

Ecole Nationale Polytechnique d'Oran – Maurice Audin
Département de Génie des Systèmes Informatique



POLYCOPIE DE COURS (Cours & TDs)

4^{ème} année Spécialité :
Ingénierie et Management des Systèmes d'Information
&
Réseaux et Télécommunication

Module : _____
Management de projets IT

Réalisé par : Dr. Menaouer BRAHAMI

Préface

Le monde du travail s'est vigoureusement réformé au cours de ces dernières décennies. En effet, l'acquisition d'expériences en matière management des projets a amené les organisations à développer une approche qui est actuellement largement appliquée. Celle-ci répartit le management des projets en un nombre de phases distinctes. L'ensemble de ces phases constitue ce qu'on appelle Cycle de Projet et la gestion harmonieuse et intégrée de ces différentes phases constitue le Management du Cycle de projet ou MCP. A cet effet, le management est le fruit de la succession d'évènements marqués par l'accumulation des crises et échecs qui vont donner des nouvelles réflexions sur l'organisation du travail des entreprises. De même, le management de projet vise un ensemble de méthodes et d'outils servant à permettre le développement et la livraison d'un produit ou d'un service à un niveau de qualité défini, dans les délais et au coût prévus. Pour cela, la mise en place d'un projet est un des enjeux fondamentaux pour les entreprises, les organisations et les institutions d'optimiser l'utilisation de leurs ressources humaines et matérielles. De plus, un projet IT nécessite des connaissances et une réelle expertise technique de l'équipe projet. Le responsable SI, au cœur du développement et de l'évolution des systèmes, tient souvent le rôle de sponsor du projet IT. Il veillera à l'adaptation de la solution déployée aux besoins des utilisateurs.

Nous proposons donc dans cet ouvrage de :

1. définir la notion de projet IT ;
2. établir un lien entre les éléments qui le caractérise et la manière de le conduire ;
3. étudier le processus mis en œuvre ;
4. analyser les rôles des différents acteurs d'un projet ;
5. déterminer les causes d'échec et les conditions de réussite ;
6. montrer l'intérêt de cette démarche.

Dans ma vie d'enseignant chercheur, ce que je vis au quotidien, ce sont de nombreux recadrages et beaucoup de temps à clarifier les attentes et à préciser le contenu non exprimé de phrases telles que : «vous voyez ce que je veux dire», «faites- moi cela»,...

Très pragmatique, cet ouvrage apporte un fil rouge méthodologique, des techniques, des conseils, des outils et montre quels écueils éviter, pour que cette étape incontournable ne se fasse plus dans les problèmes ! Bref, enfin, une méthode structurée accompagnée d'outils pratiques et simples à utiliser !

Mots-clés : *Gestion de projet, Management de projet, Conduite de projet, Conduite de changement, Étude de faisabilité, Conception de projet, Cycle de vie d'un projet, Méthode PERT, Diagramme de GANTT, Management des risques projet, Gestion des conflits, Outils de planification.*

Chapitre 3

Les parties prenantes d'un projet IT

1. Les acteurs ou les parties prenantes d'un projet

Les parties prenantes du projet sont les individus, les organisations et les entreprises activement impliquées dans le projet, ou dont les intérêts peuvent subir l'impact de l'exécution ou de l'achèvement du projet. Elles peuvent aussi influencer les objectifs et les résultats du projet. L'équipe de management de projet doit identifier ces parties prenantes, déterminer leurs exigences et leurs attentes et, dans la mesure du possible, gérer leur influence par rapport aux exigences de façon à assurer le succès du projet. Lorsque les parties prenantes participent à un projet, elles ont différents niveaux de responsabilité et d'autorité qui peuvent évoluer au cours du cycle de vie du projet. Leur responsabilité et leur autorité vont de contributions occasionnelles à des enquêtes et des groupes de réflexion, jusqu'à un parrainage complet du projet avec soutien financier et politique. De plus, les parties prenantes peuvent avoir une influence positive ou négative sur un projet. Les parties prenantes positives sont celles qui devraient normalement bénéficier de la réussite du projet, tandis que les parties prenantes négatives sont celles qui considèrent que le succès du projet aurait pour elles des conséquences négatives.

Par exemple, lorsqu'une communauté va bénéficier d'un projet d'expansion industrielle, ses leaders du monde des affaires peuvent être des parties prenantes positives car ils voient dans sa réussite un intérêt économique pour leur communauté. À l'inverse, des groupes environnementaux peuvent être des parties prenantes négatives s'ils considèrent que le projet nuira à l'environnement. Pour les premiers, aider le projet à aboutir sert leurs intérêts au mieux et ils pourront par exemple contribuer à l'obtention des permis nécessaires pour continuer le projet. Pour les intérêts des seconds, la logique serait d'entraver la marche du projet en exigeant des études plus poussées de l'impact sur l'environnement. Les parties prenantes négatives sont souvent négligées par l'équipe de projet, avec le risque de ne pas voir son projet couronné de succès.

1.1. Les acteurs participants au projet

Tous ceux qui sont susceptibles d'assurer une fonction indispensable au bon déroulement du projet :

1.1.1. Le maître d'ouvrage (MOA) [owner, client, contracting part]

L'AFNOR¹ définit la maîtrise d'ouvrage comme « *une personne physique ou morale qui sera propriétaire de l'ouvrage. Elle assure le paiement des dépenses liées à la réalisation* ». De plus, on trouve dans la littérature que la maîtrise d'ouvrage est toujours la responsable globale du projet car elle dispose de moyens de contrôle et de corrections auprès des parties prenantes du projet. Toutefois, la maîtrise d'ouvrage n'est pas responsable de la solution tant que celle-ci n'est pas livrée.

¹ Association Française de NORmalisation : Association chargée d'élaborer les référentiels de normalisation demandés par les acteurs économiques français pour faciliter leur développement stratégique et commercial.

Dans un projet informatique, *la maîtrise d'ouvrage informatique* est responsable de la bonne restitution des besoins et de leurs spécifications détaillées. Elle doit assimiler le métier des utilisateurs et être capable de le traduire dans un langage compréhensible à la maîtrise d'œuvre. La maîtrise d'ouvrage doit donc avoir une appétence particulière à l'informatique et même maîtriser certaines techniques telles que l'accès aux bases de données. Plus précisément, elle est le liant entre le fonctionnelle et le technique. Aussi, elle peut être issue des services fonctionnels ou bien du service informatique.

1.1.2. Le maître d'œuvre (MOE) [fournisseur, contractor, engineer]

L'AFNOR définit la maîtrise d'œuvre comme « *une personne physique ou morale qui réalise l'ouvrage pour le compte du maître d'ouvrage, assure la responsabilité globale de la qualité technique, du délai et du coût* ». Aussi, le maître d'œuvre peut avoir plusieurs composantes comme, pour une entreprise, le conseil d'administration, le directeur (souvent responsable du projet qui fera office d'arbitrage, ou encore l'équipe projet qui comprend le chef de projet (qui peut être différent du directeur) dont les experts associés. La maîtrise d'œuvre est par conséquent responsable de la solution jusqu'à la livraison officielle de celle-ci. Toutefois, elle reste responsable de la qualité de la solution développée, même après la livraison, au minimum pendant la période de garantie. Enfin, la maîtrise d'œuvre a également un devoir d'information auprès de la maîtrise d'ouvrage.

Dans un projet informatique, *la maîtrise d'œuvre informatique* est responsable de la conception, de la réalisation technique et de l'intégration du système. La maîtrise d'œuvre est aussi le liant entre le système réalisé et les services d'exploitation informatique qui auront en charge d'assurer la qualité de service du système.

N.B. : la maîtrise d'ouvrage (MOA) et la maîtrise d'œuvre (MOE) sont deux entités distinctes qui peuvent cependant appartenir à une même entreprise (ou une même organisation). Dans le cadre d'une maîtrise d'ouvrage PME, il est rare de pouvoir considérer une maîtrise d'œuvre appartenant à la même organisation que la maîtrise d'ouvrage.

1.1.3. L'Assistant à la maîtrise d'ouvrage (AMO)

La maîtrise d'ouvrage d'un projet peut faire appel à une entité (interne ou externe) pour l'assister dans le cadre du projet. Cette entité est donc un assistant à la maîtrise d'ouvrage. Il a pour unique responsabilité de conseiller la maîtrise d'ouvrage dans ses choix et éventuellement dans ses méthodes de travail. Toutefois, il arrive parfois que la maîtrise d'ouvrage délègue une partie de ses tâches à l'assistant à la maîtrise d'ouvrage. Dans ce cas l'assistant assume des tâches de réalisation, mais les aspects décisionnels et de validation restent toujours (et doivent dans tous les cas rester) du ressort de la maîtrise d'ouvrage. Il peut être force de proposition lorsque la maîtrise d'ouvrage manque d'expérience ou de visibilité sur tout ou partie de son projet. Toutefois l'assistant à la maîtrise d'ouvrage ne doit pas se substituer à la maîtrise d'ouvrage, ni la décharger de ses responsabilités.

1.1.4. La Maîtrise d'Ouvrage Déléguée (MOAD)

La Maîtrise d'Ouvrage Déléguée est mandatée lorsque :

- la MOA n'a pas vocation à réaliser les activités du projet, ni à les piloter de façon opérationnelle ;

- la MOA ne souhaite pas réaliser ou piloter tout ou partie des activités du projet.

La MOAD va alors réaliser le projet (ou la partie du projet qui lui est déléguée) en lieu et place de la MOA. Elle tiendra la MOA informée des avancements du projet, mais c'est la MOAD qui a la pleine responsabilité de l'atteinte des objectifs.

1.1.5. Le chef de projet (directeur du projet)

Il a la charge de la mise en œuvre du projet, à savoir l'atteinte des objectifs conformément au niveau de qualité, aux délais et aux coûts spécifiés et dans le respect des règles et procédures réglementaires applicables à l'entreprise. Il rend compte au Sponsor et au Comité de pilotage. D'une manière pratique, le chef de projet propose la composition de l'équipe projet, évalue les facteurs de risques et les gère à tout moment. Il affecte les travaux à réaliser aux différents membres de l'équipe projet, suit l'avancement des travaux, anime l'ensemble de l'équipe projet et s'assure du bon niveau de motivation des membres de l'équipe, valide les documents intermédiaires et finaux, arbitre les conflits entre les acteurs de l'équipe projet, suit les budgets et les délais, remonte au Comité de pilotage les décisions de son ressort et rend compte de l'avancement du projet au Comité de pilotage. Le chef de projet (responsable de projet) est un informaticien issu du service informatique. Il est en mesure de soutenir la MOE dans son quotidien.

- Le demandeur. Ce peut être l'entreprise.
- Les fournisseurs externes et éventuels sous-traitants
- Les utilisateurs finals (les bénéficiaires) : Ce peut être le maître d'ouvrage mais ce peut être pour favoriser des catégories de population. De même, les utilisateurs sont les destinataires finaux du projet. Il participe au projet sous la responsabilité de la maîtrise d'ouvrage. Le rôle des utilisateurs est important en particulier au niveau de :
 - L'expression des besoins.
 - Les tests de recette.
 - La mise en service du projet.

1.1.6. Le comité de pilotage

Il peut regrouper tout le monde cité précédemment ainsi que des personnes extérieures. Selon plusieurs auteurs, le comité de pilotage est l'organe de contrôle et de réalisation du projet global. Il se réunira chaque mois ou, si besoin, à la demande du comité de projet. Enfin, le comité de pilotage est composé d'acteurs désignés par le client, du directeur de projet et de tout acteur sollicité par l'une ou l'autre des parties.

