INTRODUCTION À LA GESTION DE PROJETS INFORMATIQUES

Cours destiné aux baccalauréats en informatique – 2ième année

Philippe Gauthier
gauphil@gmail.com

Table des matières

Projets informatiques

1. Introduction

1.1. Le guide PMBOK

Il existe de nombreux guides de gestion de projets reconnus mondialement comme PRINCE 2, PMP, PMBOK… Dans le cadre de ce cours j'ai opté de vous expliquer l'approche de gestion de projet via le guide PMBOK. Le présent cours est une version résumée de ce guide, destinée à vous familiariser avec les grands concepts et termes utilisés dans la gestion de projet.

Le *Guide PMBOK®* (*Project Management Body Of Knowledge*), ou guide du corpus des connaissances en gestion de projet, est un standard mondial qui fournit les fondations pour construire une méthode solide de gestion de projet, indépendamment du domaine d'activité.
Il a été mis en place par le PMI (Project Management Institute), association fondée en 1969 qui compte aujourd'hui environ 700.000 membres répartis dans 222 pays (chiffre de fin 2023).

1.2. Pourquoi la gestion de projet ?

1.2.1. Trop de projets échouent :

Le '*Standish group*' a publié annuellement de 1994 à 2020 le rapport CHAOS. Celui-ci présentait un instantané de l'art du développement logiciel.
(référence : https://www.infoq.com/fr/articles/rapport-chaos-2015)

Vous trouverez des exemples des résultats des analyses effectuées dans le tableau ci-dessous.

Résultats globaux pour un échantillonnage de projets informatiques (+/- 50.000 pour 2015)

MODERN RESOLUTION FOR ALL PROJECTS

	2011	2012	2013	2014	2015
SUCCESSFUL	29%	27%	31%	28%	29%
CHALLENGED	49%	56%	50%	55%	52%
FAILED	22%	17%	19%	17%	19%

The Modern Resolution (OnTime, OnBudget, with a satisfactory result) of all software projects from FY2011-2015 within the new CHAOS database. Please note that for the rest of this report CHAOS Resolution will refer to the Modern Resolution definition not the Traditional Resolution definition.

Les résultats du rapport CHAOS en tenant compte de la taille des projets

CHAOS RESOLUTION BY PROJECT SIZE

	SUCCESSFUL	CHALLENGED	FAILED
Grand	2%	7%	17%
Large	6%	17%	24%
Medium	9%	26%	31%
Moderate	21%	32%	17%
Small	62%	16%	11%
TOTAL	**100%**	**100%**	**100%**

The resolution of all software projects by size from FY2011–2015 within the new CHAOS database.

Les résultats du rapport CHAOS en tenant compte de la méthode utilisée : 'Waterfall' versus 'Agile'

CHAOS RESOLUTION BY AGILE VERSUS WATERFALL

SIZE	METHOD	SUCCESSFUL	CHALLENGED	FAILED
All Size Projects	Agile	39%	52%	9%
	Waterfall	11%	60%	29%
Large Size Projects	Agile	18%	59%	23%
	Waterfall	3%	55%	42%
Medium Size Projects	Agile	27%	62%	11%
	Waterfall	7%	68%	25%
Small Size Projects	Agile	58%	38%	4%
	Waterfall	44%	45%	11%

The resolution of all software projects from FY2011–2015 within the new CHAOS database, segmented by the agile process and waterfall method. The total number of software projects is over 10,000.

1.2.2. Quelles sont les principales raisons d'échecs ?

Une part importante de l'analyse du *Standish Group* a été l'identification et le classement des facteurs clés de réussite d'un projet. Les résultats présentent la liste ordonnée de facteurs suivants :

CHAOS FACTORS OF SUCCESS

FACTORS OF SUCCESS	POINTS	INVESTMENT
Executive Sponsorship	15	15%
Emotional Maturity	15	15%
User Involvement	15	15%
Optimization	15	15%
Skilled Resources	10	10%
Standard Architecture	8	8%
Agile Process	7	7%
Modest Execution	6	6%
Project Management Expertise	5	5%
Clear Business Objectives	4	4%

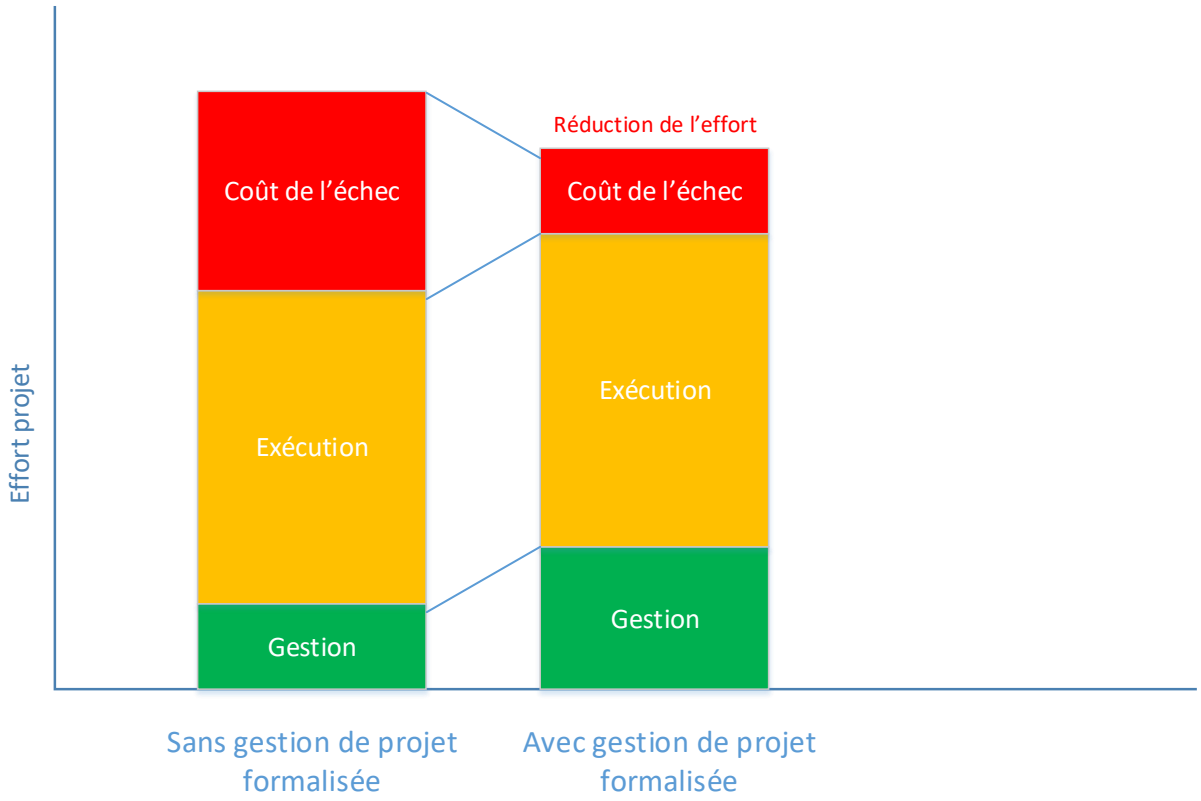
Quelques mots d'explications :

Executive Support *(Sponsorship)* implique qu'un dirigeant ou groupe de dirigeants accepte de porter financièrement et émotionnellement le projet. Le ou les dirigeants encourage(nt) et accompagne(nt) l'accomplissement du projet.
Emotional Maturity (Maturité émotionnelle) est l'ensemble des comportements permettant de travailler en équipe.
User Involvement (Engagement de l'utilisateur) a lieu quand les utilisateurs sont engagés dans les processus de décisions et de collectes d'informations du projet. Cela passe également par du feedback, des revues de demandes, de la recherche classique, du prototypage, et autres outils de construction de consensus.
Optimization est une manière structurée d'améliorer l'efficacité métier et d'optimiser un ensemble de petits projets ou de demandes importantes. L'optimisation commence avec la gestion du périmètre du projet.
Skilled Staff (Personnel compétent) sont les personnes comprenant à la fois le métier et la technologie. La compétence donne une forte capacité dans la réalisation des demandes du projet et la livraison du projet ou produit.

SAME est le "*Standard Architectural Management Environment*". Le *Standish Group* définit le SAME comme un groupe cohérent et intégré de pratiques, services et de produits permettant de développer, implémenter et opérer une application.
Agile Proficiency : signifie que l'équipe et le *product owner* maîtrisent l'agilité. Obtenir de bons ou de mauvais résultats en utilisant la méthodologie AGILE dépend en effet du niveau des connaissances de celle-ci par l'ensemble des parties.
Modest Execution (Réalisation frugale) implique d'avoir une réalisation sobre en limitant le nombre de fonctionnalités à ce qui est strictement nécessaire (pas d'usine à gaz !).
Project Management Expertise est l'usage de connaissances, compétences et techniques aux activités de projet pour atteindre ou dépasser les attentes des parties prenantes et produire de la valeur pour l'organisation.
Clear Business Objectives (Objectifs métier clairs) est la compréhension des attentes par toutes les parties prenantes et tous les participants dans le projet. Des objectifs métiers clairs peuvent aussi signifier un alignement sur les enjeux et la stratégie de l'entreprise.

1.2.3. Valeur ajoutée de la gestion de projet

Objectif : réduire d'une façon significative l'effort total du projet et son coût



Valeur ajoutée de la gestion de projet :
La gestion de projet permet à l'organisation de réaliser ses projets, en respectant le budget, le délai de livraison et les besoins des utilisateurs.

L'implémentation des meilleures pratiques en gestion de projets apporte un plus au succès des projets, par la réduction de l'effort total du projet et surtout des coûts entraînés par le '*rework*' (recommencer le travail).

Exemple :

- Le chef de projet définit mal ce qu'il attend des membres de l'équipe et le travail est à recommencer → Effort supplémentaire important peut en résulter !

2. Concepts et définitions

2.1. Qu'est-ce qu'un projet ?

« Un projet est une entreprise temporaire exercée dans le but de créer un produit, un service ou un résultat unique » (définition du *Guide PMBOK*®).

Il ne doit y avoir aucune ambiguïté quant aux termes utilisés. :
- "**Une entreprise...**" représente un travail, un engagement et un effort.
- **Temporaire** » signifie qu'un projet a un début et une fin.
- " **Exercé dans le but de créer** " car le projet est un ensemble d'objectifs à atteindre.
- "**un produit nouveau et unique, ou un service nouveau et unique, ou encore un résultat nouveau et unique** ".

