectifs ; de détermination des moyens et des ressources nécessaires pour les atteindre et définition des étapes à franchir pour les réaliser.*

La réalisation d'un projet nécessite souvent une succession de tâches auxquelles s'attachent certaines contraintes :

- **De temps** : délais à respecter pour l'exécution des tâches ;
- **D'antériorité** : certaines tâches doivent s'exécuter avant d'autres ;
- **De production** : temps d'occupation du matériel ou des hommes qui l'utilisent.

Les techniques de planification ou d'ordonnancement dans le cadre de la gestion d'un projet ont pour objectif de répondre au mieux aux besoins exprimés par un client, au meilleur coût et dans les meilleurs délais, en tenant compte des différentes contraintes. L'ordonnancement se déroule en trois étapes :

- **L'élaboration d'un calendrier** : qui vise à déterminer les différentes opérations à réaliser, les dates correspondantes, et les moyens matériels et humains à y affecter.
- **L'exécution** : qui consiste à la mise en œuvre des différentes opérations définies dans la phase de planification.
- **Le contrôle** : qui consiste à effectuer une comparaison entre planification et exécution, soit au niveau des coûts, soit au niveau des dates de réalisation.

Il existe trois techniques d'ordonnement :

- Le diagramme de Gantt ;
- Les réseaux PERT (Program Evaluation and Review Techniques) ;
- La MPM (Méthode des Potentiels Métra).

IV.2.1.1. LE DIAGRAMME DE GANTT

Le diagramme de Gantt, couramment utilisé en gestion de projet, est l'un des outils les plus efficaces pour représenter visuellement l'état d'avancement des différentes activités (tâches) qui constituent un projet. La colonne de gauche du diagramme énumère toutes les tâches à effectuer, tandis que la ligne d'en-tête représente les unités de temps les plus adaptées au projet (jours, semaines, mois etc.). Chaque tâche est matérialisée par une barre horizontale, dont la position et la longueur représentent la date de début, la durée et la date de fin. Ce diagramme permet donc de visualiser d'un seul coup d'œil :

- Les différentes tâches à envisager ;
- La date de début et la date de fin de chaque tâche ;
- La durée escomptée de chaque tâche ;
- Le chevauchement éventuel des tâches, et la durée de ce chevauchement ;
- La date de début et la date de fin du projet dans son ensemble ;

En résumé, un diagramme de Gantt répertorie toutes les tâches à accomplir pour mener le projet à bien, et indique la date à laquelle ces tâches doivent être effectuées (le planning).

1. Principe

Ce type de diagramme a été mis au point par un américain Henry Gantt. On représente au sein d'un tableau, en **ligne les différentes tâches** et en **colonne les unités de temps** (exprimées *en mois, semaines, jours, heures...*). La durée d'exécution d'une tâche est matérialisée par un trait au sein du diagramme.

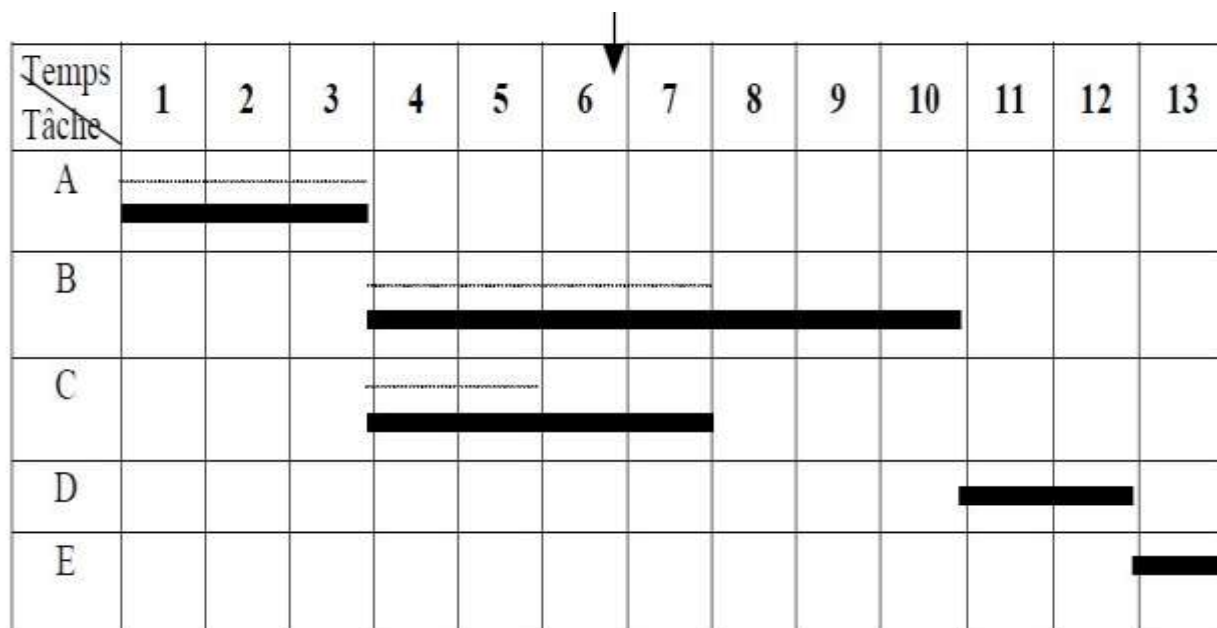
2. Réalisation.

Les différentes étapes de réalisation d'un diagramme de Gantt sont les suivantes :

- **Première étape** : On détermine les différentes tâches (ou opérations) à réaliser et leur durée.

- **Deuxième étape** : on définit les relations d'antériorité entre tâches.
- **Troisième étape** : on représente d'abord les tâches n'ayant aucune antériorité, puis les tâches dont les tâches antérieures ont déjà été représentées, et ainsi de suite...
- **Quatrième étape** : on représente par un trait parallèle en pointillé à la tâche planifiée la progression réelle du travail.

EXEMPLE :



Remarque :

- Chaque colonne représente une unité de temps.
- Les durées d'exécution prévues des tâches sont représentées par un trait épais (4 unités de temps pour C).
- Les contraintes de succession se lisent immédiatement :
 - Les tâches B et C succèdent à la tâche A.
 - D succède à B.
- Le déroulement d'exécution des tâches figure en pointillé, au fur et à mesure des contrôles. On est à la fin de la 6ème unité de temps, B est en avance d'une unité et, C est en retard d'une unité.
- On peut alors déterminer **le chemin critique** : qui est formé d'une succession de tâches, sur le chemin le plus long en termes de durées. Il est appelé chemin critique car tout retard pris sur l'une des tâches de ce chemin, entraîne du retard dans l'achèvement du projet. (Chemin critique : A, B, D, E).

Avantage :

- Permet de déterminer la date de réalisation d'un projet.
- Permet d'identifier les marges existantes sur certaines tâches (avec une date de début au plus tôt et une date au plus tard).
- La date au plus tard de début d'une tâche, la date à ne pas dépasser sans retarder l'ensemble du projet.

Inconvénient :

- Ne défait pas tous les problèmes, en particulier si l'on doit planifier des fabrications qui viennent en concurrence pour l'utilisation de certaines ressources.

Logiciels informatiques pour les diagrammes de Gantt :

- Logiciel Open Source ou logiciel libre :
 - OpenProj (Open Source)
 - Open Workbench (Open Source)
 - ProjectLibre (un fork d'OpenProj)
 - TaskJuggler (logiciel libre)
 - Teamlab (logiciel libre)
 - GanttProject
- Logiciel propriétaire :
 - Microsoft Project
 - OmniPlan
 - Primavera
 - PSNext
- Application web (logiciel en ligne) :
 - BubblePlan
 - Ganttter (peut être ajouté à Google Drive)
 - Asta Power Project
 - Sciforma
 - Redmine (*logiciel à installer sur un serveur du client*)

IV.2.1.2. LES RESEAUX PERT

La méthode PERT, acronyme de « *Programm Evaluation and Review Technique* », est une technique d'ordonnancement des tâches utilisée pour gérer les projets. Cette méthode a été mise en œuvre par la marine américaine à la fin des années 50. Au fil des évolutions, cet outil a intégré la méthode connue par le nom des « *Chemins critiques* » (*Critical Path Method, CPM*) développée un an plus tôt.

Le PERT permet d'obtenir un ordonnancement optimum des tâches les uns par rapport aux autres pour minimiser la durée totale du projet. Le PERT permet également de connaître les marges existantes sur certaines tâches (différence entre la date au + tard et la date au + tôt). A chaque tâche est associée une durée ainsi que la liste de ses antécédents (tâches qu'il faut avoir terminées pour commencer la tâche en question). Grâce à cette technique, on est en mesure d'obtenir:

- **Le chemin critique** : ensemble des tâches dont la durée a un impact direct sur la date de fin du projet. En cas de retard sur l'une de ces tâches, la date de fin du projet sera décalée.
- **La date au + tôt** : Date à laquelle chaque tâche est en mesure d'être commencée.
- **La date au + tard** : Date à laquelle chaque tâche doit nécessairement avoir démarré pour respecter la date de fin de projet envisagée.

Afin d'appliquer correctement la méthode PERT il est conseillé de mettre en place les étapes suivantes :

1. Connaître le livrable ou produit intermédiaire à l'issue de chaque étape.
2. Décrire très concrètement quelle est la tâche ou action à accomplir afin d'obtenir ce livrable. Cette définition de tâche doit comporter sa durée, les moyens nécessaires, les ressources et les contraintes à prendre en compte pour l'accomplir dans son intégralité.
3. Identifier la connexion entre les différentes tâches. Ces connexions peuvent être du type « fin à début », « début à fin », « fin à fin » ou « début à début ». Deux tâches peuvent cependant se trouver complètement indépendantes au sein d'une analyse PERT.
4. Construire le réseau PERT et identifier le chemin critique.
5. Réaliser l'affectation de ressources à chacune des tâches.
6. Faire le calcul des coûts induits par chaque tâche.