1.1.7. Les autres parties prenantes susceptible d'exercer une influence sur le Projet

- les autres projets du maître d'ouvrage qui peuvent être concurrents ou complémentaires ;
- les autres membres de l'organisation initiatrice du projet ;
- Les chercheurs ;
- Les organisations syndicales, humanitaires ;
- Les médias ;
- Les administrations ;
- Les professionnels et spécialistes concernés ;

- Les membres des sociétés civiles ;
- Souvent il faut ménager tout cela et exercer du lobbying.

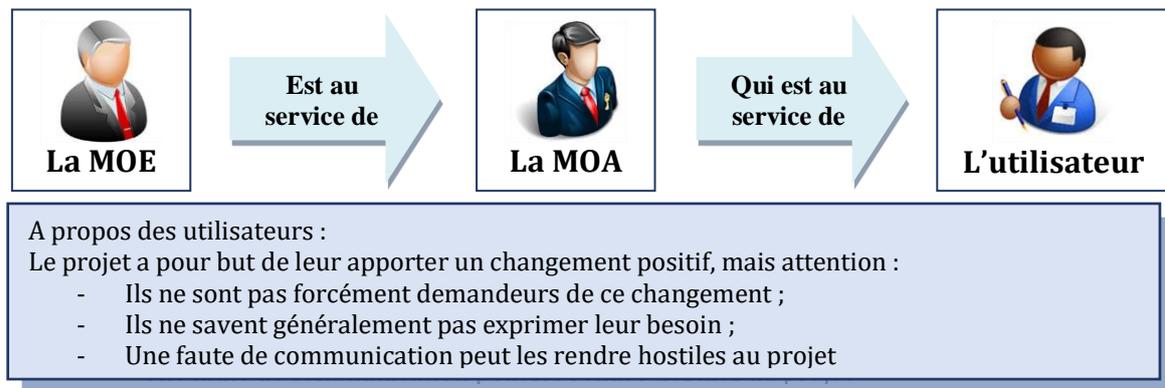


Figure. 5. Relation entre la MOE, la MOA et l'Utilisateur

Exemple :

Question : Quel est le rôle d'un chef de projet ?

Solution : Les activités de gestion d'un projet informatique sont très similaires à celles des autres domaines :

1. Écriture de proposition de projet
2. Planification du projet
3. Évaluation des coûts
4. Surveillance du projet et écriture de rapport d'étapes
5. Sélection du personnel et évaluation
6. Écriture de rapport et de présentation

1.2. Les rôles et les responsabilités des parties prenantes

Les rôles et les responsabilités des parties prenantes peuvent se chevaucher, comme lorsqu'une société d'ingénierie assure le financement de l'usine qu'elle est chargée de concevoir. Les chefs de projet doivent gérer les attentes des parties prenantes, ce qui peut s'avérer difficile car elles ont souvent des objectifs différents, voire contradictoires.

A titre d'exemple, nous citons les exemples suivants :

- Dans une société d'électronique, le vice-président pour la recherche définira le succès d'un nouveau produit en termes de « technologie à la pointe du progrès », le vice-président de la production y verra des « utilisations d'envergure internationale », alors que la préoccupation majeure du vice-président du marketing sera avant tout le nombre de nouvelles fonctionnalités.
- Un promoteur immobilier s'intéressera aux délais de réalisation, les autorités locales voudront maximiser les recettes fiscales, un groupe d'écologistes souhaitera minimiser les effets nuisibles à l'environnement et les habitants du voisinage espéreront que le projet se réalisera ailleurs.

2. Les contraintes d'un projet informatique

Coût, délais, qualité : ces trois mots résument les trois préoccupations du chef de projet. Lorsqu'un chef de projet accepte la responsabilité d'un nouveau projet, il l'accepte dans un cadre qui doit être bien défini et validé par toutes les parties concernées par le projet. A cet effet, le triangle de la triple contrainte, aussi appelé triangle de la performance, est souvent utilisé pour illustrer l'interdépendance des variables d'un projet. En effet, dans un projet, les modifications apportées à l'une des variables auront irrévocablement des répercussions sur les autres ou, en d'autres termes, privilégier une contrainte se fait généralement au détriment des autres.

Ainsi, pour un projet donné, si l'on décide de réduire le temps de développement, il faudra, pour maintenir le niveau de qualité convenu, augmenter le budget en y affectant par exemple davantage de ressources ou, sinon, accepter de diminuer les attentes au plan de la qualité. Ou encore, si l'on décide de réduire le budget du projet, il faudra alors, pour maintenir le niveau de qualité prévu, augmenter le temps de développement accordé ou, sinon, accepter là aussi d'en diminuer les attentes sur le plan de la qualité.

Enfin, si l'on décide de réduire les exigences de qualité du projet, il sera évidemment possible soit d'en réduire les coûts, soit d'en réduire le temps de développement ou encore de répartir l'économie à la fois sur les coûts et le temps de développement.

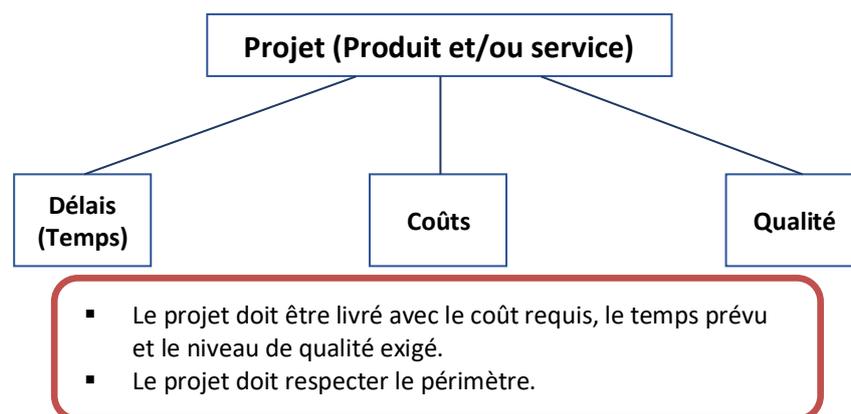


Figure. 6. Le triangle Qualité, Coût, Délai

2.1. Les coûts

Avant de lancer le projet, un chiffrage précis doit être réalisé. Ce chiffrage doit être le plus exhaustif possible. Il doit comprendre (par exemple) :

- le coût des jours consacrés à l'étude, aux réunions, à la rédaction des comptes rendus,
- les jours consacrés au développement informatique du projet,
- les jours consacrés aux tests, à la mise en production,
- etc...

Ce n'est qu'une fois ce chiffrage est réalisé et validé par tous les intervenants que le projet peut commencer.

La justesse du chiffrage est importante car la consommation du budget sert d'indicateur à l'avancement du projet. Cet indicateur est faussé d'emblée si le chiffrage a été volontairement ou involontairement surestimé ou sous-estimé.

La sous-estimation est par exemple pratiquée par une société de prestation informatique pour décrocher un appel d'offres. Généralement, la société se rattrape par des avenants au contrat sur des prétextes plus ou moins réels (seul un contrat bien ficelé, et un cahier des charges très précis permet de contrer ces tentatives).

2.2. Les délais

Un planning précis doit être établi entre la maîtrise d'œuvre (MOE) et la maîtrise d'ouvrage (MOA). Ce planning doit donner les dates jalon principales, c'est à dire celles qui correspondent à des étapes précises dans le projet.

La validation de ce planning est importante, parce qu'il sera utilisé par toutes les parties pour juger de l'avancement du projet. Ce planning doit être établi en tenant compte de tous les paramètres pouvant impacter le projet : congés, ressources disponibles, délais incompressibles de certaines actions, etc...

La maîtrise d'œuvre (MOE) risque de ne pas pouvoir respecter ses engagements en matière de délais si elle se voit contrainte d'adapter le planning non pas en fonction des véritables contraintes, mais en fonction de la demande initiale de la maîtrise d'ouvrage (méthode du rétro planning) qui souhaite un délai très court, au risque que ce délai soit impossible à tenir.

2.3. La qualité

Un développement informatique répond à des règles de l'art précis qui obligent la maîtrise d'œuvre (MOE) à livrer à sa maîtrise d'ouvrage (MOA) un outil informatique qui fonctionne sans erreurs, et surtout, qui respecte le cahier des charges fonctionnelles validées avec la maîtrise d'ouvrage (MOA).

La maîtrise d'œuvre (MOE) risque de ne pas pouvoir tenir ses contraintes de qualité si la maîtrise d'ouvrage (MOA) modifie en cours de projets ses spécifications fonctionnelles, en ajoutant ici et là des fonctionnalités impactantes, non prévues initialement.

3. Les différentes méthodes de projet informatique

Mais revenons, dans un premier temps, sur les méthodes qui existent pour conduire des projets informatiques en entreprise. On appelle cycle de vie d'un logiciel (*software life cycle*) l'ensemble des étapes à suivre lors de la création d'un logiciel ainsi que leur enchaînement. Aussi, le cycle de vie d'un logiciel (*software life cycle*) regroupe les étapes de production du logiciel, ainsi que leur ordonnancement. Il existe dans la littérature deux méthodes : méthodes classiques (en cascade, en V, etc..) et méthodologies Agiles.

3.1. Le modèle en cascade (de cycle de vie en cascade) d'un projet (logiciel)

Le modèle de cycle de vie en cascade a été mis au point dès 1966, puis formalisé aux alentours de 1970 par *Winston W. Royce*. Dans ce modèle le principe est très simple : chaque phase se termine à une date précise par la production de certains documents ou logiciels. Les résultats sont définis sur la base des interactions entre étapes, ils sont soumis à une revue approfondie et on ne passe à la phase suivante que s'ils sont jugés satisfaisants. Le modèle original ne comportait pas de possibilité de retour en arrière. Celle-ci a été rajoutée ultérieurement sur la

base qu'une étape ne remet en cause que l'étape précédente, ce qui est dans la pratique s'avère insuffisant. (Voir Figure 7)

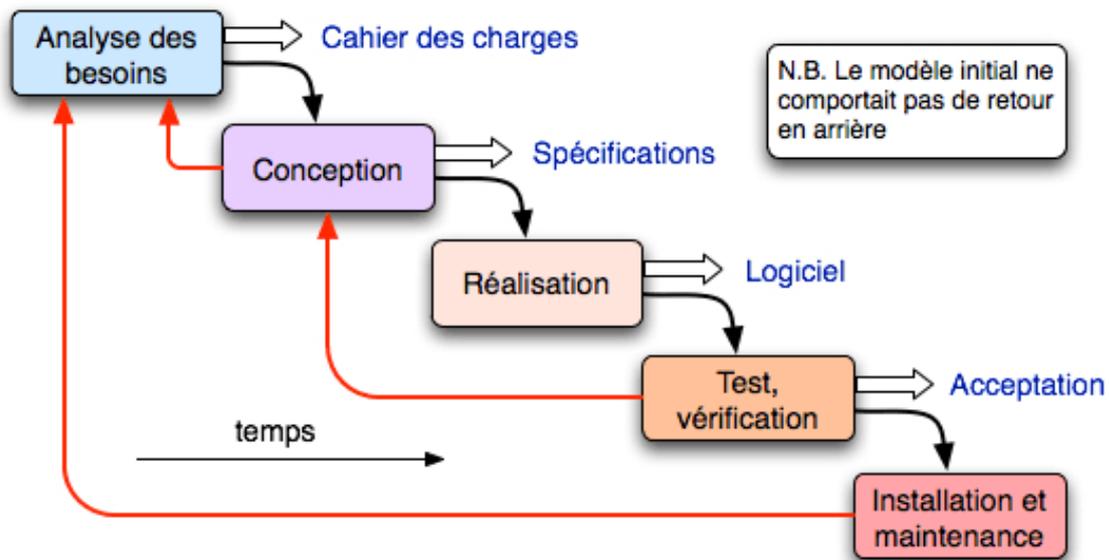


Figure. 7. Modèle du cycle de vie en cascade

En conclusion, Le modèle en cascade est un modèle de processus de développement logiciel avec un flux séquentiel linéaire. Une phase commence après l'achèvement de la phase précédente. Il n'y a pas de chevauchement entre les phases. Dans cette approche, l'ensemble du processus de développement logiciel est divisé en phases. Le résultat d'une phase devient l'entrée de la phase suivante.