Exemples de projets :
- Création d'un nouveau modèle d'automobile.
- Création d'une nouvelle tablette informatique.
- Création d'un nouveau logiciel.
- Développement d'un nouveau produit ou service.
- Changement dans la structure d'une organisation.

Projets ou activités ?

- Acheter une automobile n'est pas un projet dans la mesure où le produit n'est pas unique (on choisit la couleur, le moteur et des options), c'est simplement un achat plus important qu'un autre.
- Acheter un Progiciel de Gestion Intégrée (ex : ERP pour *Enterprise Resource Planning*) est en général un projet car l'entreprise achète un produit qui devient unique du fait de la configuration, des paramétrages et des interfaces à mettre en œuvre. De plus, l'élaboration du produit sera progressive et la conduite du changement très importante.
- Un travail personnel comme la réalisation d'un mémoire n'est pas véritablement un projet : étant seul à mener ce travail, à la fois fournisseur et client, on ne mettra pas en œuvre un management de projet. On fera sa liste de travaux et on suivra son plan personnel.

2.2. Qu’est-ce qu’un projet informatique ?

Un projet informatique peut être :
- La création d’un nouveau logiciel.
- La création d’un nouvel espace numérique.
- La création d’une nouvelle infrastructure informatique ou l’évolution majeure d’une infrastructure existante.
- La création d’un service de support.
- L’évolution majeure d’un logiciel.
- L’intégration de logiciels existants.
- …

Un projet informatique est généralement plus complexe et plus difficile à maîtriser que d’autres projets, comme une construction immobilière ou la conception d’un nouveau modèle de cafetière.

Cela est dû à plusieurs facteurs : le produit logiciel est intangible (que l'on ne peut toucher, qui doit rester intact) et « intelligent ». L’écriture du code source par des développeurs est un processus d’interprétation des exigences et souvent d’apprentissage. La connaissance et la maîtrise du produit s’apprennent en même temps que la réalisation. Un logiciel peut présenter un très grand nombre de cas d’utilisation, de valeurs des données et de combinaisons de données qui rendent l’exhaustivité des tests impossible.

Le client peut avoir de grandes difficultés à définir toutes les exigences du projet, et le travail de conception du produit peut demander de nombreux échanges, des itérations, un ou des prototypes, et des livraisons par composant.

Les caractéristiques d’un projet informatique sont aussi influencées par les nombreux facteurs technologiques :
- Les systèmes d’exploitation.
- Les protocoles de communication.
- L’architecture technique.
- L’utilisation d’outils de développement.
- L’intégration de briques logicielles.
- Les interfaces (API, Web Services…)
- Etc…

Ces facteurs vont apporter des contraintes et des risques. Ils participent à l’environnement du projet, qui est complexe et changeant, ils contribuent à une incertitude quant au résultat, sans oublier les critères de qualité du produit logiciel comme, par exemple :
- La sécurité
- La fiabilité
- Le temps de réponse
- L’ergonomie
- La capacité de maintenance
- La disponibilité
- L’exploitabilité
- Etc…

2.3. Projet vs processus

Les projets et les processus vont de pair.

Un processus est une combinaison d'activités qui sont exécutées de façon répétitive et qui mènent à un résultat spécifique.

Similarités entre un projet et un processus :
- Production d'un résultat (output)
- Utilisation de ressources
- Début et fin

Différences entre un projet et un processus :
- Caractère unique du produit final
- Nature répétitive

Lors de l'exécution d'un projet beaucoup de processus sont exécutés comme par exemple :
- Faire un plan de projet
- Modifier le plan de projet
- Faire un rapport d'état d'avancement
- Corriger des erreurs…

Un plan de projet contient tous les processus nécessaires pour atteindre les objectifs du projet (processus 'planifiés').

2.4. Exercice : Projet ou Processus ?

a. Construire une villa ?
b. Remplir un formulaire de fax ?
c. Assembler un meuble IKEA ?
d. Concevoir une nouvelle voiture ?
e. Assembler une voiture ?
f. Installer un nouveau pc ?
g. Développer une nouvelle release (version) produit ?
h. Répondre à un appel à l'helpdesk ?
i. Etablir une amélioration de performance dans une organisation ?

2.5. Elaboration progressive

Un projet est, par définition, un produit unique, ce qui signifie que ses caractéristiques doivent être élaborées progressivement. Au départ nous n'avons que peu de données à notre disposition et donc le degré d'incertitude sera grand (au niveau des prévisions pour le planning, les ressources nécessaires, les budgets…). Ce degré d'incertitude diminuera en fonction des informations complémentaires que nous obtiendrons au fil du temps, en procédant par étapes (itérations – affinage du plan de projet).

2.6. Triple contrainte

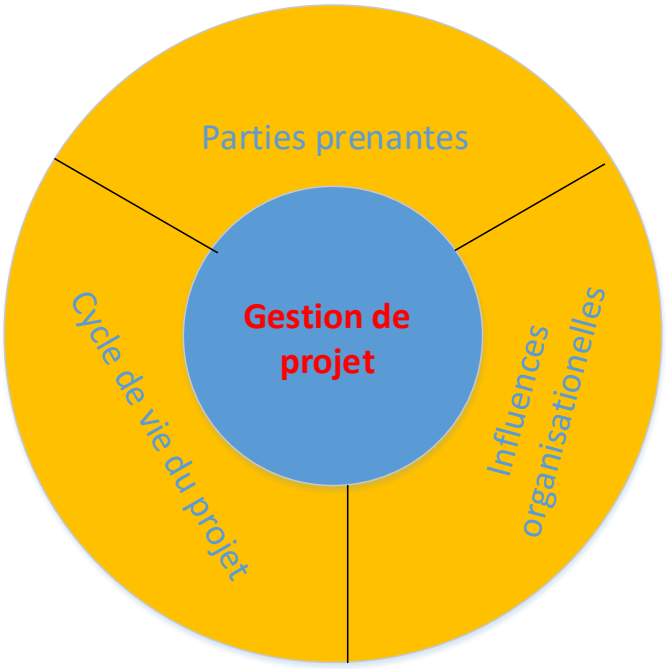
Le chef de projet doit maintenir l'équilibre entre les objectifs de **temps, coûts et contenu** (contenu/qualité). Il est très important de définir quel sera l'ordre de priorité entre ces trois contraintes. C'est une opération complexe !

Exemples :

- Ajouter des ressources au projet va quasi-certainement réduire la durée du projet mais va également augmenter le coût du projet et probablement impacter négativement la qualité du produit. En effet, quatre personnes travaillant sur un projet durant 4 semaines ne feront probablement pas ce que 2 personnes sont capables de faire en 8 semaines.
- Une demande de qualité supérieure induira forcément des coûts plus élevés et une durée plus longue.

2.7. Cycle de vie et organisation d'un projet

Le contexte de la gestion de projet

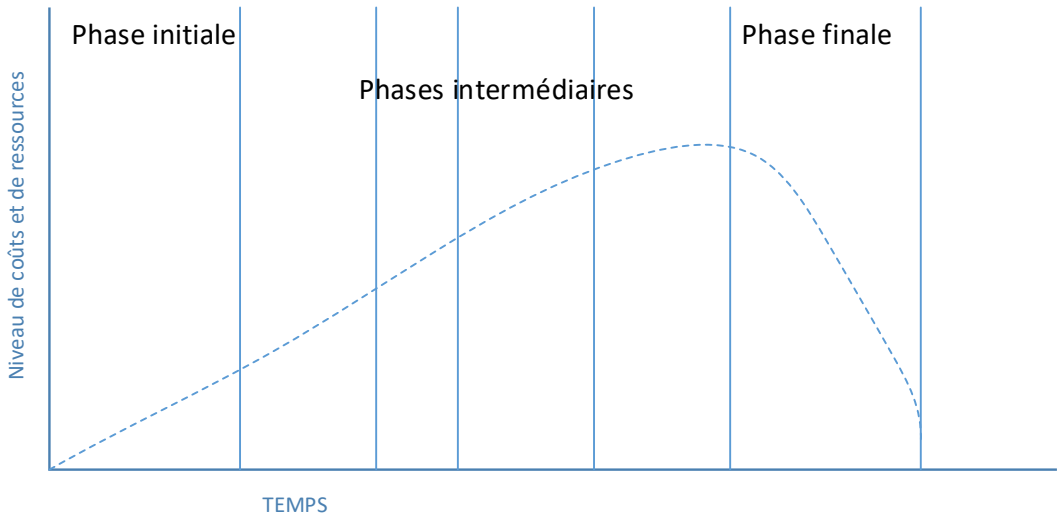


Ci-dessus les trois éléments principaux qui peuvent influencer la gestion de projet :

- Le cycle de vie du projet
- Les parties prenantes du projet
- Les influences organisationnelles