Actuellement, la représentation sous forme *d'un planning de Gantt* est plus souvent utilisée. Cette représentation a l'avantage d'être beaucoup plus lisible que le réseau PERT car la durée des tâches est représentée par des barres plus ou moins longues, proportionnelles à leurs durées. De plus, ces durées se trouvent associées à une barre calendaire. Il est possible d'indiquer la relation entre les tâches à l'aide de flèches. Il existe sur le marché plusieurs logiciels de planification permettant de faire gagner du temps au chef de projet. De cette manière, le chef de projet pourra se consacrer à la définition des tâches, à leur durée, aux ressources associées. La représentation graphique, le calcul du chemin critique, le calendrier, la comptabilisation des ressources, etc., sont réalisés par le logiciel.

1. Principaux généraux :

Si dans le vocabulaire de tous les jours, la notion de «projet» désigne assez globalement « une action future », cette notion renvoie par contre à une formulation beaucoup plus précise pour tous les acteurs impliqués dans le déroulement opérationnel d'un projet. *Un graphe de dépendances* est utilisé. Pour chaque tâche, sont indiquées une date de début et de fin au plus tôt et au plus tard. Le diagramme permet de déterminer le chemin critique qui conditionne la durée minimale du projet.

Le but est de trouver la meilleure organisation possible pour qu'un projet soit terminé dans les meilleurs délais, et d'identifier les tâches critiques, c'est-à-dire les tâches qui ne doivent souffrir d'aucun retard sous peine de retarder l'ensemble du projet. Ainsi, les règles et notations de représentation sont :

- *La tâche :*



- *La tâche fictive :*



- *L'étape :*



Ainsi, Dans un graphe PERT :

- Chaque tâche est représentée par *un arc*, auquel on associe un chiffre entre parenthèses qui représente la durée de la tâche.
- *Entre les arcs figurent des cercles appelés « sommets » ou « événement »* qui marquent l'aboutissement d'une ou plusieurs tâches. Ces cercles sont numérotés afin de suivre l'ordre de succession des divers événements.

2. Construction d'un réseau PERT :

Le recours au PERT suppose qu'aient préalablement été identifiées les différentes tâches nécessaires à la réalisation d'un projet, leur durée et leurs relations d'antériorité. Généralement ces informations sont synthétisées dans un tableau du type suivant :

Tâches	Durée	Antériorité(s)
A	2	-
B	4	-
C	4	A
D	5	A,B
E	6	C,D

Conventions de base d'un réseau PERT

Le PERT permet de représenter l'ensemble des tâches sur un graphe orienté, à partir duquel il sera possible d'identifier leurs dates au plus tôt et au plus tard et de calculer leurs marges. Un graphe orienté est un réseau composé d'une entrée et d'une sortie, ainsi que de points (*appelés « sommets »*) reliés entre eux par des flèches (*appelées « arcs »*).

Les principales conventions d'un réseau PERT sont les suivantes :

- chaque tâche est symbolisée par un arc, auquel est associée une valeur numérique correspondant à sa durée.
- les sommets auxquels aboutissent les arcs correspondent donc à des étapes, qui marquent l'aboutissement d'une ou plusieurs tâches.
- chaque étape est identifiée par un numéro d'ordre et renseignée sur la date à laquelle elle peut être atteinte au plus tôt ("date au plus tôt") et au plus tard ("date au plus tard") pour respecter le délai optimal de réalisation du projet.

- le graphe possède une entrée (*sommet sans antécédent*) et une sortie (*sommet sans descendant*) qui correspondent respectivement aux étapes « *Début des opérations* » et « *Fin des opérations* ».

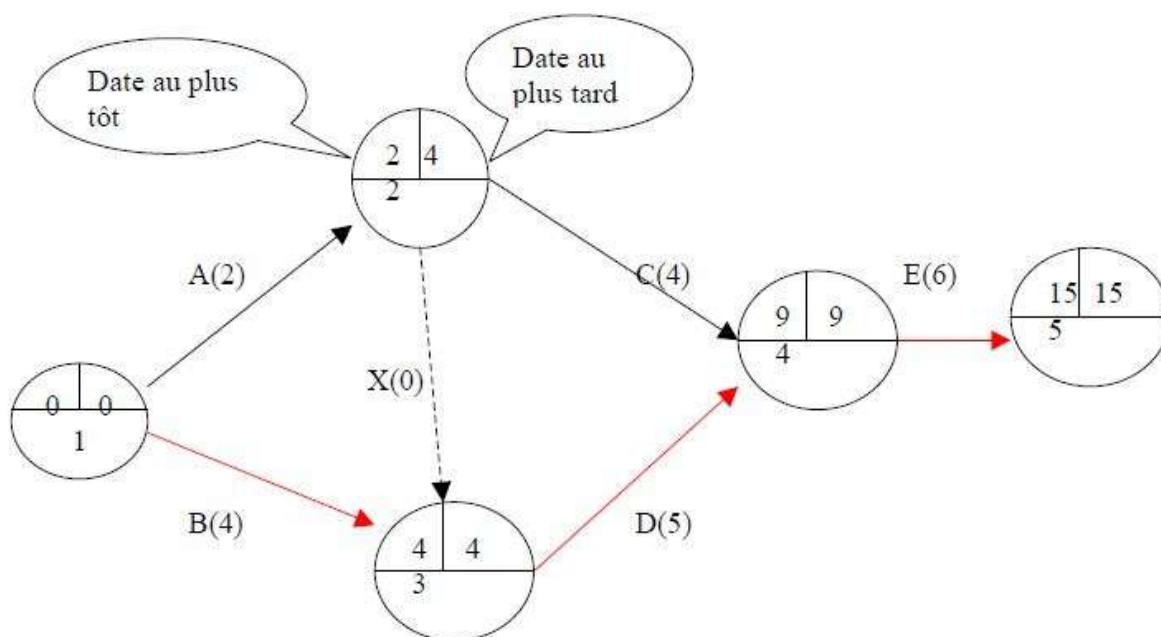
Du fait de ses conventions, il est parfois nécessaire d'introduire des "tâches fictives" pour traduire correctement sur un graphe les relations d'antériorité de certaines tâches, notamment lorsque celles-ci partagent avec d'autres une partie de leurs antécédents.

Construction d'un graphe PERT

Sur la base des conventions précédentes, la construction d'un graphe PERT ne pose pas de difficulté particulière, mais doit être réalisée avec méthode. La démarche la plus appropriée consiste à procéder par « *niveau* » :

- Déterminer les tâches sans antécédent (*tâches de niveau 1*) et les relier à l'étape de « *Début* » ;
- Identifier ensuite les tâches de niveau 2, c'est-à-dire celles dont les antécédents sont exclusivement du niveau 1 et les positionner sur le graphique en fonction de des derniers ;
- ... Continuer ainsi, jusqu'à ce que toutes les tâches aient pu être positionnées entre elles et relier celles n'ayant pas de descendant à l'étape de « *Fin* ».

Ainsi, si l'on reprend le tableau d'antériorité proposé précédemment (projet Y) :



Remarque :

- Il a été nécessaire d'introduire une tâche fictive de durée égale à 0, pour représenter la relation d'antériorité entre A et D.
- Le cumul des tâches composant la séquence la plus longue (B, D, E) permet de déterminer la date au plus tôt de réalisation du projet. Cette succession de tâches constituent le chemin critique.

Lecture d'un graphe PERT

Le graphe se lit de gauche à droite (de l'étape « DÉBUT » à celle de « FIN »). Chaque arc symbolise une tâche qui permet d'atteindre une nouvelle étape dans la réalisation du projet. Une nouvelle tâche ne peut commencer que lorsque toutes les tâches préalables à sa réalisation sont terminées.

Chaque sommet correspond à une étape qui est identifiée par une cartouche où sont précisés : son « *numéro d'ordre* », la date à laquelle elle peut être atteinte au plus tôt (« *date au plus tôt* ») et la date à laquelle elle doit être atteinte au plus tard pour respecter le délai optimal de réalisation du projet (« *date au plus tard* »).

DÉTERMINATION DES DATES « AU PLUS TÔT » ET « AU PLUS TARD » DANS UN RÉSEAU PERT

1. La date au plus tôt d'un réseau PERT correspond à la date à laquelle une étape peut être atteinte au plus tôt. Elle s'obtient en ajoutant à la date au plus tôt de l'étape précédente, la durée de la tâche qui les sépare :

$$\text{Date au plus tôt « étape } j \text{ »} = \text{Date au plus tôt « étape } i \text{ »} + \text{Durée tâche « } ij \text{ »}.$$

Lorsque plusieurs arcs arrivent à un même sommet (*c'est à dire que plusieurs tâches doivent être réalisées pour atteindre une étape donnée*), il convient de faire ce calcul pour toutes les tâches menant à l'étape en question et de retenir comme "date au plus tôt" de l'étape le maximum des valeurs ainsi trouvée (*en effet, l'étape ne sera vraiment atteinte que lorsque toutes les tâches y menant auront été accomplies*) :

$$\text{Date au plus tôt « étape } j \text{ »} = \text{Max. (Date plus tôt « étapes } i \text{ »} + \text{Durée tâches « } ij \text{ »)}$$

Dans cette formule, "i" représente l'ensemble des tâches immédiatement antérieures à "j".

La détermination des dates au plus tôt des différents sommets se fait donc par calculs successifs, à partir de l'étape initiale « *Début* » (dont, par convention, la date au plus tôt est fixée à 0). La durée minimale du projet correspond donc à la date au plus tôt de l'étape « *Fin* ».

2. La date au plus tard d'un réseau PERT correspond à la date à laquelle une étape doit être atteinte au plus tard pour que la durée globale du projet reste minimum. Elle s'obtient en retirant de la date au plus tard de l'étape qui lui succède la durée de la tâche qui les relie :

$$\text{Date au plus tard « étape } i \text{ »} = \text{Date plus tard « étape } j \text{ »} - \text{Durée tâche « } ij \text{ »}.$$

Lorsque plusieurs arcs partent d'un même sommet (*c'est à dire que plusieurs tâches commencent à partir d'une même étape*), il convient de faire ce calcul pour toutes les tâches (*y compris s'il s'agit de tâches fictives*) succédant à l'étape en question et de retenir comme « *date au plus tard* » de l'étape le minimum des valeurs ainsi trouvées :

$$\text{Date au plus tard « étape } i \text{ »} = \text{Min. (date au plus tard « étapes } j \text{ » - Durée tâches « } ij \text{ »)}$$

Dans cette formule, « *j* » représente l'ensemble des tâches immédiatement postérieures à « *j* »

Ainsi, dans notre exemple précédent (projet Y), la date au plus tard de l'étape :

$$I = \text{Min} [(9 - 4), (4 - 0)] = 4$$

La détermination des dates au plus tard des différents sommets se fait donc à rebours du graphe, par calculs successifs, en partant de l'étape finale « *Fin* » (pour laquelle, par convention, on considère que la date au plus tard est égale à sa date au plus tôt).