Le modèle de cascade est simple et facile à comprendre. Il est facile d'organiser les tâches et de comprendre les jalons. Une seule phase est traitée et terminée à la fois. Le modèle en cascade ne convient pas pour développer des projets complexes. De plus, il ne convient pas à un projet aux exigences changeantes.

3.2. Le modèle en V (de cycle de vie en V) d'un projet (logiciel)

Dans les entreprises, tous les projets suivent une méthode de conduite de projet. La plus connue est la **Démarche de cycle en V**. De manière analogue à d'autres branches de l'ingénierie, la production du logiciel suit une série d'étapes de la démarche de cycle de V. À chacune de ces étapes sont associés des outils et des méthodes propres à atteindre les objectifs visés. De même, le cycle en V est une méthode d'organisation très connue qui tire ses origines de l'industrie lourde, à propos de la conception du système SAGE (Semi-Automatic Ground Environment), un dispositif militaire de surveillance et de défense aérienne. Adaptée à l'informatique dans les années 80 et imaginée pour remplacer le cycle en cascade, cette méthodologie est toujours utilisée dans le développement des projets informatiques, même si plusieurs dérivées existent. En outre, le cycle en V permet de définir un cadre mixte composé d'offshore « loin des côtes » et de local. Par l'implémentation de versions successives, le cycle recommence en proposant un produit de plus en plus complet.

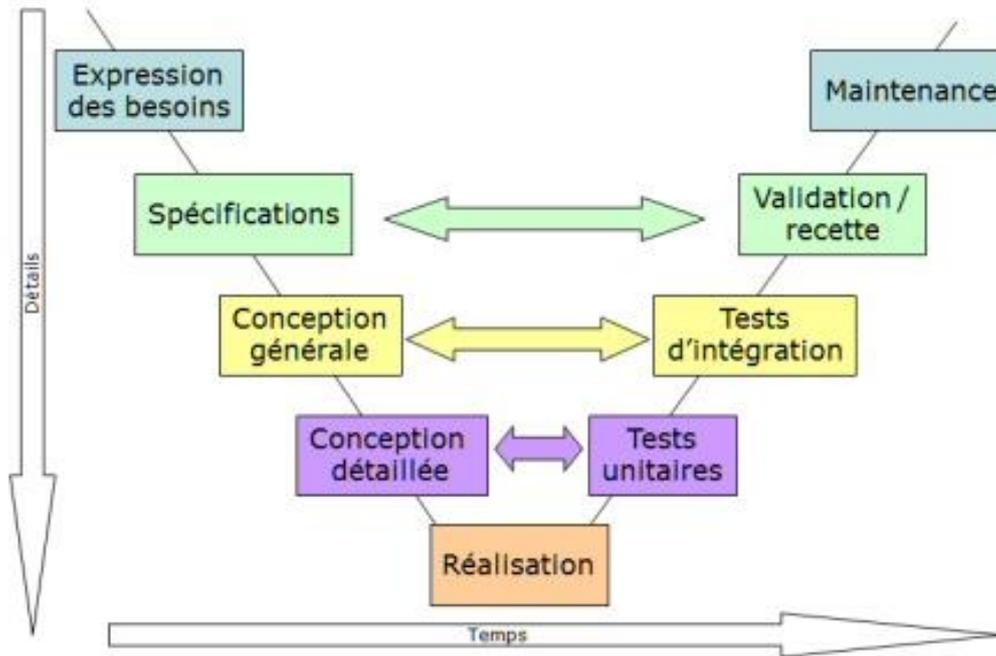


Figure. 8. Le principe du cycle en « V »

La figure précédente montre que la réalisation d'une application peut se découper en trois (03) grandes parties : La phase de conception, la phase de réalisation (codage) et la phase de validation. Les phases de conception et de validation se découpent en plusieurs parties. Chaque étape ne peut être réalisée qu'une fois que l'étape précédente est terminée, ce qui diminue les prises de risque sur le projet. Ce qui est bien visible sur le diagramme (figure 8), c'est que chaque étape de conception possède son alter ego de validation. Il devient alors assez aisé de valider un projet, car le référentiel de test est connu très précisément.

Par ailleurs, la méthode de projet sert à conduire le projet informatique à son terme en respectant les impératifs de qualité, coût et délai est le découpage du projet en phases. Chaque phase, comme on a signalé précédemment, est accompagnée d'une fin d'étape destinée à formaliser la validation de la phase écoulée avant de passer à la phase suivante. Ce qui caractérise une méthode itérative, c'est sa capacité à planifier une itération de production en termes de fonctionnalités et d'interdépendances.

De plus, tous les projets informatiques suivent une méthode de conduite de projet. Ils s'inscrivent dans les bonnes pratiques d'une entreprise. Pour mener à bien, la réalisation d'un projet informatique d'une certaine taille, le chef de projet doit découper le projet en plusieurs phases, et prévoir un environnement de travail et d'exécution par phase (voir figure 9).

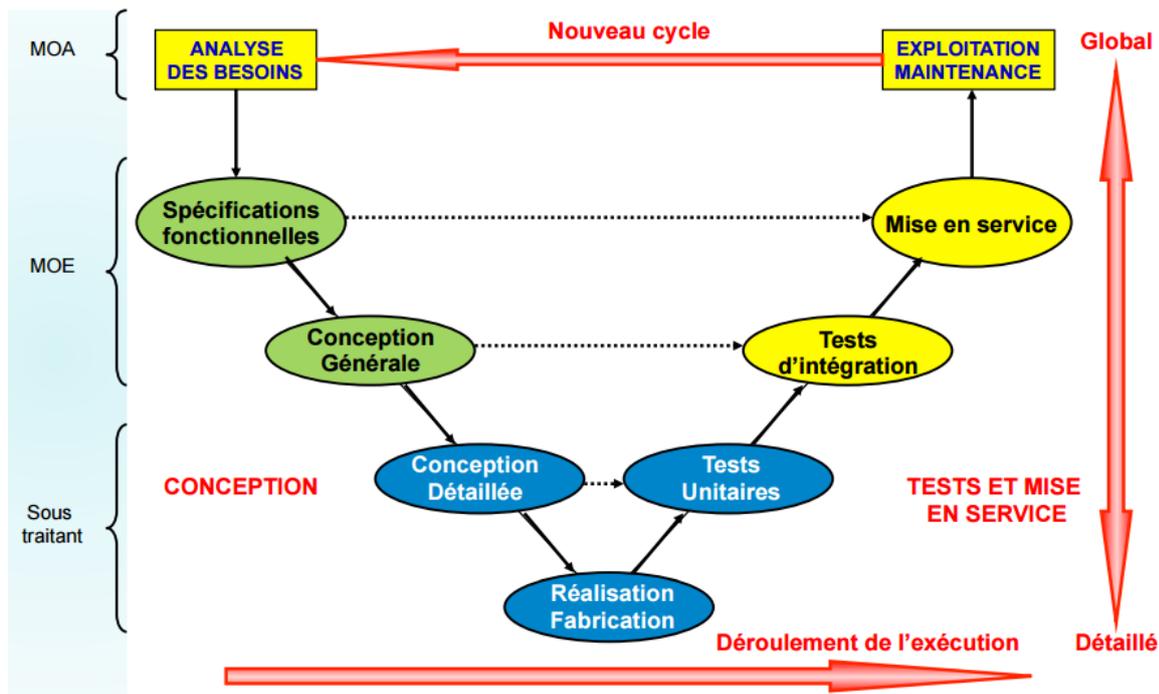


Figure. 9. Modèle du cycle de vie en V

Résumé : Ce cycle de vie permet un apprentissage et une adaptation permanents aux évolutions du projet.

3.3. Le modèle en spirale

Proposé par B. Boehm en 1988, ce modèle de cycle de vie tient compte de la possibilité de réévaluer les risques en cours de développement, il emprunte au prototypage incrémental mais lui adjoint une dimension relevant de la prise de décision managériale et non purement technique. Il couvre l'ensemble du cycle de développement d'un produit. Au contraire du modèle en cascade, le modèle en spirale ne part pas du principe que les tâches du développement logiciel doivent être organisées de manière linéaire, mais de manière itérative. À travers cette répétition cyclique, le projet avance relativement lentement vers les objectifs fixés, mais en contrepartie le risque que le processus de développement échoue est drastiquement réduit au moyen de contrôles réguliers.

Autrement dit, c'est un modèle qui repose sur le fait qu'il y a une relation contractuelle entre le fournisseur et le client. Plusieurs cycles sont effectués. Chaque cycle donne lieu à une contractualisation s'appuyant sur les besoins exprimés lors du cycle précédent. Une caractéristique clé du modèle en spirale est la minimisation des risques dans le développement logiciel, ce qui peut entraîner une augmentation des coûts globaux, ainsi que plus d'efforts et un lancement différé. Ces risques sont contrôlés par l'approche progressive en réalisant d'abord des prototypes, qui sont répétés dans les spirales ou les cycles de développement de logiciels au moins une fois. Un cycle est composé des étapes suivantes (voir figure 10) :

- Objectif et décision alternative : les objectifs sont déterminés conjointement avec le client. Dans le même temps, les alternatives possibles sont discutées et les conditions cadres sont spécifiées (par exemple le système d'exploitation, l'environnement et langage de programmation).

- Analyse du risque : les risques potentiels sont identifiés et évalués.
- Développement d'un prototype : les prototypes sont encore plus étendus et des fonctionnalités sont ajoutées. Le code réel est écrit, testé et migré vers un environnement de test plusieurs fois jusqu'à ce que le logiciel puisse être implémenté dans un environnement productif.
- Simulation et essais du prototype : Les alternatives en question sont également évaluées, tandis que les risques sont enregistrés, estimés puis réduits à l'aide de prototypes, des simulations et des logiciels d'analyse.
- Détermination des besoins (à des mailles différentes selon le cycle), à partir des résultats des essais ;
- Validation des besoins par un comité de pilotage ;
- Planification du cycle suivant : le cycle à venir est planifié à la fin de chaque cycle. Si des erreurs se produisent, les solutions sont recherchées. Si une meilleure alternative est une solution envisageable, elle sera préférée au sein du cycle suivant.