2.7.1. Le cycle de vie du projet :

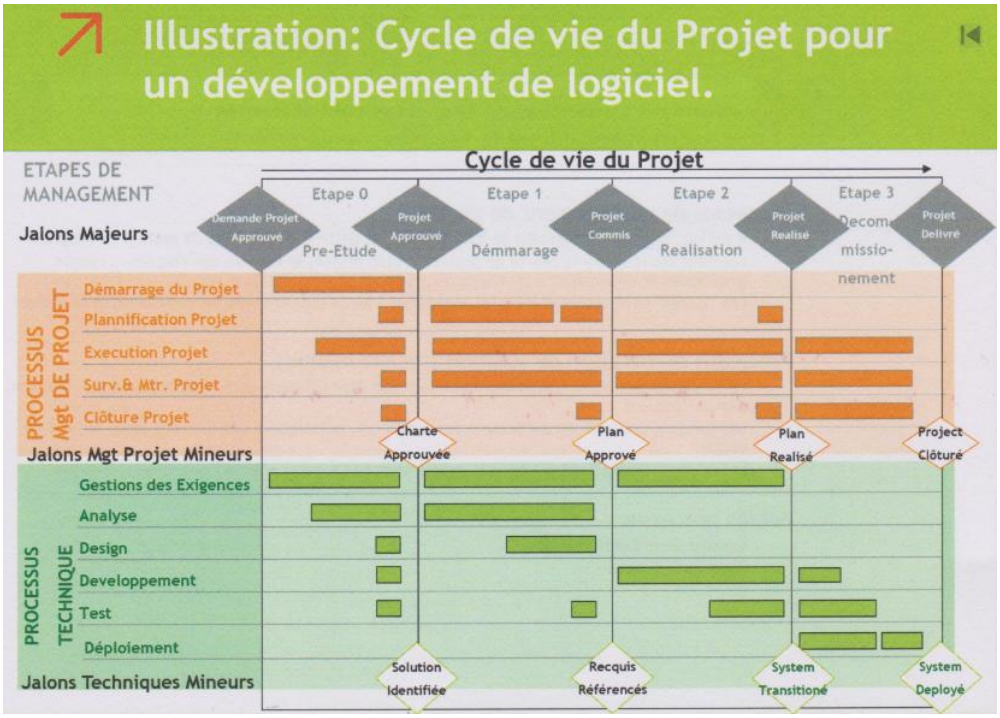
- Le cycle de vie du projet est un contrôle gestionnaire qui a comme objectif de grouper le travail projet en phases séquentielles ou en différentes étapes.
    - Il clarifie là où le projet commence et où il se termine.
    - Il définit le travail qui devra être effectué dans chaque phase.
    - Une phase se termine généralement par un ou plusieurs 'livrable(s)' (*deliverable(s)*) qui devra(ont) être acceptés/validés avant d'entamer la phase suivante.
    - Différents ensembles de processus seront exécutés simultanément durant le cycle de vie du projet :
        - Processus de gestion de projet
            - Il définit, organise, surveille le travail projet et délivre les livrables de gestion de projet
            - Les livrables (comme la charte de projet, le plan de projet, le registre des risques…) sont génériques et applicables à tous types de projet
        - Processus de livraison de solution (technologique)
            - Ce processus définit le travail technique qui doit être effectué dans chaque phase (analyse, design, développements, tests…)
            - Les livrables (comme les documents des exigences, les prototypes…) qui seront différent selon le type de projet
        - Autres processus
            - Assurance qualité
            - Gestion de la configuration
            - …



La somme de toutes les phases représente le cycle de vie du projet.

Remarque : Dans le cas d'un grand projet, il peut arriver que l'on soit amené à le diviser en sous-projets. En ce cas, chacun d'eux sera traité comme un projet à part entière.

Illustration pour un projet informatique :



L'image, ci-dessus, est un exemple de cycle de vie typique pour un développement de logiciel :

- Etapes de management : contrôle gestionnaire global en définissant des '*deliverables*' (jalons) majeurs sur lesquels des décisions devraient être prises concernant la continuation ou le réajustement du projet
- Les processus de management du projet sont définis pour initier, planifier, surveiller et contrôler le projet. Les 'deliverables' du management de projet indiquent les réalisations majeures en terme du management de projet (ex : plan approuvé)
- Les processus d'engineering sont définis pour produire un produit final. Ils sont orientés techniques.

2.7.2. Les parties prenantes du projet :

Les parties prenantes sont les personnes ou un groupe de personnes :

- Qui sont activement impliquées dans le projet
- Qui ont un intérêt dans l'exécution et l'achèvement du produit final

Exemple de parties prenantes :

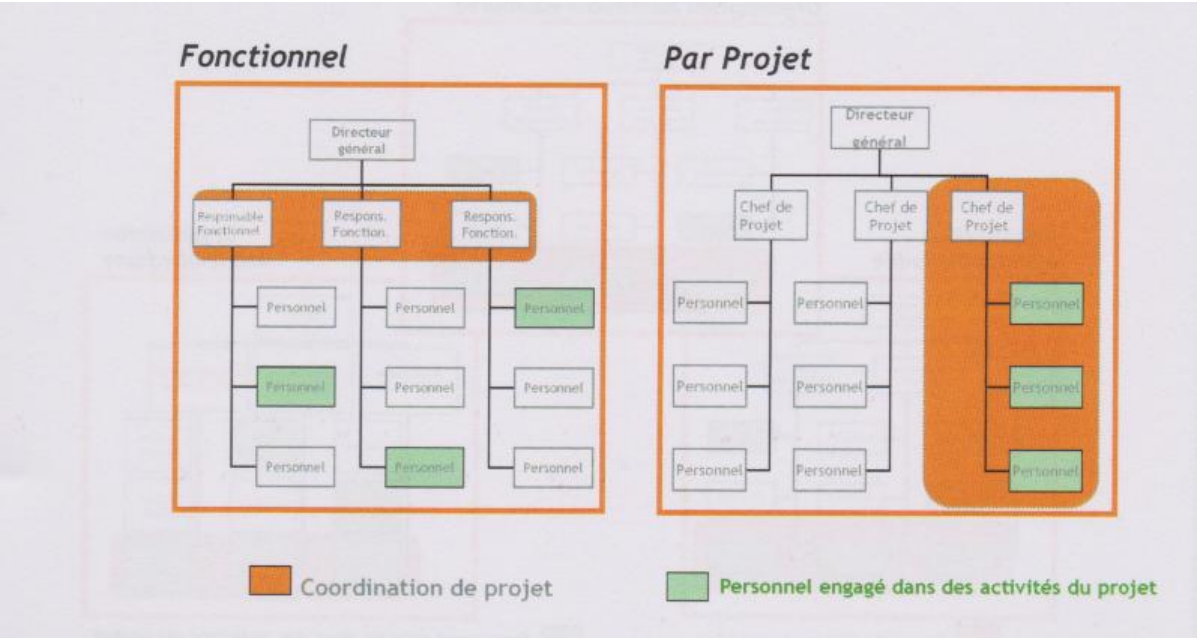
- Le client : personne ou organisation qui utilisera le produit. Le projet ne sera un succès que si les besoins de tous les utilisateurs ont été pris en compte
- Le commanditaire (sponsor) : la personne ou le groupe qui finance le projet
- L'entreprise réalisatrice : l'entreprise dont les employés sont le plus directement impliqués dans l'exécution du travail du projet
- Les membres de l'équipe projet : le groupe qui effectue le travail du projet.

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 !

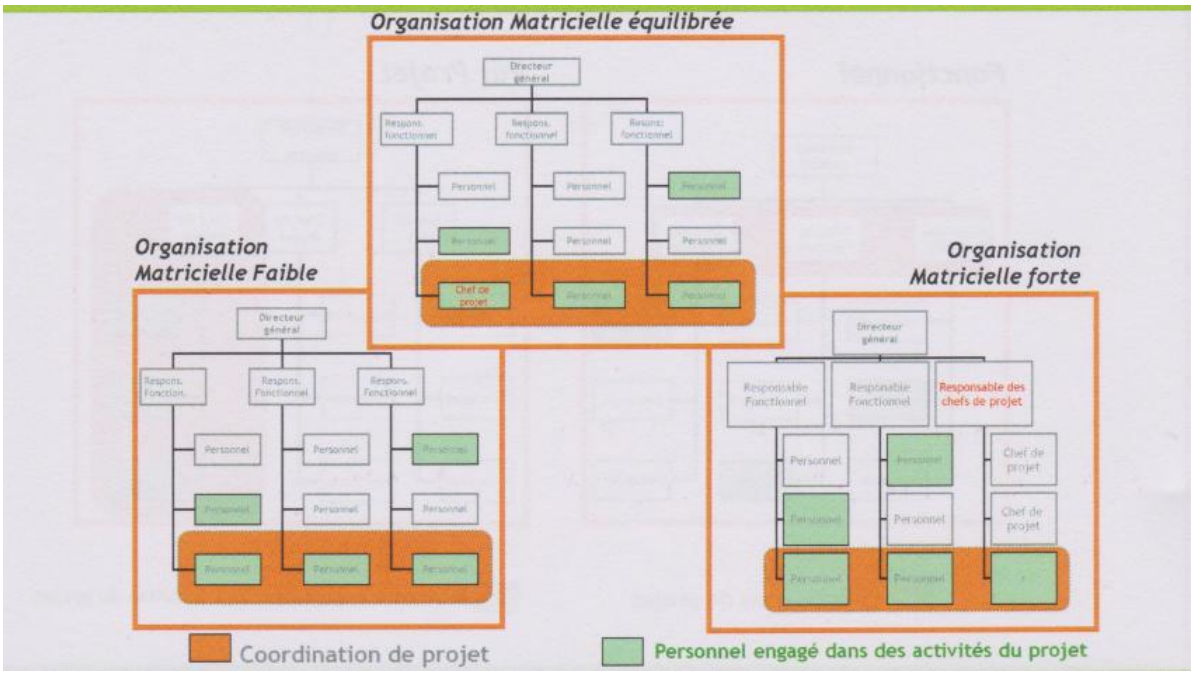
2.7.3. Les influences organisationnelles

L'organisation dans laquelle le projet sera réalisé exercera également une influence importante sur la réalisation du projet. On parle ici d'organisations fonctionnelles, par projet ou matricielles. Ce qu'il faut retenir :

- Dans une organisation fonctionnelle c'est le responsable du département qui conserve l'autorité sur les membres de leur équipe
- Dans une organisation par projets, c'est le chef de projet qui a autorité sur les membres de l'équipe affectés au projet en question



- Dans une organisation matricielle l'autorité pourra être partagée entre le chef de projet et le chef fonctionnel suivant le type de matrice utilisée (faible, équilibrée ou forte)