On appelle **chemin critique** la succession des tâches pour lesquels aucun retard n'est possible sans remettre en cause la durée optimale du projet (*tâches pour lesquelles date au plus tôt = date au plus tard*). Dans notre exemple, celui-ci est indiqué en rouge.

CALCUL DES DIFFÉRENTES MARGES D'UNE TÂCHE DANS UN RÉSEAU PERT

On appelle « **marge** » d'une tâche le retard qu'il est possible de tolérer dans la réalisation de celle-ci, sans que la durée optimale prévue du projet global en soit affectée. Il est possible de calculer trois types de marges :

- La **marge totale** d'une tâche indique le retard maximal que l'on peut admettre dans sa réalisation (*sous réserve qu'elle ait commencé à sa date au plus tôt*) sans allonger la durée optimale du projet. Elle se calcule en retirant la durée de la tâche en question à l'écart qu'il peut y avoir entre sa date au plus tôt de début et sa date au plus tard de fin :

Marge totale tâche « ij » = Date au plus tard « étape j » - Date au plus tôt « étape i » - Durée tâche « ij ».

Ainsi, dans notre exemple précédent (projet Y) :

- Marge totale de A = $(4 - 0 - 2) = 2$
- Marge totale de C = $(9 - 2 - 4) = 3$

Sauf le cas particulier, un retard correspondant à la marge totale d'une tâche se traduit par une modification des dates au plus tôt des tâches qui lui succèdent et entraîne, généralement, l'apparition d'un second chemin critique. Il n'est donc pas possible de cumuler des retards correspondant à leur marge totale sur plusieurs tâches successives, sans remettre en cause la durée optimale prévue pour le projet.

- La **marge libre** d'une tâche indique le retard que l'on peut admettre dans sa réalisation (sous réserve qu'elle ait commencé à sa date au plus tôt) sans modifier les dates au plus tôt des tâches suivantes et sans allonger la durée optimale du projet. Elle se calcule en retirant la durée de la tâche en question à l'écart qu'il peut y avoir entre ses dates au plus tôt de début et de fin :

Marge libre tâche « ij » = Date au plus tôt « étape j » - Date au plus tôt « étape i » - Durée tâche « ij ».

Ainsi, dans notre exemple précédent (projet Y) :

- Marge libre de A = $(2 - 0 - 2) = 0$
- Marge libre de C = $(9 - 2 - 4) = 3$

Un retard correspondant à la marge libre d'une tâche reste sans conséquence sur les marges des tâches qui lui succèdent. Il est donc possible de cumuler des retards, s'inscrivant dans leur marge libre, pour plusieurs tâches successives, sans remettre en cause la durée optimale prévue pour le projet.

- La **marge certaine** d'une tâche indique le retard que l'on peut admettre dans sa réalisation (quelle que soit sa date de début) sans allonger la durée optimale du projet. Elle se calcule en retirant la durée de la tâche en question à l'écart qu'il peut y avoir entre sa date au plus tard de début et sa date au plus tôt de fin :

Marge certaine tâche « ij » = Max [0, (Date au plus tôt « étape j » - Date au plus tard « étape i » - Durée tâche « ij »)].

D'après cette formule, la marge certaine est considérée comme nulle lorsque son calcul donne un nombre négatif

Ainsi, dans notre exemple précédent (projet Y) :

- Marge certaine de A = $\text{Max } [0, (2 - 0 - 2)] = 0$
- Marge certaine de C = $\text{Max } [0, (9 - 5 - 4)] = 0$

Un retard correspondant à la marge certaine d'une tâche reste sans conséquence sur les marges des tâches qui lui succèdent, même si elle commence à sa date au plus tard. Il est donc possible de cumuler des retards, s'inscrivant dans leur marge certaine, pour plusieurs tâches successives, même si elles commencent à leur date au plus tard, sans remettre en cause la durée optimale prévue pour le projet. On remarque que l'ensemble des marges des tâches composant le **chemin critique** sont nécessairement nulles, puisqu'il s'agit de tâches pour lesquels, par définition, aucun retard n'est possible sans remettre en cause la durée optimale prévue pour le projet.

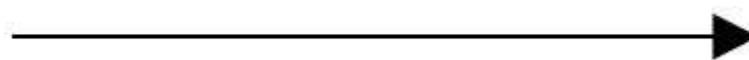
IV.2.1.3. LA MPM (METHODE DE POTENTIELS METRA)

Cette méthode a été développée par une équipe de chercheurs français. A l'identique de réseau PERT, cette outil permet de réduire la durée totale d'un projet. On étudie les délais sans prendre en compte les charges et les moyens disponibles. La méthode est une représentation graphique qui permet de bâtir un réseau. Ce réseau est constitué par des taches (ou étapes) :

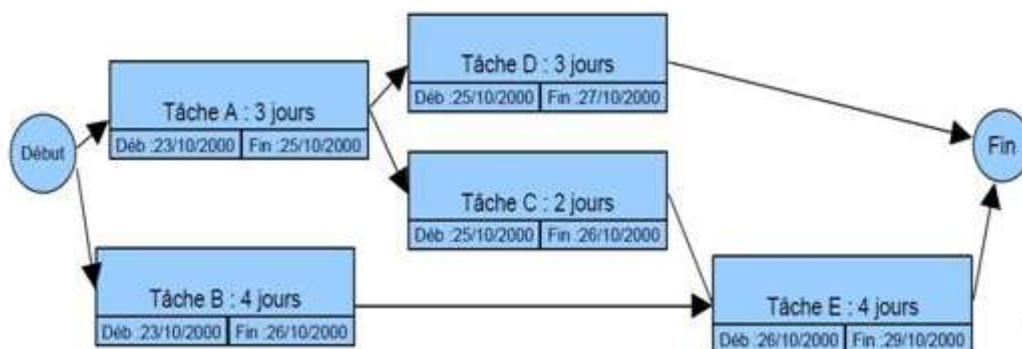
- **Tâche** : Déroulement dans le temps d'une opération. La tache est pénalisante car elle demande toujours une certaine durée, des moyens (ou ressources) et coute de l'argent. Contrairement au réseau PERT, ici elle est symbolisée par un rectangle dans lequel seront indiqués l'action à effectuer et le temps estime de réalisation de cette tache, la date de début et de fin.

Début : 28/01/2019	Fin : 31/01/2019
Tâche A : 4 jours	

- **Liaison orientées** : Elles représentent les contraintes d'antériorités des tâches.



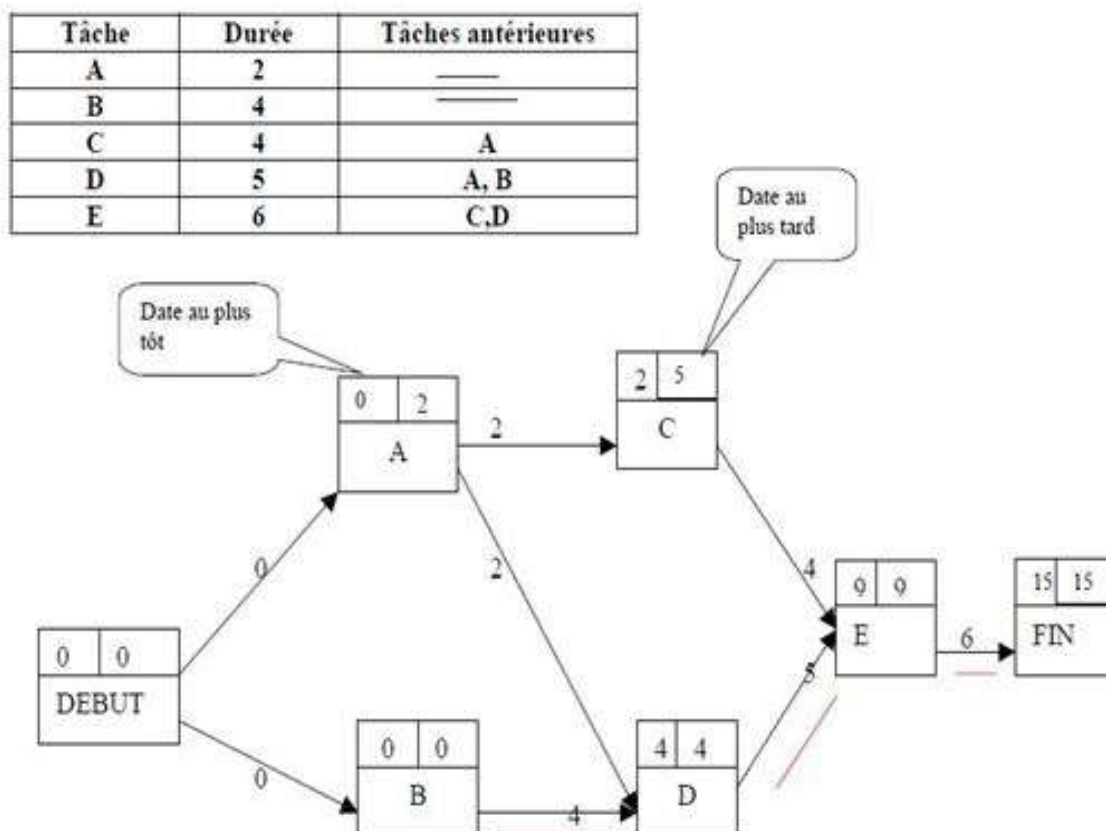
EXEMPLE :



La méthode MPM comme le réseau PERT a pour but de planifier la durée d'un projet, aussi nous devons mener des calculs sur le graphe afin d'en déduire des renseignements sur son exécutabilité.