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

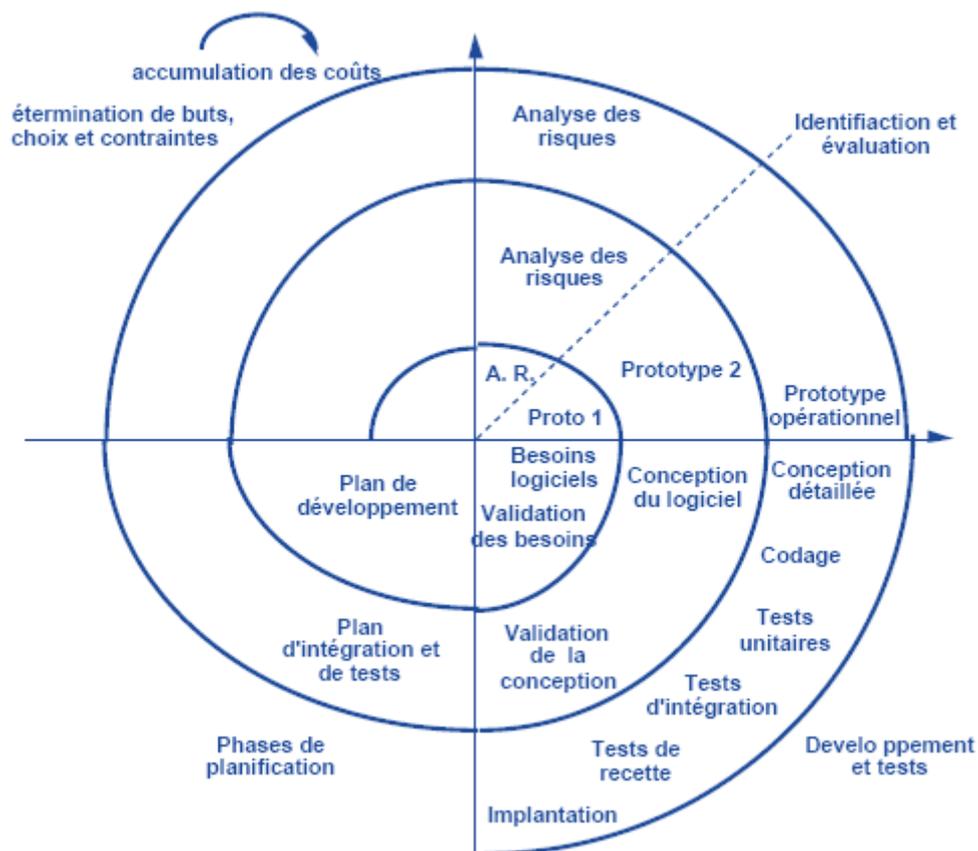


Figure. 10. Modèle du cycle de vie en Spirale

Ainsi, la force motrice la plus importante du modèle en spirale est l'analyse et l'évaluation des risques. Tout risque qui menace le projet est censé être identifié dès le début. L'avancement du projet dépend de manière décisive de la manière dont les risques peuvent être limités ou supprimés. Le projet ne sera considéré comme réussi quand il n'y aura plus de risque. Le but du cycle est de produire un produit qui est en amélioration continue. Le logiciel ou l'application est constamment affiné. Le modèle en spirale est incrémental, mais pas nécessairement répétitif. Les répétitions ne se produisent que lorsque des risques, des erreurs ou des conflits menacent

le projet. Ensuite seulement, le produit passera à un cycle suivant, ce qu'on appellera une répétition ou une itération.

Chaque enroulement de la spirale matérialise un cycle terminé, après avoir franchi tour à tour quatre quadrants différents, qui représentent dans ce cas les quatre phases possibles du modèle. Au fur et à mesure que la taille de la spirale grandit, la progression avance, de même que la validation par le contrôle (Axe X) et les coûts de développement (Axe Y). Par une analyse régulière des risques et des contrôles réguliers du produit intermédiaire, le modèle en spirale diminue considérablement le risque d'échec lors des projets logiciels de grande taille. En fin, le modèle en spirale est également générique et peut être combiné avec d'autres méthodes de développement agile classiques, c'est pourquoi il est également appelé modèle de second ordre. En conclusion, la différence clé entre le modèle en cascade et le modèle spirale (itératif) est que le modèle en cascade est utilisé pour les petits projets et les projets avec des exigences claires, tandis que le modèle en spirale est utilisé pour les grands projets complexes qui nécessitent une analyse continue des risques.

3.4. Le modèle méthode « Agile »

Depuis plusieurs années, la gestion des projets informatiques repose sur la méthode agile. La méthodologie Agile est un processus qui permet à l'équipe de gérer un projet en le décomposant en plusieurs étapes. Elle implique une collaboration constante entre les parties prenantes, une amélioration et une itération continues à chaque étape. En substance, la méthode agile consiste en des méthodes de gestion et d'évolution de projet qui reposent sur le découpage du projet sous forme de sprint. De plus, elle se base sur la notion de communauté de projet dans laquelle les développeurs et les utilisateurs sont présents en permanence pour exprimer ou répondre à une question liée au projet. De même, elle repose sur des itérations : le projet est donc divisé en plusieurs étapes (sous-projets) qui sont chacune des objectifs à court terme. Il y aura ensuite un certain nombre d'itérations qui seront réalisées jusqu'à atteindre l'objectif final.

Par ailleurs, cette méthode a aujourd'hui largement fait ses preuves et elle fonctionne dans de nombreux projets (création de site internet, développement d'une application, d'un logiciel spécifique ou d'une couche applicative supplémentaire, etc.). En revanche, elle nécessite une adaptation / une révision des contrats informatiques qui dans la majorité des cas ne prennent pas en compte les effets engendrés par l'utilisation de cette méthode.

Néanmoins, l'utilisation de la méthode agile a également une incidence sur la forme du calendrier prévisionnel transmis au client (MOA). Le calendrier doit désormais s'appuyer sur les étapes de sprint et plus sur les étapes mentionnées plus haut (expression des besoins, phase de conception puis intégration...).

En conclusion, la méthode agile est parfaitement compatible avec la réussite d'un projet informatique. Il suffit pour le client et le prestataire de réviser et personnaliser le contrat informatique qui les unit.

3.4.1. La Méthode AGILE : quels sont les principes ?

Une méthodologie agile prévoit la fixation d'objectifs à court terme utilisés dans la gestion de projets. Néanmoins, la méthode traditionnelle prévoit la planification totale du projet avant même la phase développement. Ainsi, le projet est fragmenté en plusieurs sous-parties que les équipes de développement qui en a la charge se doivent d'atteindre progressivement en ajustant

si nécessaire les objectifs pour répondre le plus possible aux attentes du client. Nous parlons alors de sprints. Chaque sprint ayant pour objectif de clôturer une brique du projet. Les méthodes agiles permettent de renforcer les relations entre les membres d'une équipe, mais aussi entre l'équipe et le client. C'est pour cette raison que la flexibilité et la souplesse sont deux piliers forts dans la méthode AGILE. Nous parlons d'un mode itératif qui permet de délivrer des prestations avec efficacité et réactivité le tout en respectant des plannings exigeants. Cela permet de gérer ses projets de façon performante.

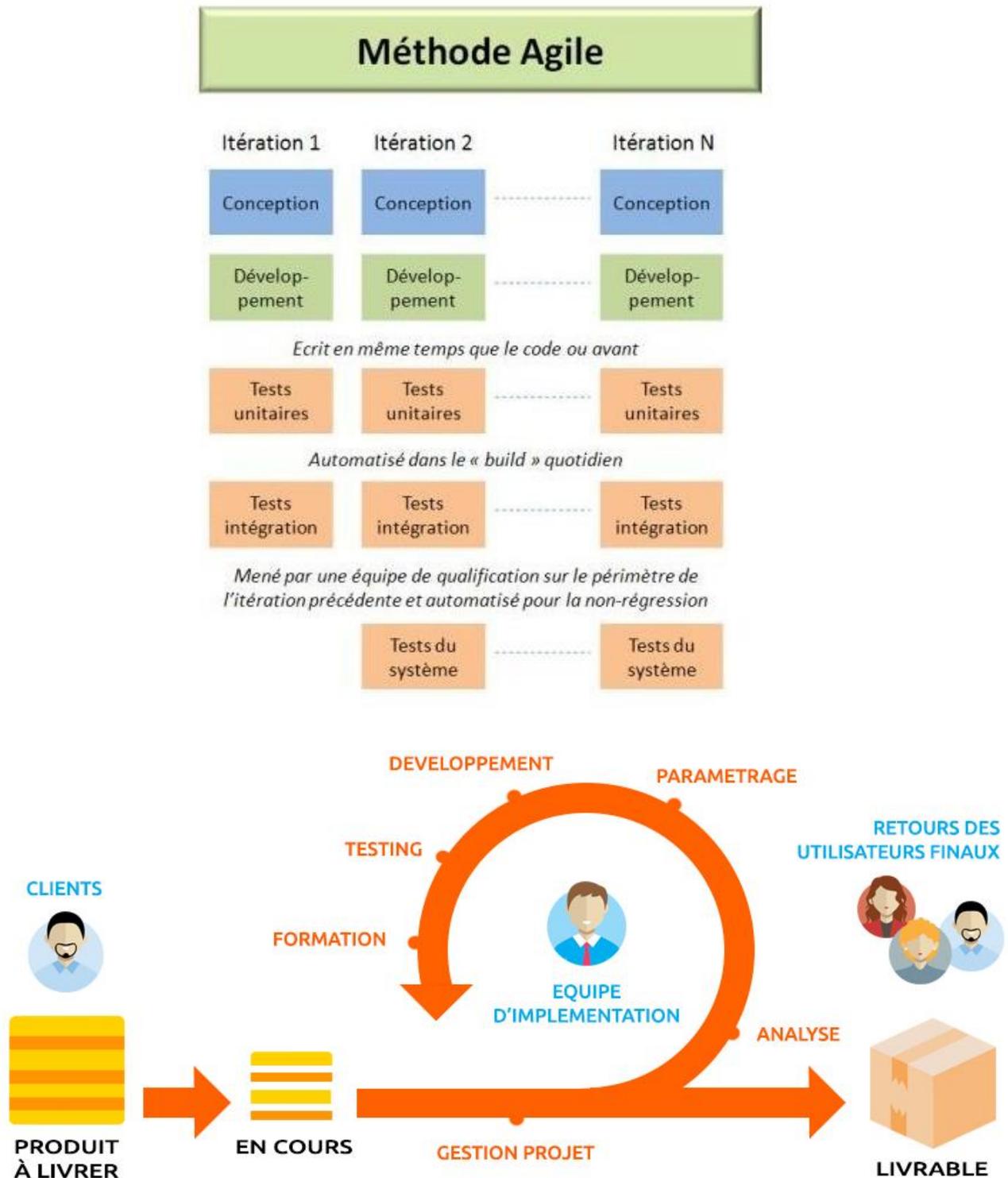


Figure. 11. Modèle méthode « Agile » (a)