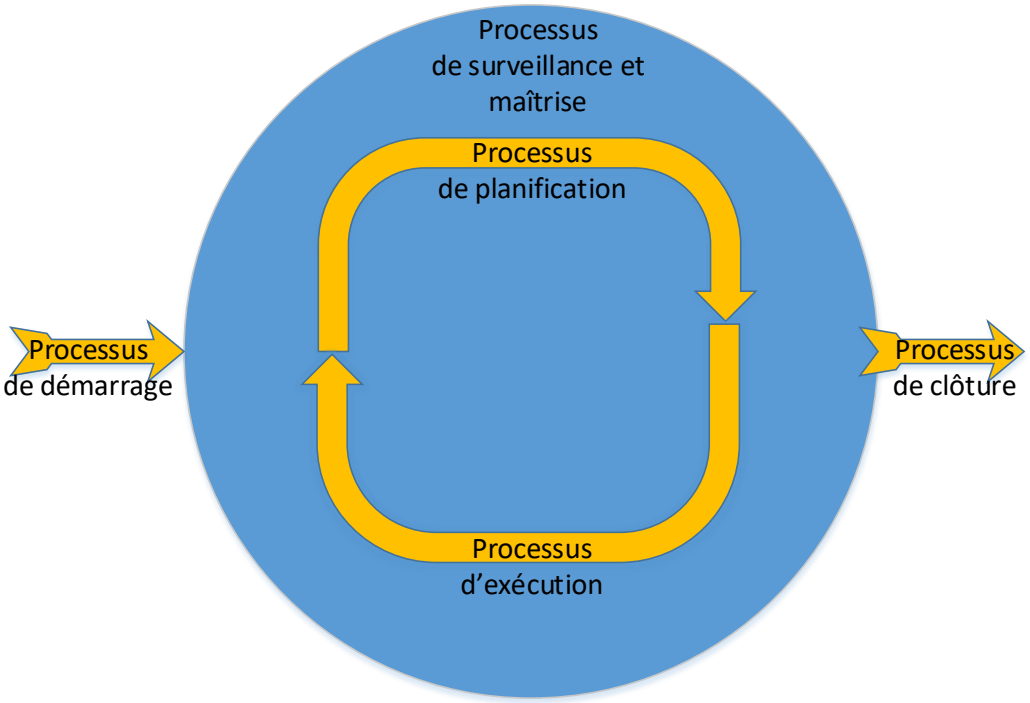


- Organisation matricielle faible : va une étape plus loin qu'une organisation fonctionnelle, en ayant la coordination du projet exécutée par une équipe de représentants de chaque département. Le rôle du chef de projet se limite donc à la coordination de l'effort des ressources sans autorité formelle sur celle-ci
- Organisation matricielle équilibrée : Même base que ci-dessus (faible) mais un chef de projet est assigné au sein de l'équipe. Il sera responsable pour la définition des besoins qui devront être accomplis, tandis que les managers en ligne seront concernés par qui accomplira la tâche.
- Organisation matricielle forte : a un département de chefs de projet. Un chef de projet est assigné à un projet. Il mènera dans certains cas, une équipe de membres d'autres départements. En ce cas il a l'autorité requise pour prendre toutes les décisions

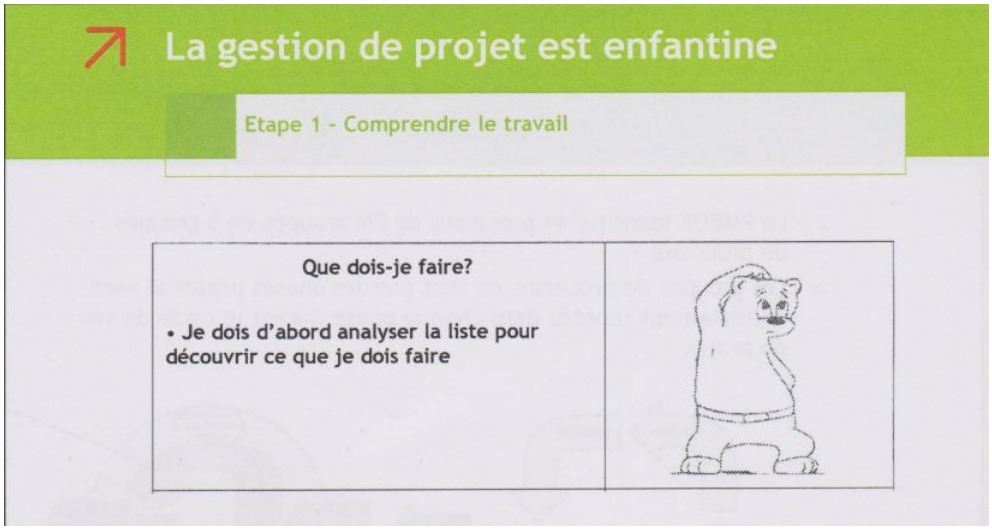
2.8. Processus de management du projet

2.8.1. Groupes de processus de management de projet

- PMBOK identifie 48 processus de gestion de projet groupés en 5 groupes de processus
- Ces groupes ne sont pas des phases du projet et sont répétés dans chaque phase durant le cycle de vie du projet



2.8.2. Illustration simple



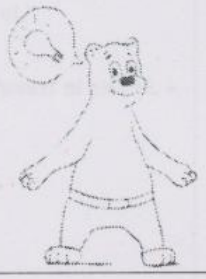
La gestion de projet est enfantine

Etape 2 - Planifier le travail

Comment vais-je le faire?

"Je ferai un plan"

- Je prend le matériel nécessaire
- Je pense aux règles



La gestion de projet est enfantine

Etape 3 - Effectuer le travail

"Je fais mon travail"

- Je fais le travail

La gestion de projet est enfantine

Etape 4 - Contrôler le travail et tirer les leçons

"Je contrôle : ai-je bien fait le travail?"

- Je le note dans mon carnet
- Je recommence si nécessaire

2.8.3. But des 5 groupes de processus en résumé

1. **Processus de démarrage** : définir qui sont les parties prenantes et quelles sont leurs attentes (output=’Charte de projet)
2. **Processus de planification** : définir le contenu total de l’effort, définir et affiner les objectifs, préciser la suite des actions nécessaires à l’atteinte des objectifs (output=’Plan de projet’ qui est un plan détaillé qui s’affinera au fil du temps)
3. **Processus d’exécution** : guider l’équipe dans l’accomplissement du travail défini dans le plan de management du projet pour respecter les spécifications du projet
4. **Processus de surveillance et maîtrise** : suivre, revoir et réguler l’avancement et la performance du projet, identifier les parties dans lesquelles des modifications du plan s’avèrent nécessaires, et entreprendre les modifications correspondantes
5. **Processus de clôture** : permettent de terminer le projet de façon ordonnée

Ces groupes de processus sont valables pour les projets, sous-projets ou phases d’un projet.

2.8.4. Les 9 domaines de connaissances en management de projet

- Management de l’intégration du projet : décrit les processus et activités qui intègrent les divers éléments de management de projet
- Management du contenu du projet : décrit les processus nécessaires pour garantir que le projet comprend tous les travaux nécessaires à sa réalisation et uniquement ces travaux.
- Management des délais du projet : décrit les processus nécessaires pour assurer la réalisation du projet en temps voulu.
- Management des coûts du projet : décrit les processus de planification, d’estimations, de budgétisation et de maîtrises des coûts nécessaires pour s’assurer que le projet soit réalisé en respectant le budget approuvé.
- Management de la qualité du projet : décrit les processus nécessaires pour s’assurer que le projet réponde aux besoins définis au départ.
- Management des ressources humaines du projet : décrit les processus nécessaires pour organiser et diriger l’équipe du projet.
- Management des communications du projet : décrit les processus nécessaires pour assurer, en temps voulu et de façon appropriée, la génération, la collecte, la diffusion, le stockage et le traitement final des informations du projet.
- Management des risques du projet : décrit les processus liés au management des risques dans le cadre d’un projet.
- Managements des approvisionnements du projet : décrit les processus nécessaires à l’achat ou à l’acquisition de produits, services ou de résultats, ainsi que les processus de management des contrats.
- Management des parties prenantes du projet

Nous retrouvons à la page suivante les 48 processus de management de projet du guide PMBOK regroupés par groupe de processus et par domaine de connaissances.

2.8.5. Domaines de connaissances – Matrice du groupe de processus

Domaines de connaissance	Groupes de processus de management de projet				
	Groupe de processus d'initialisation	Groupe de processus de planification	Groupe de processus d'exécution	Groupe de processus de maîtrise	Groupe de processus de clôture
4. Gestion de l'intégration du projet	4.1 Élaborer la charte du projet	4.2 Élaborer le plan de management du projet	4.3 Diriger et gérer le travail du projet 4.4 Gérer les connaissances du projet	4.5 Maîtriser le projet 4.6 Maîtriser les changements	4.7 Clore le projet ou la phase
5. Gestion du périmètre du projet		5.1 Planifier la gestion du périmètre et du contenu 5.2 Recueillir les exigences 5.3 Définir le périmètre 5.4 Créer le WBS		5.5 Valider le périmètre 5.6 Maîtriser le périmètre et le contenu	
6. Gestion de l'échéancier du projet		6.1 Planifier la gestion de l'échéancier 6.2 Définir les activités 6.3 Organiser les activités en séquence 6.4 Estimer la durée des activités 6.5 Élaborer l'échéancier		6.6 Maîtriser l'échéancier	
7. Gestion des coûts du projet		7.1 Planifier la gestion des coûts 7.2 Estimer les coûts 7.3 Déterminer le budget		7.4 Maîtriser les coûts	
8. Gestion de la qualité du projet		8.1 Planifier la gestion de la qualité	8.2 Gérer la qualité	8.3 Maîtriser la qualité	
9. Gestion des ressources du projet		9.1 Planifier la gestion des ressources 9.2 Estimer les ressources nécessaires aux activités	9.3 Obtenir les ressources 9.4 Développer l'équipe 9.5 Gérer l'équipe	9.6 Maîtriser les ressources	
10. Gestion des communications du projet		10.1 Planifier la gestion des communications	10.2 Gérer les communications	10.3 Maîtriser les communications	
11. Gestion des risques du projet		11.1 Planifier la gestion des risques 11.2 Identifier les risques 11.3 Mettre en œuvre l'analyse qualitative des risques 11.4 Mettre en œuvre l'analyse quantitative des risques 11.5 Planifier les réponses aux risques	11.6 Appliquer les réponses aux risques	11.7 Maîtriser les risques	
12. Gestion des approvisionnements du projet		12.1 Planifier la gestion des approvisionnements	12.2 Procéder aux approvisionnements	12.3 Maîtriser les approvisionnements	
13. Gestion des parties prenantes du projet	13.1 Identifier les parties prenantes	13.2 Planifier l'engagement des parties prenantes	13.3 Gérer l'engagement des parties prenantes	13.4 Maîtriser l'engagement des parties prenantes	

3. Projet et méthodologie

Un projet peut être issu d'origines diverses comme, par exemple :

- Un plan opérationnel d'entreprise qui est lui-même lié sur le plan stratégique d'entreprise
- Un besoin ponctuel pour un département, un service (ex : logiciel)
- Une obligation légale (ex : Mise en conformité GDPR, NIS2)
- ….