1. Principe :

- Les tâches sont représentées par des sommets et les contraintes de succession par des arcs.
- Chaque tâche est renseignée par la date à laquelle elle peut commencer (date au plus tôt) et celle à laquelle, elle doit se terminer (date au plus tard).
- chaque arc est associé une valeur numérique, qui représente soit une durée d'opération, soit un délai. Exemple :



Remarque :

- **La date de début au plus tôt d'une tâche** est obtenue en cumulant la durée des tâches qui précèdent sur la séquence la plus longue. On initialise le sommet DEBUT avec une date au plus Tôt = 0.

Date au plus tôt de la tâche $j = \text{Max}(\text{date au plus tôt de } i + \text{Durée } T_{i,j})$ pour tous les prédécesseurs i de j .

- Les dates au plus tard : dates à laquelle doivent être exécutées les tâches sans remettre en cause la durée optimale de fin du projet. On initialise à l'étape terminale, le dernier sommet par la date au plus tard = date au plus tôt.

Date au plus tard $i = \text{Min}(\text{Date au plus tard de } j - \text{durée } T_{i,j})$ pour tous les successeurs j de i .

- On peut alors déterminer **le chemin critique** : succession de tâches sur le chemin le plus long au sens des durées. Pour toutes les tâches du chemin critique, les dates au plus tôt et au plus tard coïncident. Chemin critique : B, D, E.

2. La marge totale :

La marge totale sur une tâche est le retard que l'on peut prendre dans la réalisation de cette tâche sans retarder l'ensemble du projet. *Elle est obtenue, en faisant pour chaque tâche, la différence entre la date au plus tard de début d'une tâche et la date au plus tôt :*

$$\text{Marge totale sur } A = (2-0)=2$$

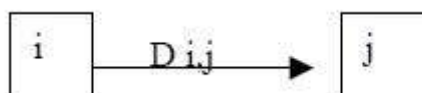
3. La marge libre :

La marge libre sur une tâche est le retard que l'on peut prendre dans la réalisation d'une tâche sans retarder la date de début au plus tôt de toute autre tâche qui suit. Si on appelle :

T_j la date au plus tôt de la tâche qui suit la tâche considérée.

T_i La date de début au plus tôt de la tâche i .

$D_{i,j}$ La durée de la tâche i .



Marge Libre de $i = \text{Min}(T_j - T_i - D_{i,j})$ pour tous les successeurs j de i .

$$\text{Marge libre sur } A = 2 - 0 - 2 = 0$$

$$\text{Marge libre sur } C = 9 - 2 - 4 = 3$$

IV.2.2. LES TECHNIQUES D'ESTIMATION DE PROJETS INFORMATIQUES.

L'estimation des projets informatiques est l'une des plus importantes activités du développement de logiciels. La planification rigoureuse et le pilotage du projet ne sont pas envisageables en absence d'une estimation sérieuse et fiable. En règle générale, notre industrie du logiciel ne sait pas estimer correctement les projets et n'utilise pas convenablement les estimations. Nous souffrons de ces conséquences et nous devons focaliser nos efforts sur l'amélioration de la situation. La sous-estimation d'un projet entraîne :

- Un sous-effectif, provoquant la surchauffe de l'équipe ;
- Une sous-appréciation de la charge d'assurance qualité, avec le risque de livrables de médiocre qualité ;
- L'établissement d'un planning trop serré, qui dégradera votre crédibilité, lorsque ces délais présomptueux sont largement dépassés.

Pour ceux qui pensent éviter cette situation en gonflant l'estimation, la surestimation d'un projet peut s'avérer aussi néfaste pour l'Organisme. Si vous accordez à un projet plus de ressources que nécessaires sans contrôler l'utilisation de ces ressources, le projet :

- Coûtera beaucoup plus cher (en grevant le bilan du projet) ;
- Durera plus longtemps que nécessaire (en manquant les opportunités ciblées) ;
- Diffèrera la disponibilité de vos ressources pour le prochain projet.

IV.2.2.1. ESTIMATION DE CHARGES DE PROJETS INFORMATIQUES

L'estimation d'un projet informatique comprend quatre étapes :

- Estimer la taille du produit à développer. Celle-ci se mesure généralement en nombre d'instructions (lignes de code) ou en points de fonction, mais il existe d'autres unités de mesure possibles. Une comparaison des avantages et des inconvénients de chacune de ces mesures est abordée dans les références bibliographiques données en fin de l'article.
- Estimer la charge en mois hommes ou en jours hommes ;
- Construire le calendrier du planning ;
- Estimer le coût du projet en monnaie locale ;

A. ESTIMATION DE LA TAILLE

Le premier stade d'une opération d'estimation consiste à estimer, le plus précisément possible, la taille du logiciel à développer. Les sources d'information, relatives au périmètre du projet, naissent avec une description formelle des besoins1 (*spécification des besoins des clients, appel d'offres, spécification du système, spécification des exigences du logiciel*).

Lors de la réestimation du projet dans les phases ultérieures du cycle de vie, les documents de conception vous fourniront des détails additionnels. Ne prétextez pas du manque de description formelle pour vous abstenir de faire une première estimation du projet. Une description verbale, une présentation succincte au tableau noir sont quelquefois les seules données concrètes pour démarrer. Dans tous les cas, vous devez informer toutes les parties concernées du niveau de risque et d'incertitude de l'estimation. De plus, vous devrez réestimer le projet dès que les limites du périmètre se préciseront. Les 2 principaux moyens d'estimation de la taille de l'ouvrage sont :

1. l'analogie. Si vous avez déjà fait un projet similaire dont vous connaissez la taille, vous estimerez chaque partie principale du nouveau projet comme un pourcentage de la taille de la partie similaire du précédent projet. Vous estimerez la taille totale d'un nouveau projet en cumulant les estimations des tailles de toutes les parties. Un estimateur chevronné peut produire des estimations convenables, par analogie, s'il connaît les valeurs précises des tailles des parties d'un projet précédent et si le nouveau projet est suffisamment voisin de ce précédent.
2. La comptabilisation des caractéristiques quantitatives de l'ouvrage. On peut utiliser une approche algorithmique telle que celle des points de fonction pour convertir le total en une mesure de la taille. Les caractéristiques globales incluent le nombre de sous-systèmes, de classes / modules, de méthodes / fonctions. Des caractéristiques plus détaillées incluent le nombre d'écrans, de dialogues, de fichiers, de tables, de rapports édités, de messages etc.

B. ESTIMATION DE LA CHARGE

Après avoir estimé la taille de l'ouvrage à produire, vous pouvez en déduire l'estimation de la charge. Cette conversion de la taille du logiciel en charge totale du projet ne peut s'envisager qu'après définition d'un cycle de vie de développement du logiciel et définition d'un processus de développement de la solution pour spécifier, concevoir, réaliser et tester le logiciel.

La réalisation d'un projet de développement de logiciel implique plus qu'un simple codage du logiciel ; car le codage ne représente souvent qu'une faible partie de la charge. Écrire et peaufiner la documentation, réaliser des prototypes, concevoir les livrables, revoir et tester le code, représentent la part prépondérante de la charge totale du projet. L'estimation de la charge du projet exige d'identifier, d'évaluer, et d'additionner les travaux que vous devez accomplir pour construire un ouvrage de la taille estimée. Il existe deux manières de déduire la charge à partir de la taille :

1. La meilleure façon est d'utiliser l'historique de votre Organisme pour recenser les charges réelles consommées par les précédents projets pour réaliser les ouvrages. Ceci suppose évidemment :
 - que votre Organisme ait documenté les résultats réels des précédents projets ;
 - que vous ayez réalisé, au moins, un projet de taille équivalente - disposer de plusieurs projets de taille équivalente, renforce la conviction qu'il existe une relation stable entre la taille d'un ouvrage et la charge nécessaire à sa réalisation;
 - que vous suivez un cycle de vie de développement similaire en utilisant la même méthodologie et les mêmes outils, grâce à une équipe qui possède les mêmes compétences et les mêmes expériences.
2. Il se peut que vous ne disposiez pas d'un historique utilisable, parce que votre Organisme n'a pas encore commencé à le constituer ou parce que ce nouveau projet est nettement différent des précédents sur un ou plusieurs aspects fondamentaux. Vous pouvez appliquer une approche algorithmique reconnue telle que le modèle COCOMO de Barry BOEHM ou la méthodologie de PUTNAM pour convertir une estimation de taille en estimation de charge. Ces modèles ont été élaborés en étudiant un nombre significatif de projets terminés par divers Organismes, pour en extraire la relation entre les tailles et les charges. Ces modèles, issus des données de l'industrie du logiciel, peuvent ne pas être aussi précis que ceux de votre historique, mais ils vous donneront, toutefois, une première approche des estimations de charges.

C. ESTIMATION DES DELAIS

La troisième étape de l'estimation consiste à déterminer les délais à partir de la charge estimée. Ce qui implique généralement d'estimer les ressources affectées au projet (*la Structure de Contribution*) ce qu'elles devront faire (*le WBS – Work Breakdown Structure – Organigramme des Tâches*) quand elles commenceront à travailler au projet et quand elles le termineront. Lorsque vous aurez ces informations, vous devez planifier les tâches.

À nouveau, les historiques des projets passés, réalisés par votre Organisme ou, à défaut, des modèles classiques, peuvent être utilisés pour déterminer le nombre de personnes dont vous aurez besoin pour un projet d'une taille donnée et pour ordonnancer ces travaux. Si vous n'avez rien d'autre, la formule²⁰ empirique suivante vous donnera une idée du temps total requis :

$$\text{Délai en mois} = 3,0 * (\text{charge en mois})^{1/3}$$

Des opinions diverses proposent au lieu de 3,0 des coefficients variant de 2,0 à 4,0. Ce n'est qu'en procédant à des essais que vous trouverez le bon coefficient applicable à vos propres travaux.