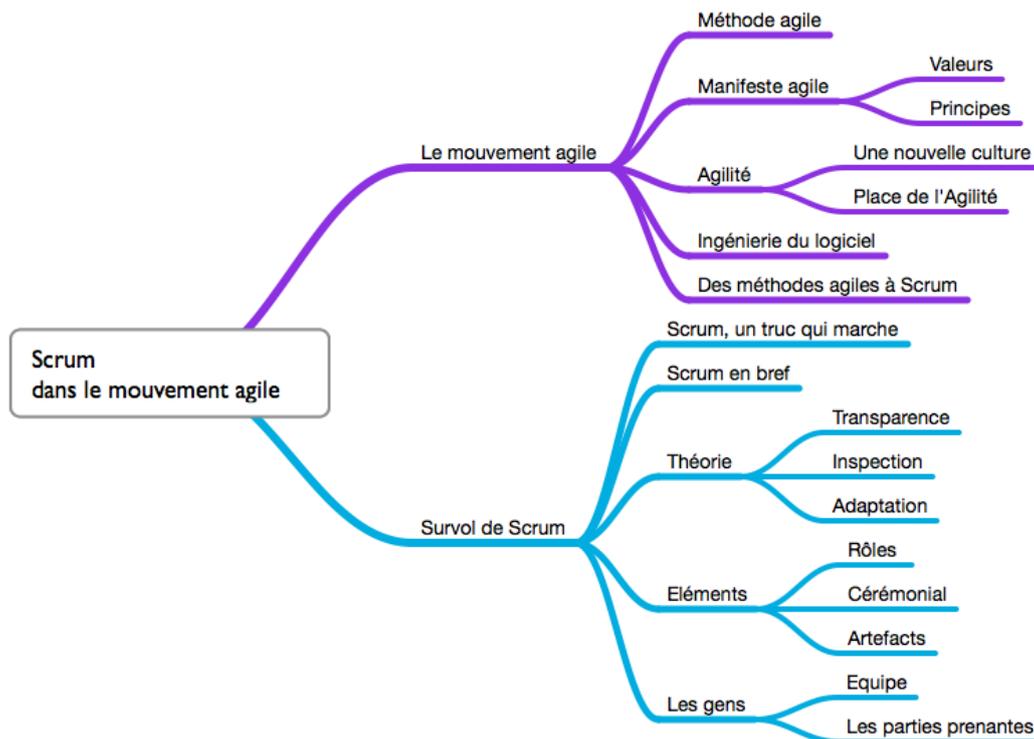


Figure. 11. Scrum dans le mouvement Agile (b)

Dans ce but, les méthodes agiles prônent quatre (4) valeurs fondamentales :

1. **L'équipe** «Personnes et interaction plutôt que processus et outils». La communication est une notion fondamentale.
2. **L'application** «Logiciel fonctionnel plutôt que documentation complète». Il est vital que l'application fonctionne.
3. **La collaboration** «Collaboration avec le client (MOA) plutôt que négociation de contrat». Le client doit être impliqué dans le développement.
4. **L'acceptation du changement** «Réagir au changement plutôt que suivre un plan». La planification initiale et la structure du logiciel doivent être flexibles (évolution de la demande du client)

Finalement, on peut dire que le choix des modèles se fait selon certains critères tel que la nature du projet, sa taille, la nature du client et les compétences de l'équipe.

3.4.2. Quelques outils d'aide à la gestion d'une équipe « Agile »

De nombreux outils d'aide à la gestion d'une équipe « Agile » ont fait leur apparition ces dernières années. On peut en citer Agilefant, IceScrum, Agilo, eXPlainPMT ou encore XPlanner. D'autres sont apparus dernièrement comme le JIRA Agile qui a l'avantage d'être facilement utilisable même pour les débutants. Il permet d'élaborer des tableaux de tâches, de créer et d'estimer les Récits Utilisateurs (user stories (US)), d'identifier l'engagement et la vélocité de l'équipe ou encore de créer divers rapports sur l'état d'avancement des projets. En outre, un outil comme PMTool offre un large choix de rapports dans une interface simple mais très complète. Ainsi, les développeurs tout comme les chefs de projet (Scrum master) dispose d'une interface simple pour saisir leur feuille de temps et rendre compte de leurs activités.

Un tableau de bord personnalisable est également disponible pour permettre à l'utilisateur d'avoir sous ses yeux les principaux indicateurs pour le suivi de son projet/développement tout

en prenant en compte l'itération en cours. Les questions/incidents pourront enfin être gérée de manière très aisée car cette action est centralisée et se fait en un seul clic grâce à un menu très discret qui s'affiche partout (quel que soit la rubrique en cours de consultation).

3.5. Synthèse des différences fondamentales entre méthode classique et méthode agile

La synthèse ci-après présente, dans le tableau 1, les différences majeures par thème entre une méthode classique et une méthode agile.

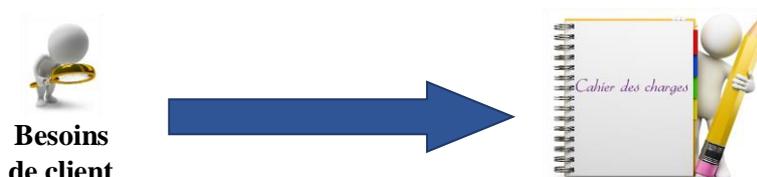
Thème	Méthode classique	Méthode agile
Cycle de vie	En cascade ou en V, sans rétroaction possible, phases séquentielles.	Itératif et incrémental.
Planification	Prédictive, caractérisée par des plans plus ou moins détaillés sur la base d'un périmètre et d'exigences définies et stables au début du projet.	Adaptative avec plusieurs niveaux de planification (macro- et micro planification) avec ajustements si nécessaires au fil de l'eau en fonction des changements survenus.
Documentation	Produite en quantité importante comme support de communication, de validation et de contractualisation.	Réduite au strict nécessaire au profit d'incrément fonctionnels opérationnels pour obtenir le feedback du client.
Equipe	Une équipe avec des ressources spécialisées, dirigées par un chef de projet.	Une équipe responsabilisée où l'initiative et la communication sont privilégiées, soutenue par le chef de projet.
Qualité	Contrôle qualité à la fin du cycle de développement. Le client découvre le produit fini.	Un contrôle qualité précoce et permanent, au niveau du produit et du processus. Le client visualise les résultats tôt et fréquemment.
Changement	Résistance voire opposition au changement. Processus lourds de gestion des changements acceptés.	Accueil favorable au changement inéluctable, intégré dans le processus.
Suivi de l'avancement	Mesure de la conformité aux plans initiaux. Analyse des écarts.	Un seul indicateur d'avancement : le nombre de fonctionnalités implémentées et le travail restant à faire.
Mesure de succès	Respect des engagements initiaux en termes de coûts, de budget et de niveau de qualité.	Satisfaction client par la livraison de valeur ajoutée.

Tableau. 2. les différences entre une méthode classique et une méthode agile.

4. Principe et différentes phases du cycle de vie d'un projet IT

Le cycle de vie est constitué de huit (08) étapes qui ont toutes leur importance :

- **Analyse (Expression) des besoins** : le client (MOA) exprime son besoin par un cahier de charge, en décrivant les usages correspondant au produit fini tel qu'il peut l'imaginer. Cela doit répondre aux questions « Que veut-on ? » et « A quel coût ? ».



- Le cahier des charges répond à la question :
« *Qu'attend-t-on de l'équipe projet ?* »
- C'est le **contrat** entre le client et l'équipe projet

Figure. 12. Relation entre l'analyse des besoins et le cahier de charge

- **Spécifications fonctionnelles (MOE) :** c'est le cahier des charges exact du produit final, tel que le désire le client (MOA). Il doit couvrir l'intégralité des cas d'utilisation du produit, en expliquant ce qu'il doit faire et non pas comment il va le faire. Le cahier des charges décrit, en langage naturel, les fonctionnalités attendues du produit ainsi que les contraintes non fonctionnelles (temps de réponse, contraintes mémoire...). Dans le cas de la refonte d'un système complet on peut avoir un cahier des charges par sous domaine.
- **Spécifications techniques :** c'est une traduction des spécifications fonctionnelles (MOE) en termes techniques. C'est durant l'élaboration des spécifications techniques que sont choisies les technologies à mettre en œuvre pour développer le produit, et qu'est conçue l'architecture logicielle du produit.
- **Codage :** c'est la phase de réalisation à proprement parler, pendant laquelle sont développées des briques qui sont ensuite assemblées pour créer le produit fini. A l'issue de cette phase les produits intermédiaires sont : les modules codés, la documentation de chaque module et le planning mis à jour.
- **Documentation :** Une documentation doit nécessairement accompagner l'ouvrage lors de la livraison. Les acteurs de la documentation sont les MOA/MOE/MOE déléguée. Il y a trois (3) types de documents sont identifiés :
 - a. *Manuel utilisateur* : manuel expliquant le fonctionnement du logiciel aux utilisateurs ;
 - b. *Manuel d'exploitation* : détaille les procédures pour le maintien opérationnel de la solution (modalité d'arrêt des bases, backup, emplacement des fichiers log, procédure de crash recovery, détail des chaînes de nuit, etc.) ;
 - c. *Manuel d'administration* : détaille toutes les options de paramétrage, de configuration et d'administration du logiciel. Lors de cette étape qui intervient après la réalisation du projet, on établit généralement le plan de test pour la recette (MOE + MOA). Ce dossier test contient également le plan des tests unitaires.
- **Tests unitaires :** ces tests interviennent à un niveau « atomique ». chaque brique logicielle a été modélisée puis codée durant les étapes précédentes. Les tests unitaires assurent que ces briques respectent de manière individuelle leur cahier des charges. l'issue de cette phase les produits intermédiaires sont : les modules testés, les résultats des tests unitaires et le planning mis à jour.
- **Tests d'intégration :** ce sont là les premiers tests grandeur nature du produit fini. On s'assure qu'il suit les indications des spécifications techniques. l'issue de cette phase, les produits intermédiaires sont : le logiciel testé, les tests de régression, le manuel d'installation et la version finale du manuel utilisateur.
- **Validation :** le produit est à ce moment testé en regard de la spécification fonctionnelle. Tous les utilisateurs qui y ont été définies doivent pouvoir se vérifier dans les faits. l'issue de cette phase, le logiciel est prêt à la mise en exploitation.

- **Mise en production et recette** : le produit est vérifié une dernière fois pré-production, avant d'être mis en production. Le client (MOA) procède à la recette, pour vérifier que son expression (analyse) de besoin est respectée. Lorsque le produit a été accepté, il passe en phase de maintenance jusqu'à son retrait. C'est pendant cette phase que tous les efforts de documentation faits pendant le développement seront particulièrement appréciés de même que la transparence de l'architecture et du code.

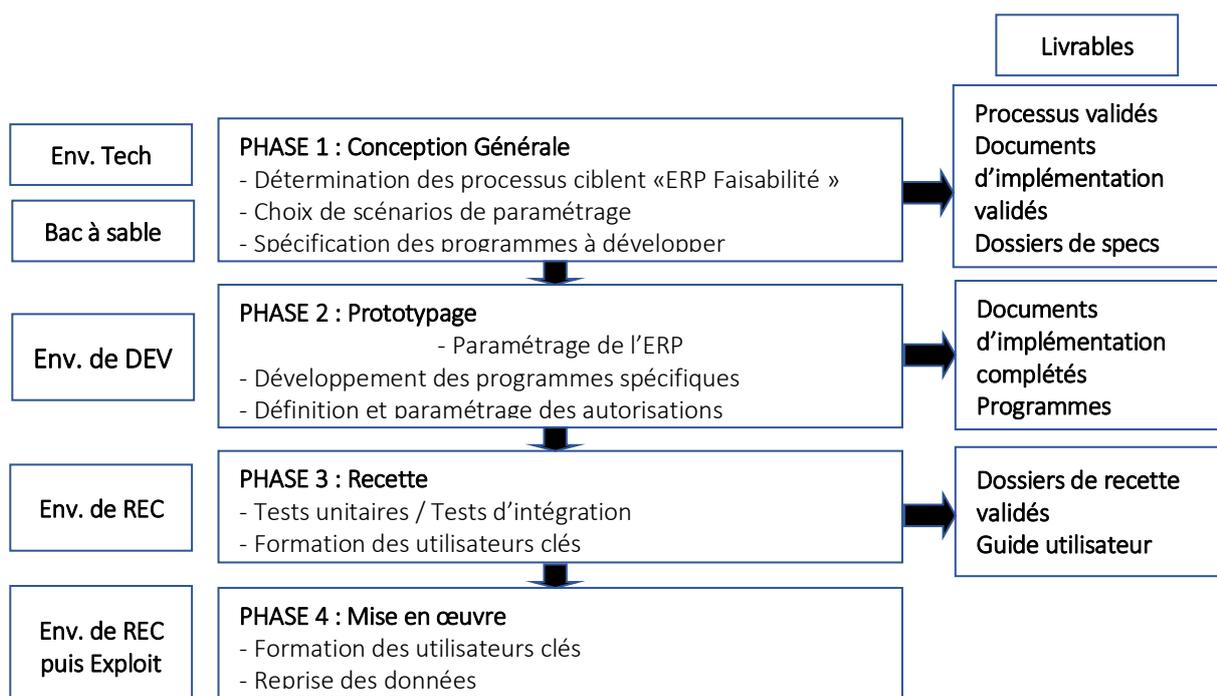
Enfin, le **principe du cycle en vie** d'un logiciel est de limiter les retours aux étapes précédentes et ainsi d'éviter par exemple de devoir revenir sur les spécifications et sur le développement une fois.