Quel que soit l'origine du projet, le chemin pour sa concrétisation sera toujours identique au sein des entreprises qui ont un certain niveau de maturité dans la gestion de projet. Leur méthodologie est en effet basée sur les bonnes pratiques d'un ou plusieurs standards (PMBOK, PMI, PRINCE 2…). **Un standard de gestion de projet n'est pas à suivre à la lettre, chaque entreprise peut l'adapter à son fonctionnement interne**. Toutes les étapes décrites ne sont donc pas obligatoires, ceci dépendra surtout du niveau de complexité de votre projet.

4. Le démarrage du projet

But du démarrage du projet :

- S'assurer que le chef de projet et le Comité de Pilotage (=COPIL) aient les mêmes attentes en ce qui concerne le résultat du projet, le copil représentant les intérêts de l'ensemble des parties prenantes
- Des attentes ont été entendues et répertoriées mais elles peuvent être :
  - Peu claires, peu réalistes, pas bien comprises, non exprimées ou conflictuelles
- Le chef de projet aura pour mission de s'assurer que le projet ait une base de démarrage stable

Processus de démarrage du projet :

En nous référant au tableau des groupes de processus ci-dessus, nous retrouvons les deux processus suivants à exécuter lors de la phase de démarrage du projet :

- 4.1. Elaborer la charte de projet
- 10.1. Identifier les parties prenantes

4.1. La feuille de route du projet – Processus de démarrage

La feuille de route du projet est une représentation schématique que nous utiliserons pour indiquer les principaux éléments intervenants dans la gestion de projet.

Nous la parcourons à deux reprises :

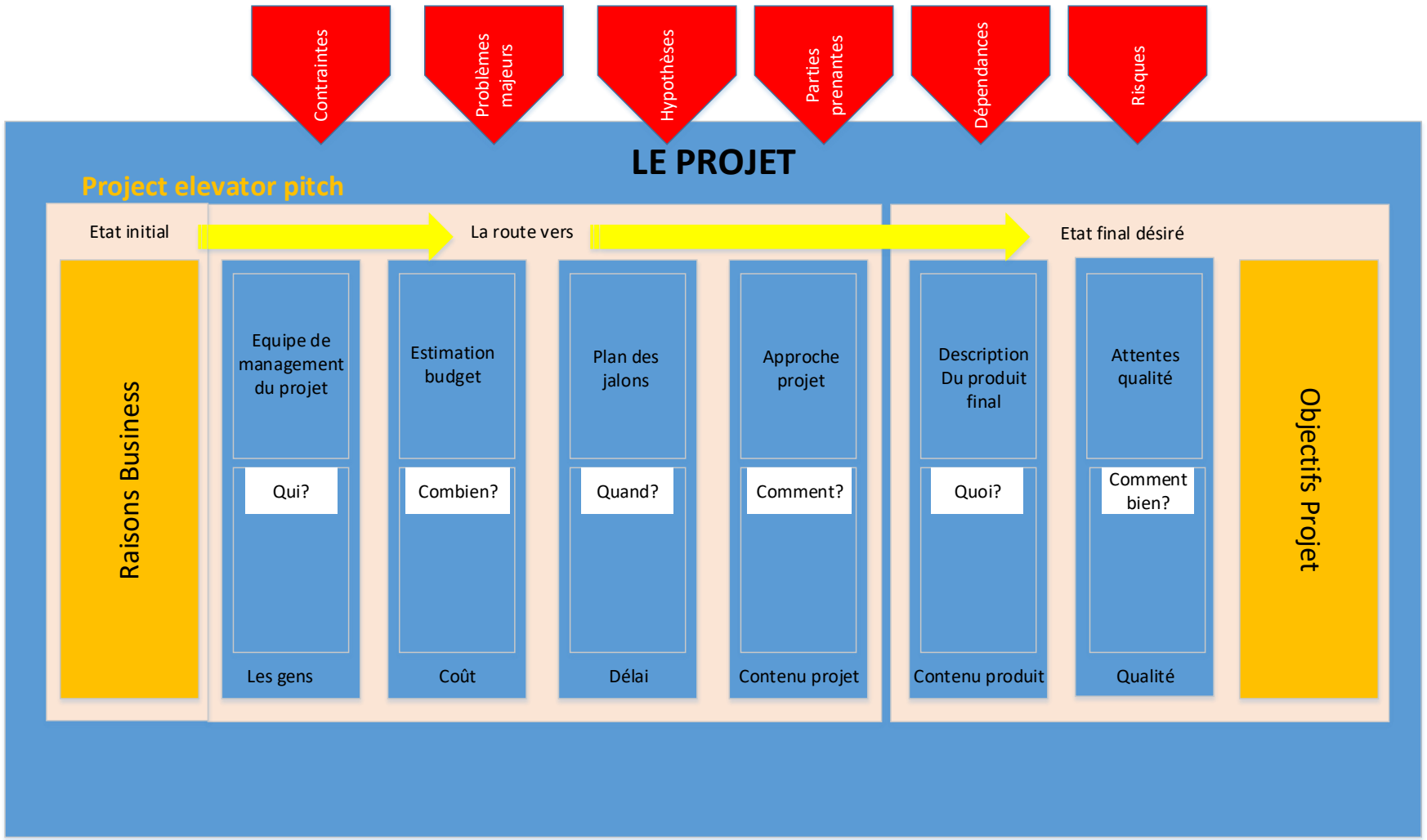
1. Lors de la phase de démarrage du projet
2. Lors de la phase de planification

Le groupe de processus de démarrage du projet a deux finalités (ou livra*bles*) :

- Le premier est la production de la **charte de projet**, aussi appelée ‘mandat de projet’ ou ‘contrat de projet’. Il s’agit d’un document qui, une fois finalisé et avalisé par le COPIL, officialisera le ‘GO’ pour exécuter le projet ou une phase du projet. Nous expliquons plus loin dans ce syllabus comment établir ce document et les détails qu’il peut contenir. Pour rappel, les modèles proposés sont des exemples. Les entreprises sont libres d’adapter ou modifier ces documents afin de les adapter à leur mode de fonctionnement.

- Le second est l’**identification des parties prenantes** du projet. Cette notion de partie prenante est expliquée plus loin dans ce syllabus et réapparaîtra à plusieurs reprises dans les différentes phases du projet.

La feuille de route du projet



4.2. Raison Business

La question qui se pose est :

‘’ Pourquoi allons-nous faire ce projet ? ‘’

Elle doit s’exprimer en termes de besoin / opportunités et non en termes de solution.
Par exemple,

- Il ne faut pas dire :
  - "Nous désirons un système intégré de gestion de ressources"
- mais bien :
  - "Nous voulons réduire les frais généraux en ce qui concerne le temps et coût investis dans la planification des ressources et la facturation".

Autres exemples d’argumentations :

- Augmentation de la productivité / performance
- Amélioration des marges
- Être conforme avec une nouvelle législation
- Réduction du temps de livraison
- Accroître la satisfaction client
- …

4.3. Objectif(s) du projet

Les objectifs du projet doivent à ce stade être définis clairement.
Ils représentent le résultat final attendu et nous devons exprimer, dans la phase de démarrage du projet, un résumé du résultat désiré du projet qui doit être SMART (dans la mesure du possible).

S.M.A.R.T. – explications :

S.M.A.R.T. est un acronyme simple à retenir qui permet de n’oublier aucun critère pour la définition d’un objectif de gestion de projet :

S = Spécifique/ Simple : l’objectif doit être spécifique, c’est-à-dire clair, simple, précis et adapté à la situation. Il doit désigner l’objet du projet avec le plus d’exactitude possible. Il doit être compris par tous les participants.

Exemple **:**

- Je veux augmenter mon chiffre d’affaires de 5% par rapport au chiffre d’affaires de l'année précédente
- Je veux augmenter le nombre de tickets résolus par mon équipe technique de 2 par technicien par jour

M = Mesurable : Votre objectif doit être aussi mesurable. En effet, vous devez posséder les outils qui vous permettront d'analyser vos résultats.

La mesure de l'atteinte de l'objectif peut se présenter sous forme de valeurs, de seuils à atteindre, d'indicateurs d'évaluation. Ces indicateurs permettront aussi de suivre l'avancement du projet, ce qui aidera à motiver les équipes tout au long du projet.

Les deux exemples repris ci-dessus permettent une mesure d'atteinte de l'objectif puisqu'ils font référence à des chiffres précis du passé

A = Acceptable : La lettre A est parfois assimilée à d'autres mots comme atteignable, ambitieux ou approprié. Dans tous les cas, cela signifie que suffisamment ambitieux pour être challengeant et pour motiver l'équipe. Il doit cependant rester faisable pour éviter de vous décourager en cours de route.

Exemple :

- si augmenter son chiffre d'affaire de 5% est acceptable, avoir l'ambition de l'augmenter de 80% semble, sauf circonstances exceptionnelles, irréalisable et risque, plus que probablement, de démotiver les troupes avant de commencer

R = Réaliste : l'objectif doit être réalisable, c'est-à-dire qu'il doit être atteignable par l'équipe. Il doit tenir compte du contexte et des contraintes. Vous pouvez déterminer ce que vous pensez atteignable grâce à vos précédentes expériences avec des projets similaires.

Exemple :

- Augmenter votre chiffre d'affaires de 20% durant l'année courante par rapport à l'année précédente en pleine période de récession économique n'est pas réaliste, tout au moins pourrions-nous nous fixer comme objectif de conserver notre chiffre d'affaires
- de l'année dernière.

T = Temporellement défini : il faut vous imposer une limite dans le temps, une période au terme de laquelle vous devez avoir réalisé votre objectif. Si vous ne fixez pas de limite temporelle, vous serez moins motivé et donc, votre projet risque de traîner en longueur, voire d'être abandonné.

En somme, la **méthode de définition d'objectif SMART** présente l'avantage de ne négliger aucun aspect, elle est complète et permet de fixer dès le début du projet, une orientation précise pour celui-ci. De plus, il permet d'éviter de se rendre compte, au cours du processus de réalisation que, par exemple, le projet n'est ni assez précis, ni réalisable, ni mesurable.

source : https://www.kiwili.com/Blog/post/definir-objectifs-gestion-de-projet-methode-smart/)

Exemples : Objectifs SMART ou pas ?