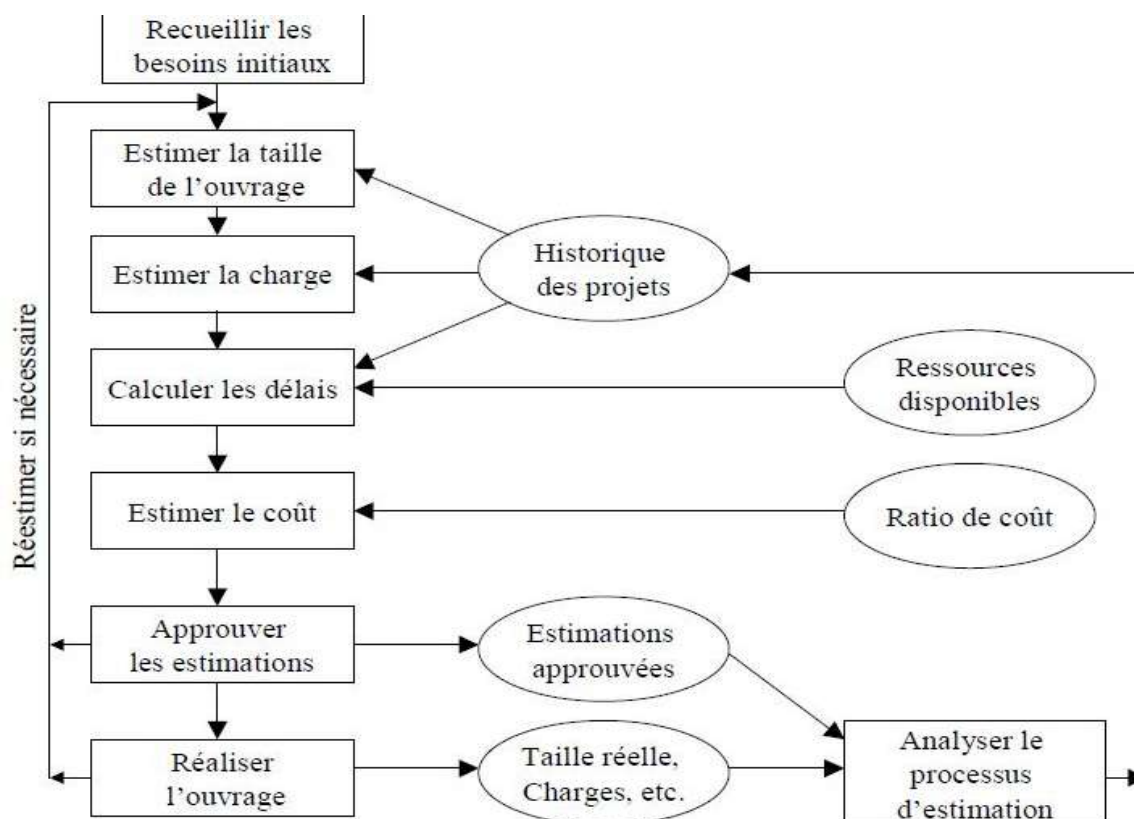
D. ESTIMATION DU COUT

Il faut prendre en compte de nombreux facteurs pour estimer le coût total d'un projet. Ces facteurs incluent les charges des travaux, les acquisitions ou les locations de matériels et de logiciels, les frais de déplacements (réunions et essais) les télécommunications (*appels à longue distance, vidéoconférences, lignes dédiées aux tests, etc.*) les formations, les frais de locaux etc.

L'estimation exacte du coût total du projet dépend de la façon dont votre Organisme affecte les coûts. Au lieu d'être affectés aux projets, certains coûts peuvent être pris en compte en les intégrant dans les taux horaires (*en euros par heure*). Souvent, un Directeur de projet estimera seulement le coût du travail et n'identifiera que les coûts additionnels qui ne sont pas considérés, par l'Organisme, comme des frais généraux. Le coût du travail peut être obtenu en multipliant, simplement, l'estimation de charge en heures par un taux en euros par heure).

Un calcul plus précis de coût de la charge résulte de l'utilisation d'un taux pour chaque catégorie de personnel (*technicien, qualitatif, encadrement, documentation, support, etc.*). Vous devrez déterminer quel pourcentage de la charge totale du projet doit être affecté à chaque profil. À nouveau, les données historiques ou des modèles classiques peuvent apporter une aide.

²⁰ MCCONNELL 1996



Le processus d'estimation d'un projet.

E. ESTIMER A PARTIR DES DELAIS IMPOSES

Souvent, la date de livraison de l'ouvrage n'est pas négociable « La nouvelle version doit tourner dans 6 mois ». « Le nouveau service téléphonique pour les clients démarre dans 12 mois et votre logiciel devra être prêt ». Si vous connaissez déjà votre date de mise en œuvre, la seule chose que vous pouvez négocier est le champ des fonctionnalités à mettre en œuvre dans le temps imparti. Lorsque le temps imparti ne permet pas de tout faire, les fonctionnalités doivent être classées par priorité décroissante et regroupées en ensembles homogènes qui peuvent être développés en temps voulu.

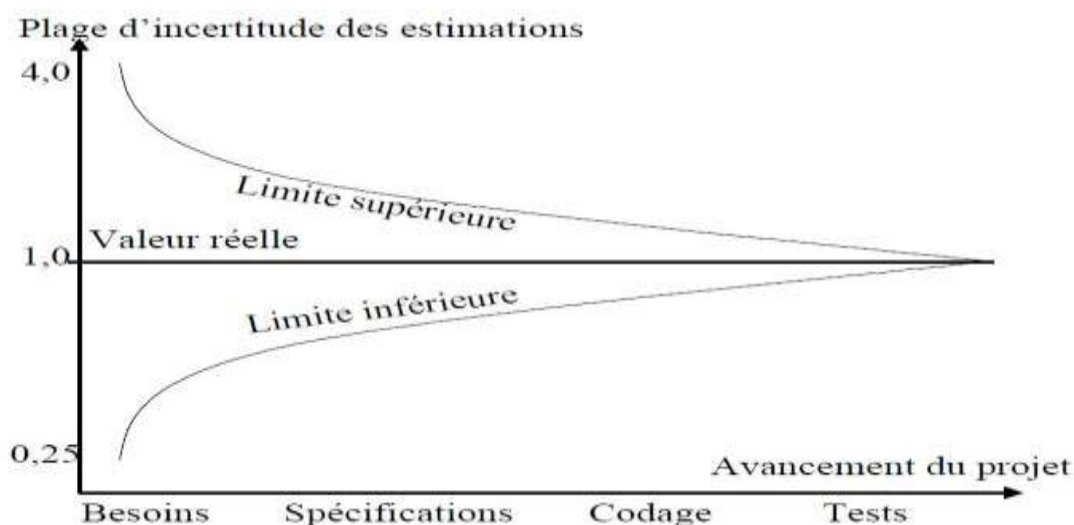
L'estimation à partir des délais impartis n'entraîne pas l'abandon des étapes du processus d'estimation indiqué ci-avant. Vous devrez toujours définir la taille de l'ouvrage, vous devrez l'éclater en diverses parties que vous pourrez soit sélectionner, soit soustraire de la livraison ; vous devrez toujours estimer les charges les délais, et les coûts. C'est là que les outils peuvent être réellement utiles. Essayer de réaliser un ensemble de fonctionnalités dans un temps limité exige de dérouler de nombreuses simulations. Des simulations manuelles prendraient trop de temps et consommeraient trop de charges ; des outils appropriés permettent de dérouler ces simulations facilement et rapidement.

IV.2.2.2. EXACTITUDE ET PRECISION D'UNE ESTIMATION

Nous aimerions connaître l'approximation d'une estimation prévisionnelle. Certes, vous ne connaîtrez cet écart à la réalité qu'à la fin du projet et vous devrez vivre avec cette incertitude. Naturellement, vous voulez que chaque estimation soit aussi exacte que possible en fonction des données dont vous disposez alors. Et, naturellement, vous ne voulez pas présenter une estimation d'une façon trop rigide qui inspirerait une fausse sensation de trop grande crédibilité de ces valeurs.

Qu'appelons-nous une estimation exacte ? ... *L'exactitude* caractérise l'approche de la réalité, alors que la *précision* caractérise la finesse avec laquelle une grandeur est mesurée. Par exemple, une estimation de taille de 70 ou 80 kilo-instructions peut être à la fois la plus *exacte* et la plus *précise* que vous puissiez faire à la fin de la phase de spécifications des besoins d'un projet. Si vous simplifiez votre estimation en annonçant 75 000 instructions, celle-ci *semble plus précise* mais en réalité elle est *moins exacte*. Vous pouvez annoncer 75 281 avec *une précision* d'une instruction, mais on ne pourra mesurer cette taille qu'à la fin de la phase de codage du projet, après décompte des instructions.

Si vous donnez l'estimation de la taille sous forme d'une plage (intervalle entre une valeur minimale et une valeur maximale) plutôt qu'avec une simple valeur, toutes les valeurs ultérieurement calculées (charges, délais, coûts) seront également représentées par des plages. Si, lors du déroulement du projet, vous faites plusieurs estimations au fur et à mesure que les spécifications de l'ouvrage deviennent plus détaillées, l'intervalle peut se resserrer et votre estimation se rapprochera du coût réel de l'ouvrage que vous développez.



Graphique de convergence des estimations « Développement rapide » (MCCONNELL 1996)
adapté de Modèles de coûts pour le cycle de vie (BOEHM 1995).

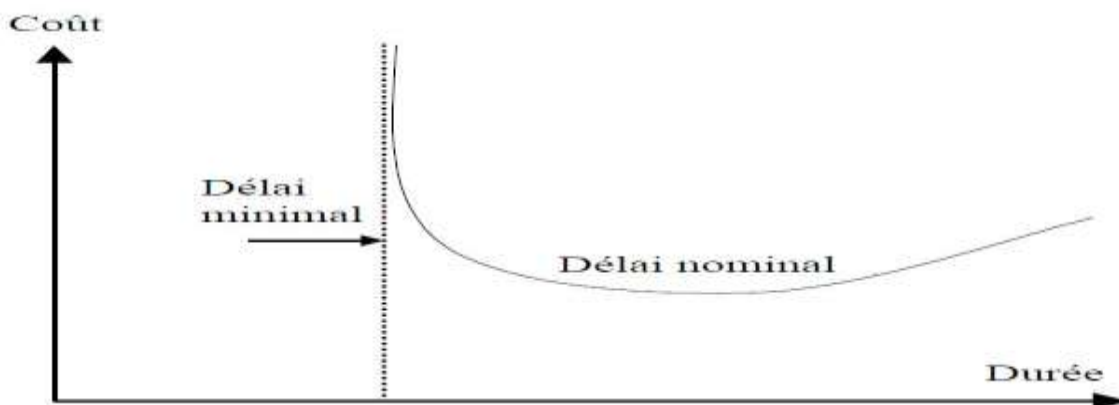
Bien sûr, vous devez aussi garder à l'esprit d'autres facteurs importants qui affectent l'exactitude de vos estimations :

- l'exactitude de toutes vos données d'entrées des estimations (le vieil adage « flou en entrée, flou en sortie » reste vrai) ;
- l'exactitude de tous les calculs (par exemple, la conversion des points de fonction ou des nombres d'instructions en charges, conserve une certaine marge d'erreur) ;
- la façon dont vos données historiques ou les données classiques utilisées pour calibrer le modèle s'appliquent au projet en cours d'estimation ;
- le respect du processus de développement préconisé par votre Organisme ;
- les conditions de management du futur projet : rigueur de la planification, conduite et contrôle ;
- l'absence d'incident majeur déclenchant des retards intempestifs.