Exemple 1 :

1. Présenter la méthodologie de conduite de projet Web ?
2. Mettre en place un projet ERP ?
3. Quelles sont les dérives à éviter à la création de votre équipe projet ERP ?
4. Quelles sont les critères de choix d'un projet ERP ?

1. **Solution** : voir Ci-dessus

2. **Solution** : Mise en place d'un projet ERP



3. Nous donnons quelques conseils à retenir pour la structuration de cette équipe projet pour éviter les dérives pendant la création de votre équipe projet ERP :

- Doit être parfaitement dimensionnée par rapport à la taille du projet,
- Doit impliquer au moins une personne de la direction,
- Ne doit pas intégrer uniquement des responsables opérationnels,
- Ne doit pas concentrer toutes les responsabilités sur un seul individu,
- Avoir des responsabilités clés confiées aux individus dédiés à 100% au projet.

4. Nous trouvons dans la littérature 7 Critères de réussite d'un projet ERP qui sont :

1ère Critère : en fonction de vos besoins : Votre futur système ERP doit parfaitement répondre à vos priorités.

2nd critère : en fonction de votre organisation : Un ERP peut participer à l'ensemble des opérations d'une entreprise. Il doit être calqué sur l'organisation de la société.

3ème critère : en fonction du coût : Le coût d'intégration d'un logiciel ERP est vite abordé. Les budgets ne sont pas extensibles.

4ème critère : en fonction de l'expérience utilisateur : L'adoption d'un nouvel outil de gestion est aussi un défi pour l'entreprise. La résistance au changement peut être plus ou moins forte selon les organisations et les habitudes de travail.

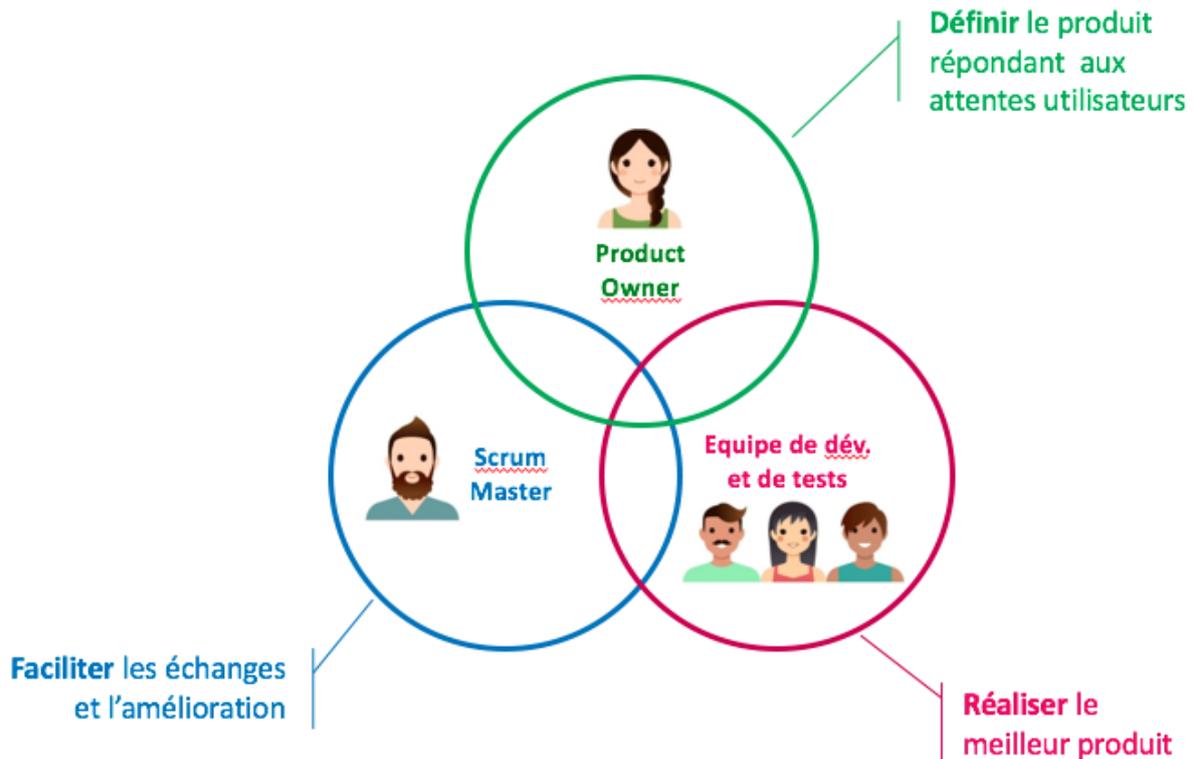
5ème critère : en fonction de l'accompagnement : Un projet ERP est le choix d'une solution, mais c'est aussi choisir un prestataire. **6ème critère** : en fonction de votre système d'information : Une entreprise peut utiliser une multitude d'outils se complétant et composant son système d'information actuel.

7ème critère : en fonction de la technologie : Les logiciels ERP on-premise (ou sur site) ont longtemps été la norme en entreprise. Avec l'arrivée du cloud, la plupart des éditeurs se sont rués sur la technologie et proposent les deux formules au client.

Exemple 2 :

Question : Est-ce que le rôle de Chef de Projet disparaît dans la méthode Scrum ?

Réponse : La nature même de l'approche agile conduit à changer de paradigme et à remplacer la notion de projet par celle de produit. Par conséquent, le rôle de chef de projet, tel que connu dans l'approche « Cycle en V » disparaît. Cela semble logique : pas de projet, pas de Chef de Projet. Les créateurs de Scrum font le choix de clarifier les rôles et les responsabilités, au sein de l'équipe, en faisant émerger des acteurs sur chacun des axes produit : le fonctionnel, l'organisationnel et la technique (voir figure ci-dessous).



Dans une équipe Scrum, les fonctions du Chef de Projet, comme le planning, la coordination et le suivi, ont donc été répartis à l'ensemble de l'équipe Scrum. D'où l'inquiétude de certains et ce questionnement : quel est donc l'avenir d'un Chef de Projet ?

Réponse : Si un Chef de Projet veut intégrer aujourd'hui une équipe agile de type Scrum, on lui recommanderait donc de se positionner sur l'un de ces 3 axes en précisant sa préférence.

On m'explique : Si son appétence est liée à la définition du produit à réaliser, on lui proposerait de devenir Product Owner. Si son appétence est liée à l'organisation et à la coordination des acteurs de l'équipe, on lui proposerait de devenir Scrum Master. Si son appétence est liée à la réalisation du produit, on lui proposerait de devenir leader technique, développeur ou même testeur.

Chapitre 4

Les outils de structuration du projet IT

1. Les outils de structuration du projet informatique

La structuration d'un projet consiste à **comprendre**, **explicitier** et **formaliser** les différents **livrables** à produire dans le cadre du projet, puis à établir la liste des **tâches** qui seront nécessaires pour aboutir à ces productions. Nous trouvons dans la littérature plusieurs structures arborescentes hiérarchiques qui composent le projet ou qui peuvent être utilisées pour découper un projet notamment : Le découpage par poids, par taille, par tâches, par les ressources, par le coût, par la documentation et même par les grandes étapes de réalisation. Dans cet ouvrage on va présenter ci-dessous le PBS, le WBS et l'OBS.

1.1. La PBS : Référentiel du produit ou Project Breakdown Structure (PBS)

La PBS² - Product Breakdown Structure - est la liste hiérarchisée des livrables du projet. Le "Product Breakdown Structure" (PBS) répond au quoi ?. Connue aussi sous le nom d'organigramme produit. En outre, il permet de définir les livrables du projet et de structurer les réalisations intermédiaires. Il définit une structuration du projet et un découpage en sous-projets et ainsi de décider s'il convient d'en sous-traiter une partie. C'est la décomposition arborescente du produit en éléments plus ou moins fins. En outre, le PBS a pour objectifs de définir la nomenclature des objectifs du projet, de décomposer l'objet du projet en sous-ensembles et de définir les compétences nécessaires à la mise en place du projet.

Exemple. Pour un logiciel. Chaque nœud est une sous-partie du produit informatique. Cette décomposition sert à avoir une vision modulaire et hiérarchique du produit, dans l'optique de répartir les tâches et les responsabilités entre les acteurs (voir figure 13).

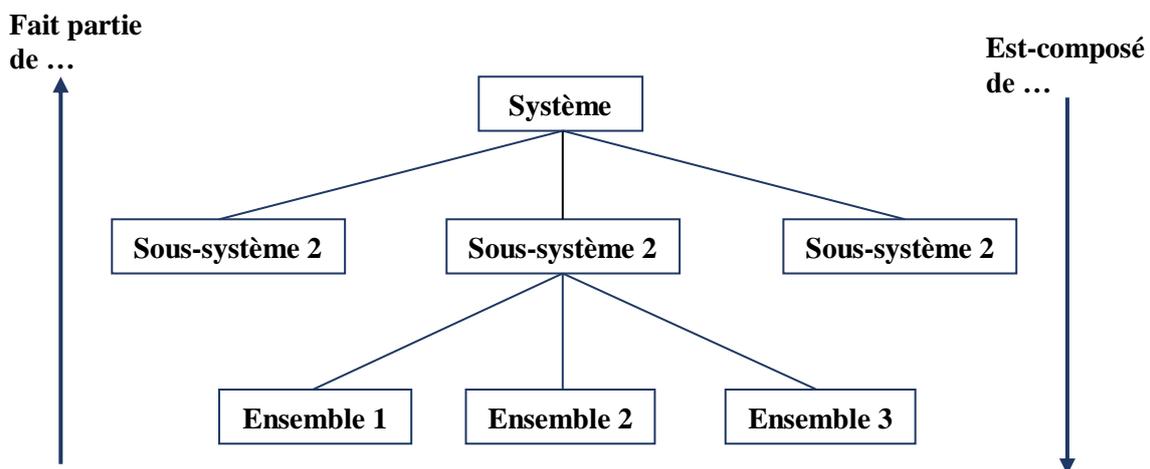


Figure. 13. Découpage du système en unités physiques hiérarchisées (PBS)

Quoi : (PBS : Product Breakdown Structure) : décomposition du projet en produits livrables.

² En français OTP pour Organigramme Technique Produit ou encore AP pour arborescence produit.