- Je veux lancer une version améliorée de mon logiciel tous les 4 mois au cours de l'année à venir, soit 3 nouvelles versions en suivant la procédure de développement interne
- Nous voulons introduire un nouveau système comptable.
- Nous voulons standardiser nos processus de management de projet.
- Nous voulons réduire la durée moyenne et le coût des projets de 20%

4.4. Elevator pitch

L'elevator pitch consiste à définir en une minute en quoi consiste le projet (en utilisant notamment les notions de raisons business et objectifs).

Pour <l'initiateur du projet>,
Qui <raison business>,
Le <nom du projet>,
Devra <objectif>

Exemple :
Pour le département informatique,
Qui doit réduire la durée moyenne et le coût des projets de 20%,
Le projet d'amélioration de la gestion de projets,
Délivrera à tous les chefs de projet une copie du manuel de gestion de projet avant le 1er novembre 2025.

4.5. Description du produit final

La description du produit à ce niveau est une description de haut niveau exprimée en termes de livrables. C'est en fait le « Quoi » du projet qui sera délivré au client à la fin du projet.

Il convient de décrire :

1. ce qui est attendu au niveau du produit final, comme par exemple : un système d'enregistrement pour des formations en ligne avec les quelques attentes utilisateurs que vous aurez pu récolter

2. la taille du produit comme par exemple la taille en termes de nombre d'écrans, nombre d'objets, nombre de points de fonctions (=métrique qui a pour but d'évaluer ou de mesurer les systèmes d'information en termes de richesse fonctionnelle livrée à l'utilisateur, de son point de vue, donc du métier), …

3. ce qui est dans le contenu et hors du contenu du projet, comme par exemple les formations des utilisateurs, la maintenance du produit, le guide utilisateur…
4. Autres points qui pourraient intervenir dans la description du produit final

4.6. Attentes qualité du produit

Les attentes en ce qui concerne la qualité à laquelle le produit doit correspondre doivent être connues dès le début sans oublier toutefois que la qualité a un coût !

Quelques éléments qui pourraient être pris en considération dans un projet informatique :

- Performance : temps de réponse, temps de démarrage…
- Exactitude : le résultat est-il toujours correct à 100% ?
- Sécurité : les accès sont-ils bien sécurisés ? (double authentification…)
- Compatibilité : dans quelle mesure notre produit doit-il être compatible avec d'autres ? (Interopérabilité – API – Web Services...)
- Fiabilité : Quelle est la fréquence de panne tolérée ?
- Entretien : le produit doit-il être facile d'entretien et de réparation ?
- Flexibilité : Est-il facile de modifier le produit ?
- Comparaison avec d'autres produits : pouvons-nous référencer le produit avec d'autres produits similaires ? (Benchmarking)
- Portabilité : est-il facile de porter le produit dans un autre environnement ?
- …

4.7. Approche projet

Comment allons-nous réaliser l'objectif projet ?

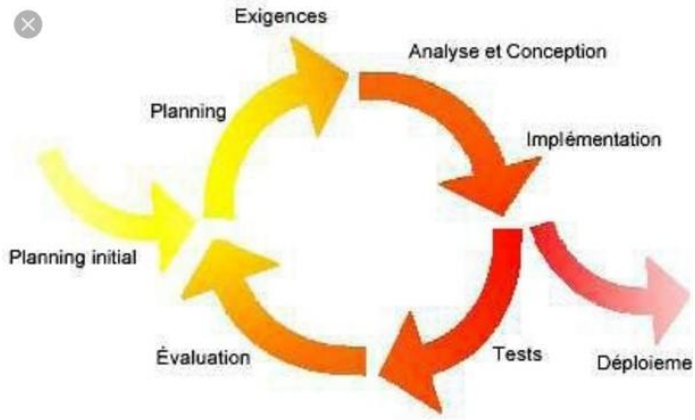
- Allons-nous acheter ou développer en interne ?
- Si développement en interne, allons-nous le faire en collaboration avec des consultants externe (qui travailleraient au sein de l'équipe de développement) ?
- Quel type de cycle de vie allons-nous appliquer pour le développement du produit ?
    - Waterfall (Analyse, Design, Développement, Tests, Implémentation)
    - Itératif
    - Agile
    - …

Un certain nombre de ces méthodes feront l'objet du dernier chapitre de ce cours.

Waterfall

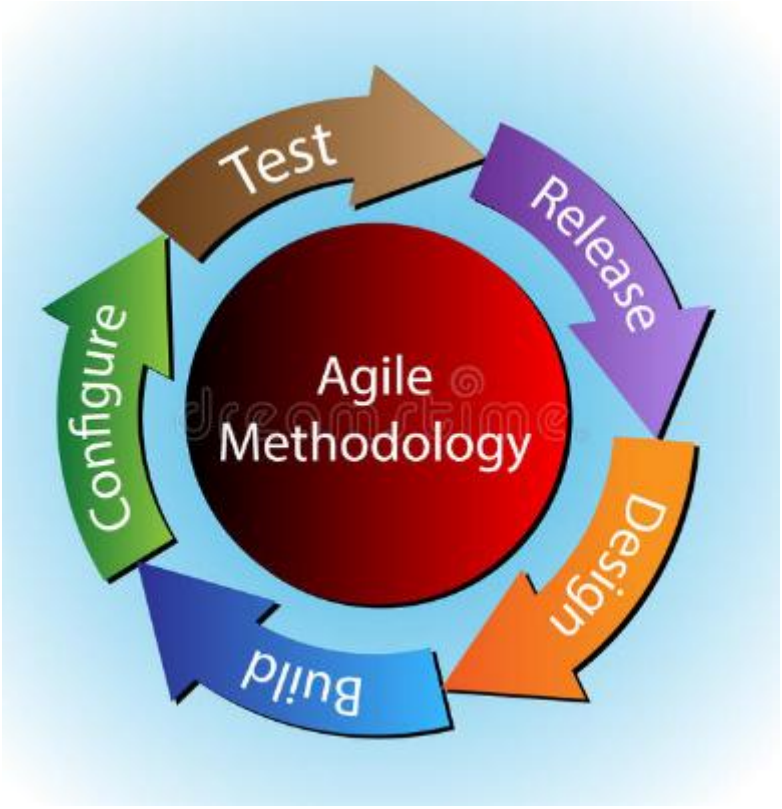
Source :
https://www.google.be/search?biw=1366&bih=662&tbm=isch&sa=1&q=cycle+de+vie+waterfall&oq=cycle+de+vie+waterfall&gs_l=psy-ab.3..0i24k1.420536.425293.0.426831.11.11.0.0.0.0.560.1529.2j0j2j1j0j1.6.0....0...1.1.64.psy-ab..5.6.1527...0j0i67k1.gETJc8y64AU#imgrc=xTnoHrcsWofBjM:

Cycle itératif



Source :
https://www.google.be/search?biw=1366&bih=662&tbm=isch&sa=1&q=cycle+de+vie+it%C3%A9ratif&oq=cycle+de+vie+it%C3%A9ratif&gs_l=psy-ab.3..0j0i24k1.94639.101366.0.101880.17.17.0.0.0.0.61.872.17.17.0....0...1.1.64.psy-ab..0.17.864...0i67k1.BrB3-4CSvhQ#imgrc=U8aG7Ft-Ec-emM:

Cycle AGILE



Source : Agile (https://fr.dreamstime.com/illustration-stock-dirigez-l-illustration-du-cycle-de-vie-agile-de-d%C3%A9veloppement-de-m%C3%A9thodologie-et-de-logiciel-image66724144)

L'approche sélectionnée devra être alignée avec le type de projet, les capacités et attentes des parties prenantes les plus importantes, la culture de l'entreprise…

4.8. Plan à jalons (Milestones Plan)

Le but ici est de développer un échéancier (planning) sommaire en mentionnant les jalons les plus importants et leurs dates

Exemple

N°	Evénement	juin	juillet	août	sept	oct	nov
1	Signature charte de projet	v					
2	Analyse terminée		v				
3	Prototype approuvé			v			
4	Release 1 : prêt pour les tests d'acceptation					v	
5	Release 2 : mise en production						v

Le planning d'un projet dépend :

- De la date à laquelle le projet devrait être livré (pour un développement d'un nouveau produit)
- De l'atteinte des résultats spécifiques à des intervalles donnés.

Un jalon est un événement significatif d'un projet. Un jalon est, la plupart du temps, lié à la livraison d'un ou de plusieurs livrables. Il est suffisant de définir, pendant la phase de démarrage, la date des jalons majeurs.

Remarque :
Un planning détaillé des activités sera effectué lors du processus de planification. Il est injustifié, lors de la phase de démarrage, de faire un planning détaillé pour les activités qui ne seront effectuées que dans 6 mois par exemple. Lorsque le chef de projet utilise des outils (MS Project par exemple), il sera tenté de faire une planification détaillée dès le démarrage du projet. Or, beaucoup de tâches sont encore incertaines et changeront. Le chef de projet passera dès lors beaucoup trop de temps à la planification des activités. Ces outils peuvent néanmoins être utilisés pour définir un planning des jalons à plus haut niveau.

4.9. Estimation budget

A ce stade, une estimation précise des coûts ne peut probablement pas être effectuée, or le propriétaire du projet veut la connaître.