IV.2.2.3. COMPRENDRE LES ARBITRAGES

Lorsque vous avez élaboré une estimation du projet, le travail réel commence. Il faut trouver un compromis de fonctionnalités, de délais, de coûts et d'effectifs, qui puisse être accepté à la fois par le management et par les clients ! C'est alors qu'une bonne compréhension des relations entre ces différentes variables s'avère particulièrement importante, et où la disponibilité des résultats des différents arbitrages rendus sur des projets précédents vous est très utile pour établir vos propres limites. Voici quelques évidences à rappeler pendant les phases d'arbitrages des estimations :

- Si vous allongez les délais, vous pouvez généralement réduire le coût global en utilisant moins de personnes. Quelquefois, il suffit d'augmenter le délai de quelques semaines pour obtenir un bénéfice. En règle générale, le management et les clients souhaitent un délai court, ce qui ne les empêche pas d'étudier attentivement les conséquences d'un allongement si ces conséquences sont acceptables pour eux. En effet, de nombreuses personnes n'envisageront une hypothèse de planification qui augmente le délai, que si elles sont fortement motivées par une réduction concomitante du coût du projet et de la taille de l'équipe.
- Il n'y a que trois décisions possibles pour réduire le délai :
 - diminuer les fonctionnalités (réduire la charge pour en faire moins) ;
 - augmenter les effectifs, lorsqu'il existe des tâches que l'on peut paralléliser ;
 - faire travailler l'effectif constant en dépassement d'horaires.



Relation entre coût et planning sur un projet logiciel Source Développement rapide (MCCONNELL 1996). Le coût de réalisation selon un planning normal est très inférieur au coût de réalisation la plus rapide possible.

- Si vous ne pouvez réduire les fonctionnalités de l'ouvrage, choisir l'une des autres possibilités peut s'avérer extrêmement coûteux. Il pourra vous en coûter beaucoup plus que votre budget prévisionnel selon la façon dont vous voulez réduire le délai. Et de plus, vous augmentez vos risques d'échec du projet ! Rappelez-vous la règle incontournable « Augmenter l'effectif d'un projet en retard, ne peut qu'aggraver le retard ». Ce même principe s'applique aux projets informatiques ; vous pouvez ajouter plus de ressources, mais la quantité de travail augmentera, car vous devrez gérer des communications supplémentaires et renforcer l'encadrement. Si vous escomptez une réduction du délai par le recours aux heures supplémentaires, vous devez vous souvenir que la productivité pourra, certes, augmenter à court terme mais elle risque de décroître à long terme, car les développeurs se fatigueront et commettront plus d'erreurs.
- Pour tout projet, il existe un délai minimal possible, que vous devez connaître. Vous pouvez seulement approcher ce délai pour des fonctionnalités bien définies, réalisées selon un processus minimal de développement et de test afin d'obtenir le niveau minimal de qualité souhaité. N'espérez pas franchir cette barrière. Vous n'êtes pas toujours en mesure d'atteindre ce délai minimal. Pour tenir le délai minimal, votre équipe de projet doit être particulièrement compétente et expérimentée, votre processus de développement doit être bien défini et stable et le projet doit se dérouler parfaitement. Il y a peu d'Organismes qui puissent espérer tenir le délai minimal et il est plus sage de ne pas le viser. Vous devez déterminer votre plus court délai possible (*ce que l'on appelle le délai nominal*). Les données historiques de vos projets passés demeurent votre meilleure source d'information.
- Gardez toujours à l'esprit l'exactitude de l'estimation que vous essayez d'ajuster. Si votre délai estimé est de 5 à 7 mois, alors un petit décalage de 2 semaines n'apparaîtra pas. Vous pouvez seulement ajuster le délai en ajouts significatifs par rapport à l'exactitude de l'estimation.

Il est intéressant d'observer les réactions de ceux qui apprennent à estimer des projets quand on leur demande de faire plusieurs estimations différentes en utilisant des options variées. Quand ils analysent les résultats, ils sont troublés par les conséquences des différentes options. Par exemple, le tableau suivant compare 3 estimations pour un projet de 75 kilo-instructions.

	Plan nominal Plan	délai minimal Plan	coût minimal
Charge (mois hommes)	40	97	14
Durée en mois	12,4	10	16,2
Coût (15\$ par mois)	600\$	1460\$	210\$
Effectif maximal	4,8	14,6	1,3
Effectif moyen	3,2	9,8	0,9

Les différentes estimations pour un projet de 75 000 instructions.

La différence entre le délai du plan nominal et celui du plan minimal est de l'ordre de deux mois mais pour viser le délai minimal, l'effectif maximal monte à plus de 10 personnes et le coût augmente de 860\$ (1 460 – 600). Ces résultats amènent à s'interroger si une diminution de 2 mois du délai vaut une telle augmentation de coût et si dix personnes supplémentaires peuvent être mobilisées en temps voulu pour accomplir le projet. Pour quelques rares projets, une réduction du délai peut être exigée, coûte que coûte, pour les autres, ce jeu n'en vaut pas la chandelle. Tous les projets ne présentent pas de telles différences entre les options, mais la relation entre taille, charge, délai, effectif, coût, suit quelques règles simples que vous ne pouvez transgresser. Disposer de plusieurs options, lorsque vous discutez l'estimation d'un projet, donne à chaque responsable, une vision des conséquences de ces règles simples et lui permet de prendre ses décisions en toute connaissance de cause.

La différence entre le délai du plan nominal et celui du plan minimal est de l'ordre de deux mois mais pour viser le délai minimal, l'effectif maximal monte à plus de 10 personnes et le coût augmente de 860\$ (1 460 – 600). Ces résultats amènent à s'interroger si une diminution de 2 mois du délai vaut une telle augmentation de coût et si dix personnes supplémentaires peuvent être mobilisées en temps voulu pour accomplir le projet. Pour quelques rares projets, une réduction du délai peut être exigée, coûte que coûte, pour les autres, ce jeu n'en vaut pas la chandelle. Tous les projets ne présentent pas de telles différences entre les options, mais la relation entre taille, charge, délai, effectif, coût, suit quelques règles simples que vous ne pouvez transgresser. Disposer de plusieurs options, lorsque vous discutez l'estimation d'un projet, donne à chaque responsable, une vision des conséquences de ces règles simples et lui permet de prendre ses décisions en toute connaissance de cause.

IV.2.2.4. LES DIFFICULTES DES ESTIMATIONS

L'estimation des charges des projets informatiques est absolument nécessaire ; mais c'est aussi l'une des activités les plus difficiles du développement de logiciels. Pourquoi est-ce si ardu ? ... La liste suivante indique quelques-unes de ces difficultés que nous devons surmonter :

- L'estimation de la taille est, intellectuellement, l'étape la plus difficile (mais non impossible) ; elle est souvent esquivée en passant directement à l'estimation des délais. Cependant, si vous ne vous interrogez pas sur l'objectif que l'on vous demande d'atteindre, vous n'aurez aucune base suffisante pour prévoir un délai ou évaluer les conséquences d'un changement de périmètre.
- Souvent, les clients et les techniciens de logiciels, ne comprennent pas que le développement de logiciels est un processus de raffinement progressif et que les estimations faites en amont du projet sont floues. Les bonnes estimations, elles-mêmes, ne sont que des paris, avec des hypothèses sur les risques inhérents ; cependant, on a quelquefois tendance à les considérer comme gravées dans le marbre ! Il est pertinent de présenter les estimations comme une plage de valeurs possibles, en exprimant, par exemple, que le projet prendra de 5 à 7 mois, au lieu d'affirmer qu'il sera achevé le 15 juin. Méfiez-vous d'une plage trop étroite qui reviendrait à donner une date précise ! Vous pouvez inclure l'incertitude sous forme d'une probabilité en disant, par exemple, qu'il est probable, à 80 %, que le projet soit achevé avant le 31 janvier 2019.
- Souvent, l'Organisme ne recueille ni n'analyse les mesures des performances des projets terminés. L'utilisation de données historiques est cependant la meilleure manière pour élaborer des estimations d'un nouveau travail. Il est très important d'établir une liste de caractéristiques fondamentales que vous mesurerez dans chaque projet.
- Il est souvent difficile d'obtenir un planning **réaliste** accepté par l'encadrement et les clients. Chacun préfère que les résultats soient disponibles au plus tôt, mais pour chaque projet, il y a un délai minimal qui vous permet d'intégrer toutes les fonctionnalités avec la qualité requise. Vous devez définir ce que vous pouvez faire dans un délai donné et expliquer à toutes les parties concernées ce qui est possible et ce qui ne l'est pas. Oui, il arrive, de temps en temps, que l'impossible se réalise ... mais c'est très rare et très coûteux. Espérer l'impossible relève d'une téméraire imprudence.