1.2 La structure de découpage de projet (SDP) « WBS - Work Breakdown Structure » ou « OT - Organigramme des Tâches »

La structure de découpage du projet³, aussi parfois appelée structure de fractionnement des tâches ou encore « organigramme des tâches », est une division hiérarchique du travail global à réaliser, répartie en résultats de travail ou livrables qui peuvent eux-mêmes être subdivisés en lots de travaux. Les lots de travaux peuvent être estimés, planifiés et confiés à une personne nommée qui en assurera la réalisation ou la coordination de la réalisation. La structure de découpage du projet donne donc une vue hiérarchique et graphique du projet.

La structure de découpage du projet permet :

- de valider les objectifs et l'envergure du projet en proposant une formalisation graphique qui définit les divers rôles, identifie les tâches, les activités, ou, le cas échéant, les lots de travaux ainsi que les relations logiques entre les différents éléments ;
- de suivre et contrôler le déroulement du projet en suivant l'état de réalisation des tâches et activités et d'en communiquer l'état aux parties prenantes ;

Elle permet aussi de faciliter la planification du travail à effectuer, d'estimer la durée totale du projet et de déterminer les ressources nécessaires pour chaque étape.

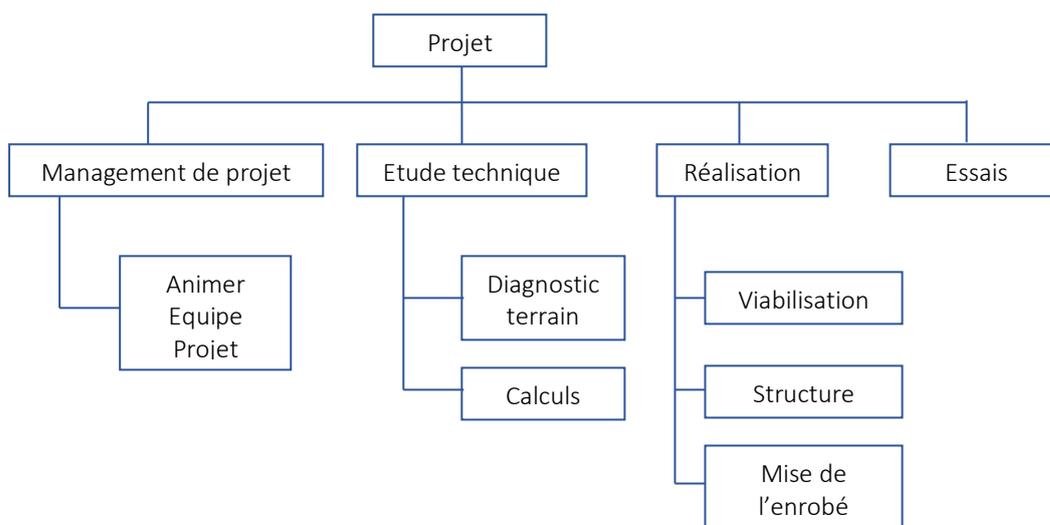


Figure. 14. Organigramme des tâches par WBS

Dans une structure de découpage du projet (SDP), chacun des livrables est subdivisé en composants plus petits : **le lot de travail**.

Le lot de travail est le niveau le plus bas de la structure de découpage parce qu'il est plus petits et moins complexes. Ainsi, le coût et l'échéancier de réalisation de chaque composant d'un livrable peut être estimé de façon plus fiable. La fiche de chacun des lots de travail doit comporter les informations suivantes :

- un titre et une description de la tâche ;
- un responsable unique ;
- une durée d'exécution exprimée en jours ou en heures ;

³ La structure de découpage du projet correspond au WBS (Work Breakdown Structure) dans la terminologie anglophone.

- une description des ressources nécessaires à son exécution :
 - les ressources humaines,
 - les ressources matérielles,
- un coût estimé ;
- une description des extrants attendus au terme de la tâche.

Néanmoins, le WBS est directement lié au PBS : à chaque nœud du PBS correspond une tâche du WBS.

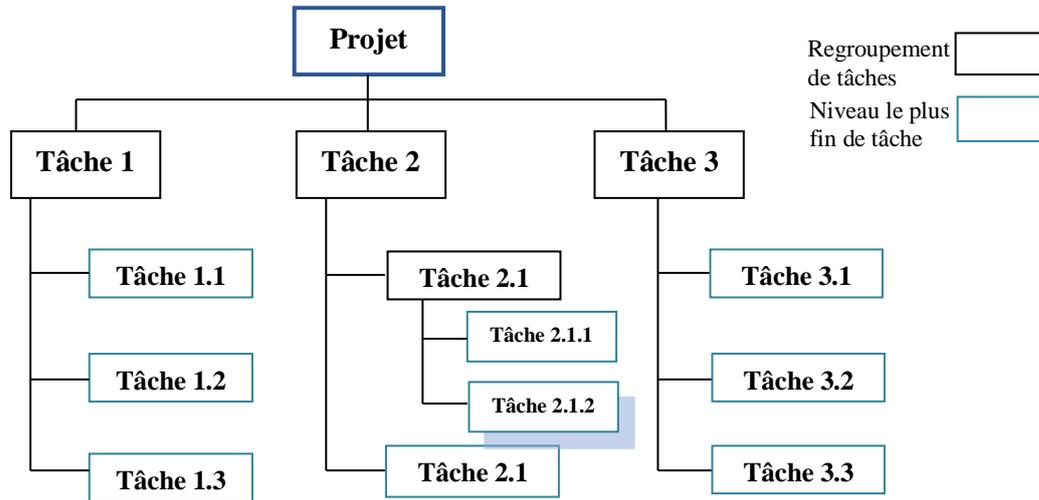
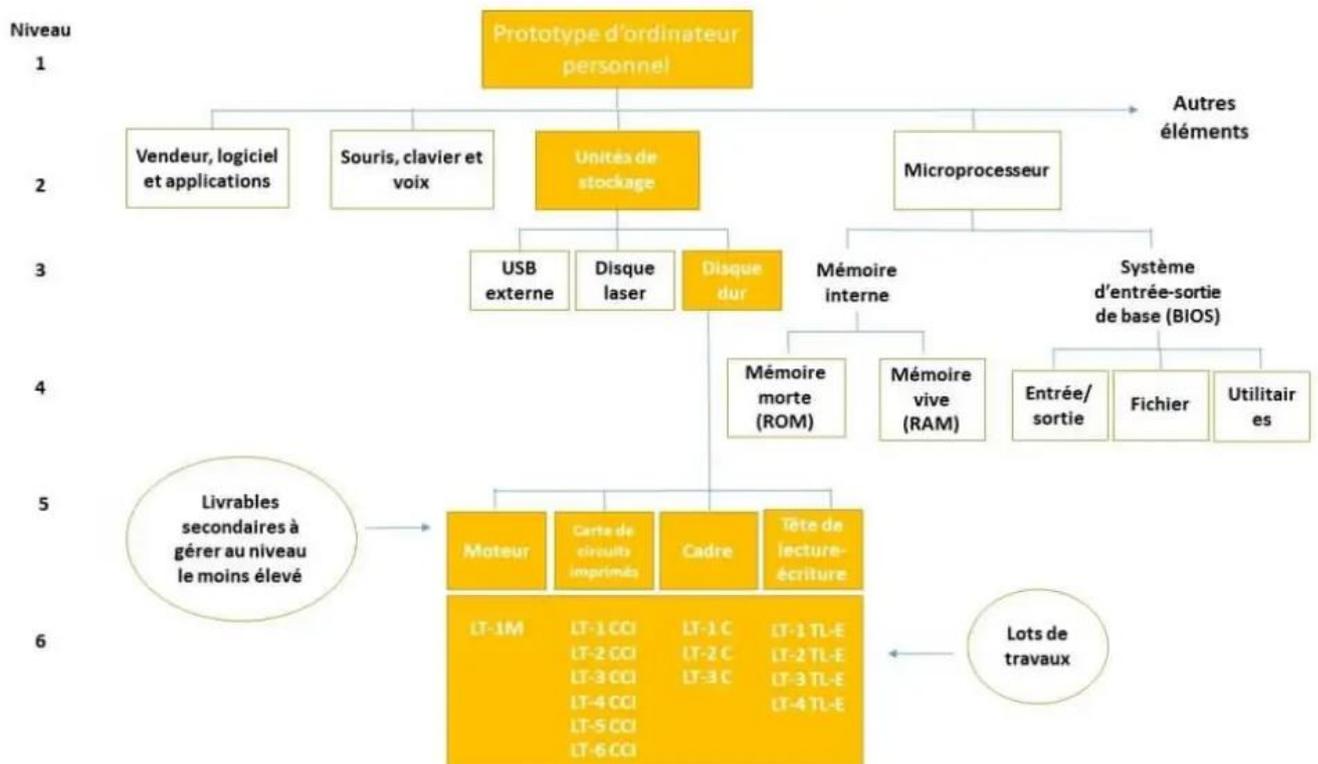


Figure. 15. Modèle de structure de découpage d'un projet

Comment : (WBS : Work Breakdown Structure) : décomposition en tâches élémentaires

En conclusion, la structure de découpage de projet ou le WBS projet est un outil essentiel de la gestion de projet et un complément du diagramme de réseau puis du diagramme de GANTT. Ces trois outils combinés vont vous donner une visibilité unique de votre projet et du travail à effectuer.

Exemple :



Dans notre exemple de l'ordinateur personnel, on peut bien sûr détailler encore au-delà du niveau 4, mais nous nous sommes arrêtés ici car le niveau est suffisant pour estimer le travail à faire.

En effet, la décomposition des livrables du projet doit permettre de créer des tâches individuelles pour lesquelles il est possible :

- D'affecter une ou plusieurs ressources
- D'estimer la charge de réalisation

Exercice :

Étude de cas simple : La tâche consiste à implanter un mini compilateur d'un sous-ensemble de C qui offre également une interface usager pour écrire le code et le compiler.

- Identifier les activités du projet en précisant les types de décomposition
- Allouer des ressources à chaque activité
- Allouer du temps à chaque activité.

Solution :

Compilateur C	
<i>Commencement du projet</i>	
	Allouer les ressources
	Etablir l'environnement du projet
	Planifier la gestion du projet
<i>Contrôle du projet</i>	
	Identifier et analyser les risques
	Composer le plan d'aversion de risques
<i>Gestion de contrôle du projet</i>	
	Définir les métriques
	Gérer la qualité du logiciel
Compilateur C	
<i>Besoins</i>	

	Analyser les fonctions
	Développer l'architecture de système
	Développer les besoins et les spécifications
	Définir les besoins sur l'interface
Conception	
	Concevoir l'interface graphique
	Concevoir le parseur
	Concevoir l'analyseur syntaxique et sémantique
	Concevoir le générateur du code
Implantation et tests unitaires	
	Coder l'interface graphique et faire les tests unitaires
	Coder le parseur et faire les tests unitaires
	Coder l'analyseur syntaxique et sémantique et faire les tests unitaires
	Coder le générateur du code et faire les tests unitaires
Intégration	
	Installer le compilateur
	Tester l'interface graphique
	Tester le parseur
	Tester le générateur du code
	Tester l'analyseur syntaxique et sémantique
Documentation	
	Ecrire les documents d'utilisateur
	Ecrire les documents de développeur

1.3. Organisation Breakdown Structure (OBS)/ Découpage de l'organisation de projet

Il permet d'identifier et établir les responsabilités ainsi que les services concernés par la réalisation des tâches et donc l'achèvement des produits du projet. De plus, le découpage par responsabilités ou "Organization Breakdown Structure" (OBS) se base par la responsabilité des intervenants grâce à la constitution d'une arborescence ou "Organization Breakdown Structure" (OBS) qui attribue un domaine de compétences propres des participants. Cet organigramme est réalisé après le PBS et le WBS.