Il faudra faire une estimation budgétaire :

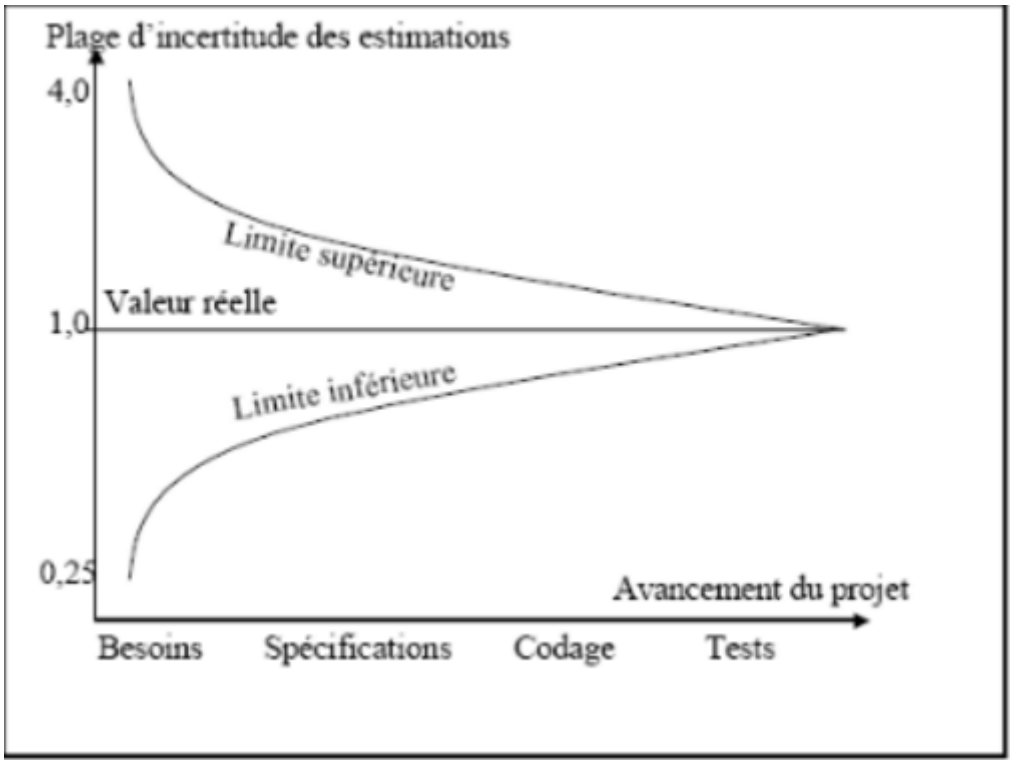
- En mentionnant une mention concernant la précision (ex : +/- 50%...)
- En incluant une mention des hypothèses et restrictions qui ont été utilisées pour déterminer l'estimation

L'estimation du budget est basée sur :

- La comparaison avec des projets similaires dans la même ou une autre organisation
- La comparaison avec d'autres projets dans la même organisation (*)
- Le coût actuel des ressources utilisées
- Autres sources…

* Ceci dépend toutefois du niveau de documentation dans l'organisation (actifs organisationnels)

Schéma de la plage d'incertitude budgétaire au fil de l'avancement du projet



Source : https://www.google.be/search?biw=1366&bih=662&tbm=isch&sa=1&q=estimation+budg%C3%A9taire+projet+informatique&oq=estimation+budg%C3%A9taire+projet+informatique&gs_l=psy-ab.3...3322.21610.0.22102.41.41.0.0.0.0.104.2150.40j1.41.0....0...1.1.64.psy-ab..0.28.1468...0j0i67k1j0i30k1j0i8i30k1j0i24k1.Dj4Mhi9kQW4#imgrc=zhnp0scH2Go5pM:

Méthode d'évaluation budgétaire P.E.R.T.

Cette méthode est une bonne technique pour évaluer budgétairement un projet durant le processus d'initiation du projet. On l'appelle également la méthode à 3 points.

En résumé de très haut niveau :

- Minimum (**m**) : le délai/effort minimum possible requis pour accomplir une tâche, en assumant que tout se déroule mieux que prévu
- Maximum (**M**) : le délai/effort maximum possible requis pour accomplir une tâche, en assumant que tout aille mal (tout en excluant les catastrophes majeures)
- Plus vraisemblable (*most likely* = **ml**) : la meilleure estimation du délai/effort requis pour accomplir la tâche, en assumant que tout se déroule normalement
- Délai espéré (*expected time* = **ET**) :
  ET = (m + 4ml + M) / 6
  ET donnera la meilleure estimation selon cette méthode

4.10. Equipe de management de projet

L'équipe de management de projet est formée durant l'initiation du projet.

Celle-ci est constituée :

- Du comité de pilotage
- Du chef de projet

Le **comité de pilotage** est constitué de :

- Le commanditaire : la personne qui est responsable au niveau business, de la raison d'être du projet et donc du business case du projet.
- L'utilisateur senior : la personne qui représente l'utilisateur du produit final du projet. Il assurera que les exigences soient bien claires, que l'on puisse tester le produit et qu'il réponde bien aux besoins de l'utilisateur.
- Le fournisseur senior : la personne qui est responsable de la livraison des produits du projet ; il est responsable de la qualité de tous les produits fournis par les fournisseurs. Il a l'autorité pour allouer des ressources.
- Du représentant assurance qualité : son rôle est de s'assurer, durant toute la durée du projet, que le produit réponde aux exigences qualité et aux exigences des clients et que les processus qualité sont bien suivis et répondent aux besoins
- Autres membres éventuels : responsable des Finances, de l'IT…

4.11. Facteurs influents

- **Hypothèses** : Faits, événements ou actions que vous supposez être vrais. Par exemple, vous assumez que vous avez toute la compétence nécessaire pour le développement d'un logiciel dans la société.
- **Dépendances** : Nous devons déterminer si des dépendances externes existent entre notre projet et d'autres activités non-projet. Par exemple, la mission de la société doit être définie et validée par le Comité de Direction avant que l'on puisse la publier sur notre site internet.
- **Contraintes** : Facteurs qui limitent les choix des chefs de projet comme par exemple :
    - Date imposée
    - Budget maximal
    - Nombre de ressources disponibles
    - Connaissances et disponibilités
    - …
- **Problèmes majeurs** : questions ouvertes auxquelles nous ne pouvons pas répondre à l'heure actuelle comme le nombre d'utilisateurs simultanés d'un système, les problèmes relationnels entre membres dans les différentes équipes, la maturité de l'entreprise en gestion de projets…
- **Parties prenantes** : Liste des parties prenantes les plus importantes
    - Utilisateurs clefs
    - Management / Décisionnaires
    - …

Les différents éléments répertoriés ci-dessus seront des sources de risques. Ils pourront donc apparaître dans le registre des risques selon leur importance (probabilité/impact)

- **Risques** : éléments incertains qui peuvent survenir et avoir un impact sur le bon déroulement de notre projet (donc sur l'atteinte des objectifs).
    - Les différents points mentionnés ci-dessus sont des sources de risques, ceci ne veut pas dire qu'ils devront d'office être repris dans le tableau des risques, ceci dépend surtout de la probabilité qu'ils se produisent et de l'impact qu'ils pourraient avoir sur le projet.
    - Il peut y en avoir d'autres risques (légal, juridique…).
    - La notion de risque dans un projet sera expliquée plus dans le détail lors de la phase de planification.

4.12. Meeting de pre-kickoff

C'est une réunion de présentation du projet avec le 'Comité de Pilotage' (COPIL) en vue d'obtenir le 'GO' passer à la phase suivante du projet, la planification.

Si nous n'obtenions pas cet accord, deux situations possibles :

- Le COPIL peut demander de revoir certains éléments de la charte de projet et une nouvelle présentation aura lieu une fois cette révision réalisée.
- Le COPIL refuse le projet et celui-ci est arrêté.

Organisation de cette réunion :

- Passer en revue la charte de projet avec les propriétaires du projet et les autres parties prenantes
- Prenez le temps nécessaire pour clarifier les choses !
- Formaliser la décision à la fin de la réunion.

5. La gestion des exigences

La gestion des exigences peut démarrer lors de la phase de démarrage du projet et être exécuté plusieurs fois durant le projet.

Recueillir les exigences est un processus qui consiste à déterminer, documenter et gérer les besoins et les exigences des parties prenantes dans le but d'atteindre les objectifs du projet. L'intérêt principal de ce processus est qu'il sert de base à la définition et à la gestion du contenu et du périmètre du projet.

La gestion des exigences est donc fondamentale, même si souvent négligée ou incomplète ce qui en fait une des causes majeures d'échecs de projets !

Quelques exemples de dangers lorsque la gestion des exigences n'est pas bien utilisée :

- On accepte des exigences, quel que soit la source
- Le projet subit un grand nombre de changements et donc beaucoup de travail à recommencer…
- Il n'est pas possible de donner la preuve que le produit réponde aux exigences approuvées (manque de traçabilité des exigences).
- Des clients clés ont été oubliés lors des interviews.
- …

5.1. Qu'est-ce qu'une exigence

" Quelque chose que le produit doit faire ou une propriété que le produit doit avoir et qui est désirée par les parties prenantes ou dont ils ont besoin " (même s'ils n'en sont pas conscients pour l'instant !)

5.2. Quelques points importants au niveau des exigences :

a) Souhaits – Exigences

Un souhait est une valeur ou une qualité produit souhaitée par une des parties prenantes, sur laquelle aucun accord n'a été pris

Une exigence est un souhait validé, donc retenu par le chef de projet et le COPIL.

b) Les degrés des exigences : MOSCOW (source WIKIPEDIA)

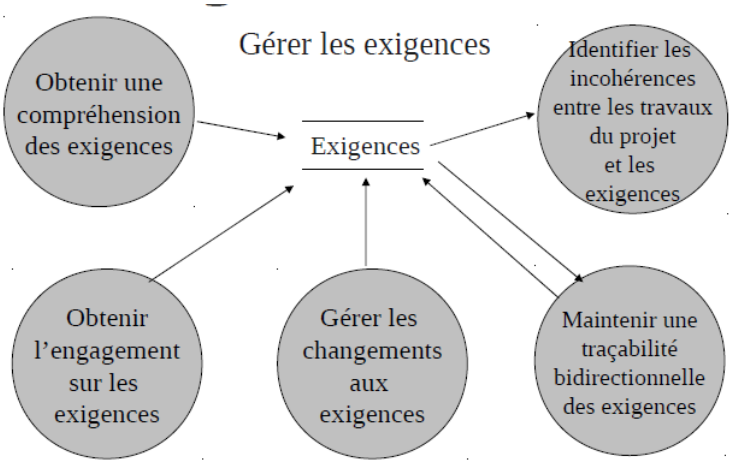
La méthode **MoSCoW** est une technique visant à prioriser des besoins ou des exigences en matière d'assistance à maîtrise d'ouvrage et de développement logiciel. L'objectif est que le maître d'œuvre et le maître d'ouvrage s'accordent sur l'importance des tâches à réaliser par rapport aux délais prévus.