IV.2.2.5. PROJETS DE MAINTENANCE ET D'EVOLUTION, COMPARES AU NOUVEAU DEVELOPPEMENT

L'industrie du logiciel procède plus souvent à des travaux de maintenance et d'évolution sur des produits existants qu'à des développements complètement nouveaux. De nombreux projets de maintenance sont des combinaisons de nouveaux développements et d'adaptations de logiciels existants. Bien que toutes les étapes du processus d'estimation énoncé ci-dessus, s'appliquent aux projets de maintenance et d'évolution, il faut prendre en compte quelques aspects particuliers :

- Lorsque vous définissez la taille d'un nouveau développement intégré à un projet de maintenance, vous devez être conscient qu'insérer cette nouvelle fonctionnalité ne sera faisable que si l'architecture existante du produit peut l'intégrer. Si elle ne le peut pas, la charge de maintenance doit être augmentée pour remodeler cette architecture.
- Il est vain d'essayer de calculer la charge de maintenance évolutive comme celle d'un nouveau travail. : en définissant la taille du travail d'adaptation en nombre d'instructions ou en points de fonction et en la convertissant en charge (bien que cette approche ait été discutée. Il vaut mieux procéder à une estimation de la charge de maintenance par analogie avec les charges d'autres opérations similaires.
- Les modèles d'estimation, calibrés pour des estimations de charge et de délais de projets de nouveaux développements, supposent que chaque partie de l'ouvrage soit créée de toutes pièces. Ce n'est pas le cas pour les projets de maintenance dans lesquels vous modifiez une certaine partie de la documentation existante, le code, les cas de tests. Utiliser ces modèles peut entraîner une surévaluation des projets de maintenance.
- Souvent, le travail de maintenance est soumis à des délais fixes (par exemple, une version maintenue tous les 6 mois ou une fois par an) ou est réalisé par un effectif fixe (par exemple, une équipe de maintenance). Dans ce cas, les estimations doivent jouer avec un délai imposé et un effectif constant. Quelques modèles d'estimation prétendent s'adresser aux aspects de la maintenance. Mais actuellement, il y a beaucoup plus de support, de conseil, et de discussion, disponibles pour de nouveaux développements que pour les projets de maintenance et d'évolution. Heureusement, il y aura une évolution, car il existe une forte demande d'aides dans le domaine de la maintenance.

IV.2.2.6. ESTIMATION DES PETITS PROJETS

De nombreux développeurs travaillent sur des petits projets, généralement réalisés par une ou deux personnes en moins de 6 mois. Les modèles d'estimation diffusés (qui ne sont pas calibrés pour des petits projets) ne sont pas d'une grande utilité sauf lorsqu'ils peuvent être ajustés par les données historiques de petits projets déjà réalisés par l'Organisme.

Les estimations des petits projets sont largement dépendantes des performances des individualités impliquées et ainsi les meilleures estimations sont faites par ces futurs réalisateurs. Une approche telle que celle du PSP (*Personal Software Processus - Processus personnel pour le logiciel*) de Watts HUMPHREY²¹ est beaucoup plus réaliste pour les petits projets.

IV.2.2.7. ESTIMATION DES PROJETS DANS DE NOUVEAUX DOMAINES

Comment estimez-vous les charges d'un projet, dans un nouveau domaine applicatif, dans lequel aucun membre de votre Organisme n'a d'expérience ? Pour un projet novateur très pointu, personne (y compris en dehors de votre Organisme) ne saurait avoir d'expérience. La première fois que vous accomplissez un nouveau travail, vous affrontez de nombreuses incertitudes et il n'y a aucune solution autre que de conduire le projet avec précaution et de gérer le projet prudemment. Ces projets, à haut risque, sont généralement sous-estimés par les processus utilisés pour les estimations²². Connaissant ces deux aspects, vous devez :

- associer l'encadrement et les clients à la maîtrise des risques ;
- éviter les engagements majeurs sur les délais ;
- réestimer lorsque vous serez plus familier du domaine et dès que vous aurez spécifié le produit d'une façon plus détaillée.

Choisir un cycle de vie de projet pour mieux maîtriser les incertitudes des projets novateurs est souvent une étape clé manquante dans le processus de développement. Des cycles de vie, itératifs :

- modèle incrémental de révision (*IRM – Incremental Release Model*) : livraison par parties ;
- modèle en Spirale (Boehm) – révision des estimations et évaluation de risques avant de procéder à chaque nouvelle étape ; sont souvent de meilleures approches que le modèle traditionnel en Cascade.

²¹ HUMPHREY 1995

²² VIGDER 1994

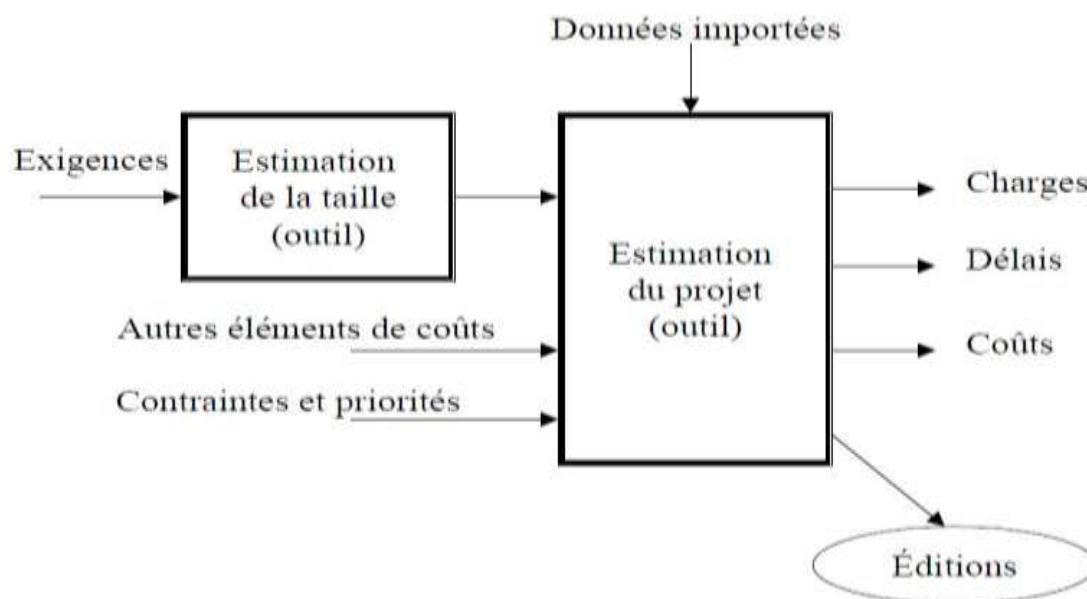
IV.2.2.8. CONSEILS PRATIQUES POUR L'ESTIMATION DES PROJETS INFORMATIQUES

1. Donnez-vous le temps suffisant pour faire une bonne estimation de votre projet. Les estimations bâclées sont imprécises et périlleuses. Pour des projets de grands développements, l'étape d'estimation doit être considérée comme un véritable mini-projet.
2. Si possible, utilisez des données enregistrées de votre propre Organisme sur des projets similaires. Il en découlera une estimation plus précise. Si votre Organisme n'a pas encore constitué cet historique, il est grand temps de commencer à recueillir les données.
3. Utilisez des estimations fondées sur les expériences des développeurs. Les estimations, préparées par des personnes différentes de celles qui feront le travail, sont moins précises.
4. Utilisez au moins un outil d'estimation du logiciel. Les outils d'aide à l'estimation mettent en oeuvre des modèles complexes qui demanderaient trop de temps d'apprentissage significatif pour les appliquer manuellement. Ces outils vous assurent que vous n'oublierez rien et vous permettent d'ajuster une estimation rapidement et facilement.
5. Utilisez plusieurs estimateurs et utilisez plusieurs techniques différentes (utiliser un outil peut être considéré comme l'une de ces techniques) et comparer les résultats. Observez la convergence ou la dispersion des estimations. La convergence confirme que vous avez vraisemblablement fait une bonne estimation. La dispersion signifie qu'il y a probablement des aspects qui ont été négligés pour certaines estimations ; vous devez approfondir ces aspects.
6. Réestimer le projet, plusieurs fois, au cours de son cycle de vie. Au fur et à mesure que vous spécifiez les détails du produit, vos estimations se rapprocheront des consommations réelles nécessaires pour achever le projet.
7. Créez une procédure standardisée d'estimation que toute personne impliquée adoptera. On pourra discuter les entrées de cette démarche mais pas remettre en cause les sorties. Votre effort se répartira d'une façon progressive, tout en découvrant le périmètre et les facteurs de coût du projet.
8. Centrez vos efforts sur l'amélioration de votre processus d'estimation des projets informatiques de votre Organisme. Comparez les charges consommées de chaque projet terminé aux charges prévisionnelles. Comment aviez-vous prévu les charges et les délais ? Qu'aviez-vous oublié ? Qu'auriez-vous pu améliorer ?

IV.2.2.9. LES OUTILS D'AIDE A L'ESTIMATION DES PROJETS INFORMATIQUES

Les outils d'estimation peuvent être des produits indépendants ou être intégrés à des fonctionnalités de produit d'aide à la gestion de grands projets. Ces outils peuvent simplement aider au processus d'estimation de la taille ou se contenter de la conversion de la taille en charge, délais et coûts ou les deux à la fois. Les outils qui aident au calcul de la taille incluent la détermination du nombre d'instructions, l'analyse des points de fonction et, parfois, la saisie des exigences et la connaissance des applications de gestion.

Il sied de rappeler qu'aucun outil d'estimation n'est une « *baguette magique* » pour résoudre vos problèmes d'estimation. Ce sont des aides précieuses de votre panoplie d'estimation et vous devez sérieusement en utiliser un ou plusieurs outils, mais n'oubliez pas que la qualité des sorties dépend de la qualité de vos données d'entrée et que vous devez définir un processus de développement et d'estimation. Attention, méfiez-vous de ces fournisseurs qui prétendent que leur outil est capable d'élaborer des estimations d'une excellente approximation, sauf s'ils indiquent toutes les actions préparatoires et toutes les actions à accomplir pendant le projet, pour vous assurer en permanence de l'exactitude de l'estimation²³.



Le contexte des outils d'estimation

²³ Il existe un large éventail d'outils disponibles. Chercher un outil d'estimation sur la Toile (web) n'est pas aussi immédiat que l'on pourrait l'espérer. Il faut utiliser des combinaisons de mots-clés avec un moteur de recherche pour découvrir 80 % des outils et des sites qui contiennent la liste des autres. Les informations recueillies sur la Toile sur les aptitudes et les prix des outils sont très variables et parfois superficielles, aussi quelques adresses électroniques et quelques numéros de téléphone doivent être utilisés pour approfondir les premières moissons d'information.