Cette organisation peut se faire :

1. dans le cadre de l'équipe projet ;
2. au niveau d'un service au sein de l'entreprise ;
3. à l'extérieur de l'entreprise dans le cadre de la "sous-traitance" etc.

En outre, l'OBS peut représenter l'organisation des équipes côté MOE (ou prestataire) et côté MOA (ou client).

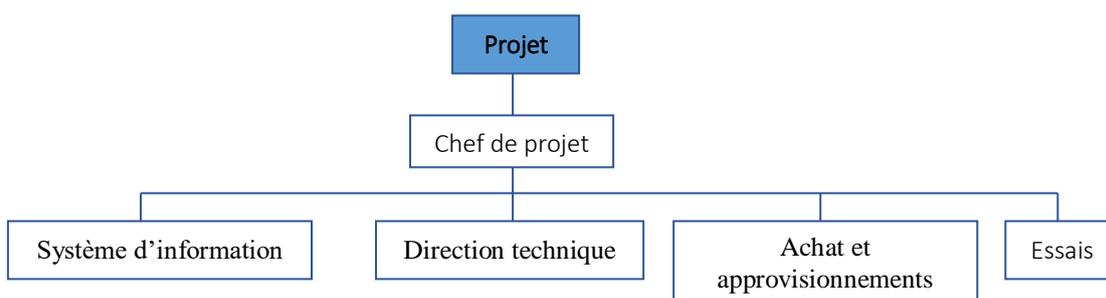


Figure. 16. Organigramme OBS

1.4. Cost Breakdown Structure (CBS) : « Combien ? »

La structure de décomposition des coûts (CBS) consolide les résultats de l'estimation des coûts sur le WBS. Le CBS est présenté au tableau suivant qui indique une première estimation des coûts par le chef de projet.

Libellé WBS Niveau 1	Libellé WBS Niveau 2	Lot de travaux	Code Lot de travaux	Coût en K\$
Projet ERP				241 000
Management				5 400
		Management	LT-000	2 420
		Qualité	LT-005	870
		Coûtenance	LT-010	870
		Planification	LT-015	870
		Achats	LT-020	870
Provisions pour risques				5 400
		Provisions	LT-025	5 400
Logistique				2 300
		Assistance technique	LT-030	267
		Formation	LT-35	373
Conception globale				22 000
		Conception globale	LT-40	22 000
Intégration				44 000
		Intégration	LT-45	44 000

Les coûts des équipements définis lors de l'étude de faisabilité sont distribués sur le WBS, notamment sur les activités « lots de travaux » de type « réalisation » pour les équipements. Concernant les coûts de main-d'œuvre, le planning directeur et la matrice « RBS versus WBS » sont utilisés pour estimer la charge pour chaque lot de travaux qui, valorisée par les taux horaires des ressources affectées, fournit le budget de la main-d'œuvre. L'objectif est de réaliser 14 % de marge brute.

2. Synthèse de la démarche de structuration d'un projet IT

Pour résumer et pendant le cycle de vie de projet, les structures utilisées sont le PBS, le WBS, l'OBS, ainsi que les lignes budgétaires. Ces structures sont mises en relation par l'intermédiaire d'une matrice dont le cœur est le planning. Les coûts sont en effet imputés sur un lot de travail (WBS) et une ligne budgétaire, et d'autre part, les ressources (OBS) sont affectées à des tâches du planning. Concrètement, les structures sont mises en relation par l'intermédiaire du codage de leurs éléments. De même, les coûts et les délais peuvent être consolidés sur toutes les structures. Pour cela, les structures de gestion de projet s'imbriquent de la façon suivante (voir la figure)

Qui : (OBS : Organisation Breakdown Structure) : **Qui fait quoi**, qui est responsable de quoi, qui est responsable de qui ?

RBS (Resources Breakdown Structure) : C'est le « **Avec Quoi ?** »

WBS (Work Breakdown Structure) : Il représente le « **Comment** » du projet ?

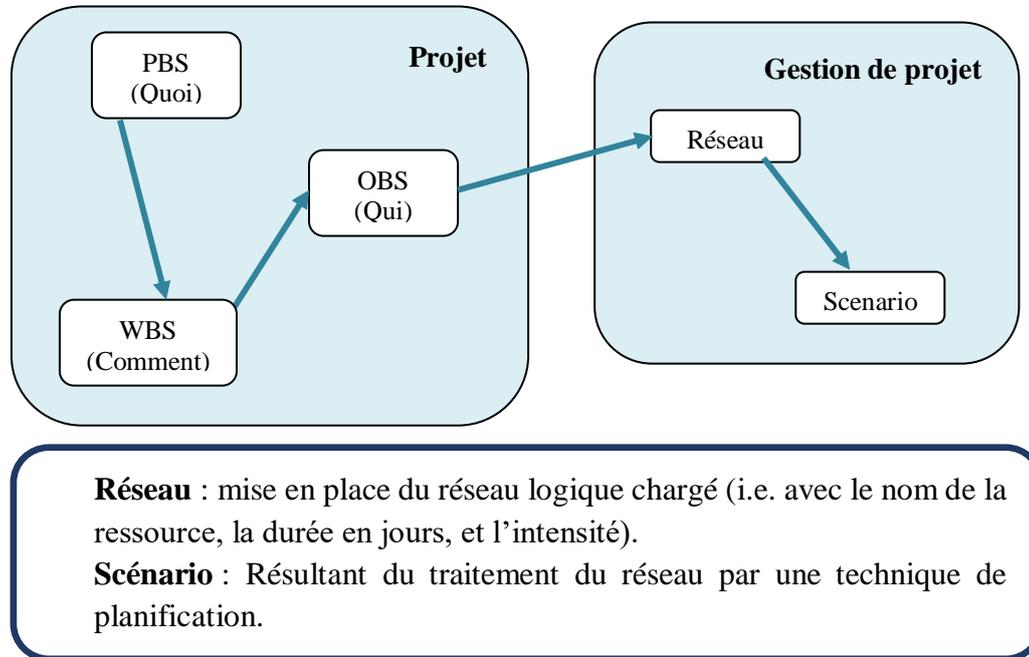


Figure. 17. Formalisme de structuration du projet



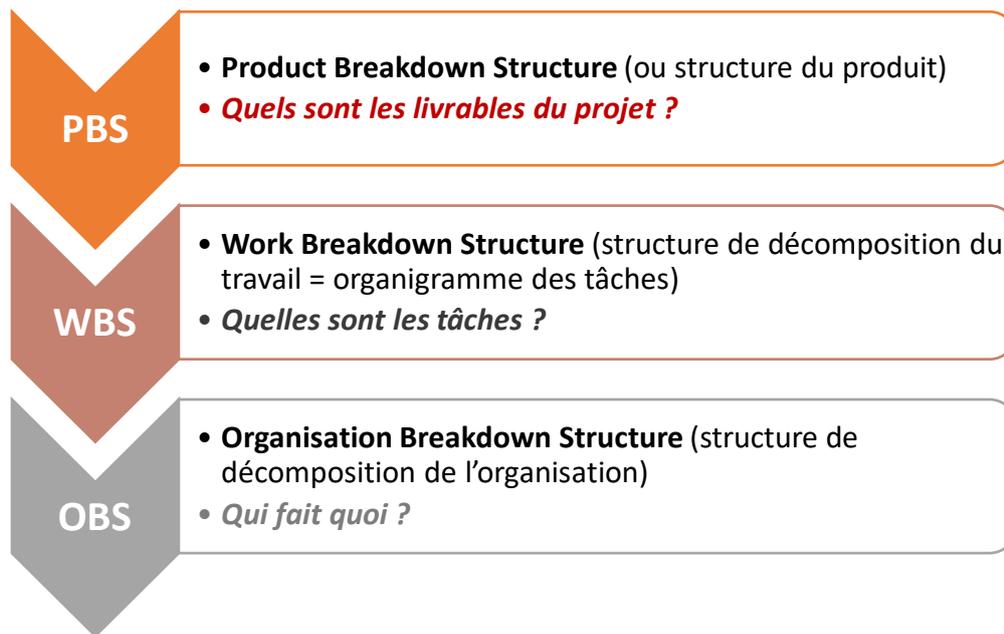


Figure. 18. Interaction entre le cahier de charge et Les normes internationales

Par ailleurs, on trouve d'autres structures arborescentes hiérarchiques qui composent le projet tels que :

1. Ressource Breakdown Structure (RBS)/ Organigramme des Ressources
2. Activity Breakdown Structure (ABS)
3. Geographical Breakdown Structure (GBS)

Exemple :

Question :

Q1. Pourquoi le découpage du projet ?

R1.

- Maîtriser la gestion de la production du produit final
- Répartir dans le temps la production et les ressources
- En réduisant la difficulté d'estimation « Plus facile d'estimer et organiser un tout en estimant une à une chacune des parties qui composent ce tout »
- Et en s'appuyant sur la prise en compte des liens entre les éléments constituant le « tout »

Q2. Comment construire le WBS de mon projet ?

Q3. Etablir le PBS (Product Breakdown Structure) de votre projet « site web ».

Q4. Qu'est-ce que WBS ?

R4. « Décomposition hiérarchique, axée sur les livrables, du travail que l'équipe de projet doit exécuter pour atteindre les objectifs du projet et produire les livrables voulus. » (Guide PMBOK).

WBS est une décomposition successive d'une activité plus grande (le projet lui-même) dans des activités plus petites.

La fiche de chacun des activités doit comporter les informations suivantes :

- un titre et une description de la tâche
- un responsable unique

- une durée d'exécution exprimée en jours ou en heures
- ne description des ressources nécessaires à son exécution
 - les ressources humaines
 - les ressources matérielles
- un coût estimé

Q5. À quoi ça sert le WBS ?

R5. Nous avons besoin de WBS afin de faire des estimations de coût et de travail (effort) à faire et de développer un calendrier consiste.

Estimer le coût :

- Estimer le coût de toutes les activités
- Inclure le coût des éléments dans le coût total du système

Performance du calendrier :

- Savoir quelles activités sont finies
- Mesurer le progrès

Q6. Comment créer un OBS ?

R6. Un OBS est créé beaucoup de la même manière comme un WBS.

- 1. Identifier la structure organisationnelle pour les ressources impliquées dans le projet
- 2. Une fois que la structure a été rempli, identifier tous les membres de l'équipe. Attribuez à chaque membre de l'équipe une position dans la structure.
- 3. Être sûrs que le OBS est structuré du département le plus responsable et ensuite par les départements d'exécution aux niveaux plus bas. le plus souvent avec le chef de projet au sommet, les chefs d'équipe de niveau 2, et les membres de l'équipe sous le niveau 2.

Q7. Quelles sont les règles qui devraient être prise en considération lors de la création d'un WBS ?

Les règles suivantes devraient être prises en considération lors de la création d'un WBS :

- Le niveau supérieur représente le projet.
- Les éléments du WBS n'ont pas nécessairement le même nombre de niveaux.
- Effectuer l'inventaire exhaustif des tâches à réaliser.
- Une codification des tâches peut être mise en place pour faciliter la lecture et l'identification de chaque tâche du projet.