Les caractéristiques importantes et les critères que vous devez respecter quand vous évaluez un outil d'aide à l'estimation de projets informatiques :

1. **Prix** : Les outils d'estimation se classent selon leur mode de rémunération, c'est-à-dire, **en location** (*contre une redevance annuelle*) ; **à l'achat** (*un seul paiement*). Seuls, les grands Organismes et les grands projets envisageront les produits de prix élevé. Les outils à moins de 1 000 € mettent en œuvre des modèles communs publiés par d'autres (par exemple COCOMO) et certaines fonctionnalités risquent de faire défaut comme le support étendu des options perfectionnées, mais ces outils peuvent cependant produire plus que les seules estimations.
2. **Plates-formes et performances** : L'outil tourne-t-il sur votre plate-forme (*matériels et logiciels*) ? Tourne-t-il sur d'autres plates-formes ? Quelles sont les capacités de mémoire (*vive et espace disques*) exigées ? Sa base de données intégrera-t-elle la quantité de données historiques présentes et à venir et gèrera-t-elle le nombre de projets que vous voulez estimer ?
3. **Facilité d'utilisation et documentation** : Pouvez-vous facilement élaborer la première estimation d'un projet, lors de la prise en main de l'outil ou devez-vous étudier le modèle sous-jacent en détail pendant plusieurs jours, déchiffrer les abréviations et assimiler les définitions des attributs ? Pouvez-vous esquisser votre estimation facilement ? Le mode d'emploi et les messages d'aide en ligne vous permettent-ils de comprendre comment utiliser l'outil pour élaborer des estimations de projets au-delà d'une liste des fonctionnalités offertes par l'outil ? Existe-t-il un exemple type ?
4. **Aptitude à travailler en réseau** : Existe-t-il une base de données commune partageable à laquelle plusieurs utilisateurs peuvent accéder et peuvent-ils enrichir l'historique à partir de leurs propres données, visualiser et mettre à jour les estimations (en supposant que cette aptitude soit importante pour vous) ?
5. **Évolutions** : Le modèle ne doit pas être figé. Au fur et à mesure que de nouveaux langages de programmation et de nouveaux paradigmes de développement apparaissent, que la gamme des projets de développement s'étend, il faut envisager la mise à jour du modèle interne de l'outil. Le fournisseur vous donne-t-il accès au modèle ? Le fournisseur s'engage-t-il à faire des améliorations progressives qui vous offriront de nouvelles fonctionnalités opérationnelles sur de nouvelles plates-formes ? Combien coûte cette mise à jour ?

6. **Assistance** : Il faut bien comprendre qu'en dépit des progrès des outils d'estimation (plus accessibles, plus ergonomiques) les modèles sous-jacents sont très complexes et vous pouvez être amené à poser quelques questions ou quelquefois avoir besoin d'un conseil. Le fournisseur propose-t-il un support technique et des moyens pour répondre aux questions « comment faire » ? Le fournisseur offre-t-il des cours d'estimation, au-delà du simple mode d'emploi de l'outil ou recommande-t-il des cours de soutien et de perfectionnement ? Le fournisseur vous offre-t-il des manuels d'auto formation ?

7. **Spécification du périmètre et de la taille** : La flexibilité est un facteur-clé de l'outil. Vous pouvez commencer, d'une certaine façon, l'estimation de la taille d'un projet, puis au fur et à mesure que vous découvrirez les spécifications de votre ouvrage particulier ou lorsque vous deviendrez plus compétent en estimation, vous souhaitez vous brancher sur d'autres techniques et vous voudrez que l'outil supporte ces nouveaux besoins. Quelles options l'outil vous propose-t-il pour spécifier à nouveau une estimation ? Pouvez-vous entrer soit des nombres d'instructions, soit des points de fonction ? Pouvez-vous spécifier des composants GUI, des nombres de classes ou de méthodes ou des modules et des fonctions ? Est-ce que la taille s'introduit comme une seule valeur (par exemple, 55 000 instructions, 345 points de fonction,) une plage de nombres (minimum 45 000 – probable 55 000 – maximal 65 000) ? La taille peut-elle être estimée en la divisant en modules ou en lots de travaux dont vous estimerez chaque composant particulier de façon convenable ?

8. **Modèle(s) d'estimation** : Quelques outils utilisent un ou plusieurs modèles propriétaires pour lesquels une information succincte est diffusée. D'autres utilisent des modèles non-propriétaires pour lesquels vous pouvez acheter un livre ou télécharger une information détaillée à partir de la Toile mondiale pour en savoir plus. L'élaboration d'un modèle perfectionné de développement de logiciel consomme beaucoup de ressources ; il n'est donc pas surprenant qu'il n'y ait qu'une poignée de modèles. En étudiant les algorithmes internes de l'outil, vous devez vérifier que les estimations générées par l'outil sont utiles en regard des types de projets développés par votre Organisme. Les modèles paramétriques reposent sur des spécificités, qui ont tendance à biaiser les résultats ; par exemple, quelques-uns suivent un processus de développement d'applications militaires alors que d'autres suivent un processus de développement d'applications commerciales. La seule façon de savoir rapidement si un outil peut vous apporter des résultats valables est d'obtenir une version d'évaluation ou de démonstration et de procéder à l'estimation de projets terminés pour lesquels vous connaissez les consommations réelles. Il convient de comparer les estimations fournies par l'outil avec les

données des projets terminés et d'analyser l'amplitude des écarts. Vous devez vérifier si l'outil vous permet de saisir les données historiques des projets terminés et vous devez contrôler l'ergonomie de la saisie. Quelques outils ne pourront être calibrés pour vos projets qu'en modifiant le modèle sous-jacent (c'est-à-dire que vous devez calculer les valeurs) d'autres vous permettent d'introduire simplement les métriques du projet comme la taille réelle, la charge, le délai, et alors l'outil génère les changements du modèle. L'outil supporte-t-il l'estimation de projets de maintenance et d'évolution ? L'outil supporte-t-il l'orienté objet, la réutilisation du logiciel ou d'autres particularités importantes pour votre projet ?

9. ***Autres facteurs de coûts*** : Généralement, les modèles vous permettent d'évaluer certains coûts et certains facteurs de productivité (ce sont les aptitudes et l'expérience de l'équipe, les exigences du cycle de vie, l'utilisation d'outils) de façon à adapter l'estimation à la situation particulière de votre projet, dans votre Organisme. Quels facteurs de coût sont-ils disponibles et les valeurs correspondantes sont-elles utiles dans votre contexte ?
10. ***Contraintes et priorités*** : L'outil vous permet-il de spécifier les contraintes (par exemple, un délai maximal de 12 mois avec une équipe maximale de 10 personnes) pour calculer votre estimation ? L'outil vous permet-il de spécifier les priorités (par exemple, si le délai minimal présente la plus grande priorité ou si, au contraire, le plus faible effectif a la plus grande priorité) dans le calcul des estimations ?
11. ***Génération de sorties*** : Il faut rechercher un outil dont les fonctions offrent des options, des probabilités et des plages. Les outils utilisant la simulation de Monte-Carlo pour produire des estimations, avec des probabilités différentes, fournissent d'intéressantes perspectives, liées aux inconstances des processus de développement. Les états restitués peuvent vous aider à présenter clairement les estimations et à les discuter avec les clients et l'encadrement. Quelles sortes d'états sont élaborées ? Vous sont-ils utiles ? Pouvez-vous obtenir des copies électroniques des états afin de les compléter et de les insérer facilement dans les autres documents du projet ?
12. ***Aptitude d'importation/exportation*** : Les importations possibles comprennent des éléments tels qu'estimation des tailles, module par module, des données historiques sur les projets, et des mises à jour du modèle. Les exportations possibles comprennent le planning, le WBS (*Work Breakdown Structure* = *Organigramme des Tâches*), et des informations pour les outils de gestion de projet tels que MS-Project ou des feuilles Excel.

CONCLUSION

En guise de conclusion, le pilotage et/ou la gestion de projets informatiques est un ensemble de techniques qui permettent d'identifier, de planifier et de piloter le développement d'un produit logiciel. Toutefois l'évolution actuelle a fait susciter l'aspect managériale afin d'avoir une plus grande valeur ajoutée qui permet la conduite du projet vers la réussite. Ces techniques et outils ne peuvent fonctionner pleinement que dans le cadre d'une gestion dimensionnée par projet.

En effet, nous avons voulu montrer, dans les différents chapitres de ce cours, qu'il existe des techniques permettant de maîtriser le management hors hiérarchie qu'implique une organisation par projet répondant aux nécessités du troisième millénaire. Le tout étant d'accepter d'y consacrer les moyens voulus en fonction de l'ambition du projet. Parce qu'il ne reproduit pas de modèle mais crée des modèles nouveaux, la conduite par projet est l'outil du passage à une nouvelle forme de progrès: développer à la fois le potentiel des hommes et celui des métiers de l'entreprise.

Plus qu'une méthode d'organisation, la conduite de projets informatiques représente une nouvelle approche culturelle pour les entreprises informatiques ; une approche fondée sur la culture de la transversalité. En ce sens, la conduite de projets va donc bien plus loin que le fait de faire travailler ensemble des gens venant de différents métiers sur un objectif commun. C'est à la fois un outil stratégique de l'entreprise pour répondre aux changements rapides des marchés et un formidable outil de motivation pour ceux qui y participent comme pour les métiers, si tous les apprentissages réalisés sont mémorisés. Le tout est d'avoir envie et de savoir-faire vivre cette transversalité entre les objectifs à court terme des projets et les objectifs à long terme de l'entreprise. Faire vivre un projet est une question de méthodes et d'hommes. C'est aussi, pour les entreprises, une question de volonté.

L'auteur de cet ouvrage, le Docteur YENDE RAPHAEL Grevisse, vous remercie infiniment du temps que vous avez et que vous consacrerez à l'exploitation de ce dit ouvrage usité aux différentes méthodes de conduite de projets informatiques. Bonne lecture et digestion.

YENDE RAPHAEL Grevisse, PhD.
Professeur associé