Les lettres majuscules de l'acronyme *MoSCoW* signifient (en anglais):

- **M** : must have this, c'est-à-dire 'doit être fait' (vital).
- **O** : néant
- **S** : should have this if at all possible, c'est-à-dire devrait être fait dans la mesure du possible (essentiel).
- **C** : could have this if it does not affect anything else, pourrait être fait dans la mesure où cela n'a pas d'impact sur les autres tâches (confort).
- **O** : néant
- **W** : won't have this time but would like in the future, ne sera pas fait cette fois mais sera fait plus tard' (luxe, c'est votre zone d'optimisation budgétaire).

c) Gestion des exigences et traçabilité : quelques conseils intéressants résumés dans cette source trouvée sur Internet →
https://services.renater.fr/_media/sourcesup/formation/cdp_coursgestionexigences.pdf

d) Pratiques clefs de la gestion des exigences



e) Fournisseurs d'exigences (parties prenantes) :

D'où viennent les exigences ? Qui sont les parties prenantes ?

Réponse : « Chaque personne qui a un intérêt au produit et qui, dès lors, a une exigence du produit »

Le client : paie pour le développement du produit (=sponsor)

L'utilisateur : il utilise finalement le produit. Il faut distinguer les différents groupes d'utilisateurs qui auront des exigences différentes !

Equipe de développement : crée le produit

Les autres fournisseurs d'exigences :

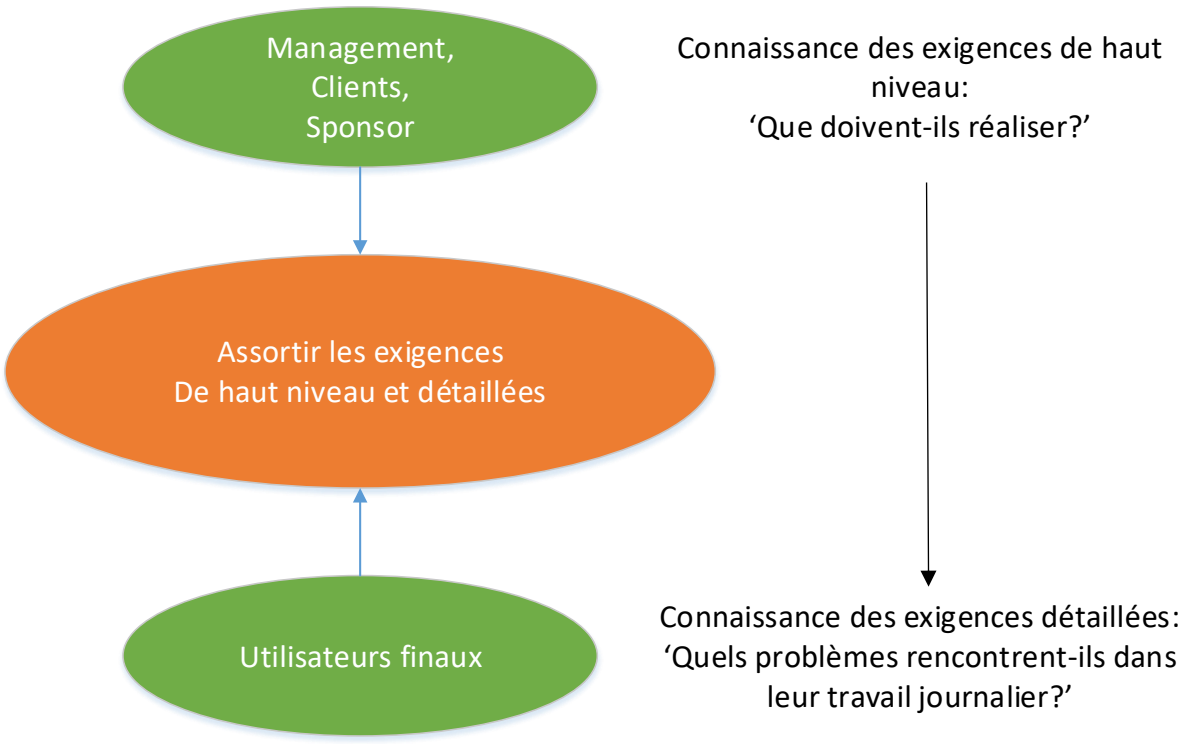
- Conseil d'Administration
- Management IT / Staff IT
- Légal
- …

Les fournisseurs d'exigences doivent être identifiés de façon formelle :

- Si nous n'identifions pas les parties prenantes 'critiques' et leurs besoins :
  - Vous ne saurez probablement pas identifier les requis critiques, ce qui mènera tôt ou tard à des problèmes
- Si vous n'identifiez pas les sources officielles d'exigences :
  - Des discussions non-officielles mèneront à des exigences additionnelles, ce qui mènera à des problèmes

5.3. Techniques de développement des exigences

5.3.1. Introduction : méthodes Top-Down et Bottom-Up



Dans le schéma ci-dessus nous retrouvons les exigences '*Top Down*' (provenant du management, des clients et des sponsors) et les exigences '*Bottom-Up*' (besoins des utilisateurs pour les aider dans leur travail quotidien).

Approche Top Down et Bottom-Up :

Source : https://www.wrike.com/fr/blog/gestion-de-projets-ascendante-et-descendante/

On observe actuellement de nombreux changements dans le monde de la gestion, et en particulier dans la gestion de projets. Nous avons appris que des organisations de renommée mondiale **ont délaissé le style de gestion descendante et ont adopté une approche ascendante**. Certaines ont notamment intégré des éléments du style de gestion ascendante dans certains de leurs services.

Nous constatons donc que la gestion de projets ascendante est de plus en plus utilisée. Toutefois, les deux approches continuent de faire l'objet de discussions animées. Pourquoi les organisations souhaitent-elles changer à tout prix leur style de gestion ? Une simple comparaison des deux approches de gestion permet de répondre à cette question.

Approche Descendante (Top-Down)

L'approche descendante reste extrêmement populaire dans la gestion de projets. L'expression « top-down » (descendante) signifie que les instructions sont données en amont. Les objectifs du projet sont fixés par la direction. Les cadres supérieurs fournissent des lignes directrices, des informations, des plans et des processus de fonds. Toutes les attentes du chef de projet sont communiquées clairement à chaque participant au projet.
Par conséquent, toute ambiguïté peut mettre en péril la bonne gestion du projet, et les chefs de projet doivent être aussi précis que possible lors de la communication de leurs attentes. Le caractère formel des processus est essentiel pour cette approche.

Vous trouverez des exemples d'application de l'approche descendante dans de nombreuses organisations.

Prenons par exemple le New York Times. Il y a quelques années, American Journalism Review (www.ajr.com) révélait que la direction du Times estimait qu'il fallait redoubler d'efforts pour mettre en place un cadre de travail plus dynamique et pour que l'organisation prospère. Ils ont dès lors procédé à une centralisation du pouvoir, accordant aux rédacteurs en chef un contrôle plus global de leurs projets. Les décisions prises pour l'ensemble des projets dépendaient du flair et de l'avis d'une seule personne : le chef de projet.

Pour quel résultat ? Les membres de l'équipe avaient le sentiment qu'ils n'étaient pas entendus et que leur avis ne comptait pas. Les journalistes ne collaboraient pas de manière efficace et n'étaient plus motivés par leur travail. Les cadres dirigeants se sont rendu compte qu'il fallait donner plus de liberté aux équipes et qu'ils devaient changer leur fusil d'épaule. L'introduction d'une gestion de projets ascendante ne s'est pas faite du jour au lendemain. Pourtant, le temps et les efforts consacrés à la mise en place de cette nouvelle approche

en valaient la peine, les employés du New York Times ayant déclaré que la collaboration était devenue beaucoup plus efficace et que les membres de l'équipe travaillaient désormais ensemble de façon plus productive.

L'expérience montre que cette gestion descendante se traduit souvent par une productivité réduite et provoque des goulots d'étranglement ou des verrouillages. Un verrouillage donne au chef de projet un contrôle total sur son équipe. Ces verrouillages peuvent causer des souffrances inutiles et ralentir considérablement la réalisation d'un projet.

Approche Bottom-Up

Les facteurs mentionnés ci-dessus peuvent fortement contribuer à l'échec d'un projet, et c'est la raison pour laquelle de nombreuses organisations ont adopté un style de gestion ascendante ou au moins certains de ses éléments ; c'est l'exemple du cas du *New York Times*.

L'approche ascendante se caractérise par une participation proactive de l'équipe dans le processus d'exécution du projet. Les membres de l'équipe sont invités à participer à toutes les étapes du processus de gestion. L'ensemble de l'équipe est amené à décider de la marche à suivre. Le style de gestion ascendante permet aux chefs de projet de communiquer les objectifs et les priorités, notamment à l'aide d'un planning reprenant les étapes clés. Les membres de l'équipe sont ensuite encouragés à développer des listes de tâches personnelles et à répartir leur temps de travail comme ils le souhaitent pour atteindre les objectifs.

L'équipe est libre de choisir sa méthode de travail et la façon dont elle exécutera ses tâches. L'avantage de cette approche est qu'elle donne aux membres de l'équipe les moyens de réfléchir de manière plus créative. Ils se sentent associés au développement du projet et savent que leurs initiatives sont appréciées.

Par ailleurs, les membres de l'équipe se sentent bien plus motivés et ont réellement envie de voir leur projet aboutir. Chaque membre de l'équipe a également l'occasion de trouver des solutions qui se concentrent davantage sur des besoins pratiques que sur des notions abstraites. Le processus de planification est facilité par un certain nombre de personnes et s'exécute ainsi plus rapidement. Les listes de tâches de tous les membres de l'équipe sont recueillies et intégrées au plan global et détaillé du projet. Les échéances, budgets et résultats sont transparents.

Bien qu'il offre de nombreux avantages, le style de gestion ascendante à lui seul ne suffit pourtant pas à faire aboutir vos projets sans encombre. Selon de nombreux experts, l'approche ascendante ne constitue pas la solution idéale, car elle manque parfois de clarté et n'offre pas toujours un contrôle absolu. Pour une gestion de projets optimale, il vaut mieux trouver un équilibre entre les deux approches opposées et prendre uniquement ce qui fonctionne le mieux dans chacune d'elles.

Equilibre entre la méthode TOP-DOWN et BOTTOM-UP

L'application des meilleures pratiques de l'approche ascendante mène à certaines difficultés à le faire en utilisant des outils traditionnels de gestion de projets**.** Un logiciel de gestion de projet traditionnel comme Microsoft Project a été conçu en grande partie pour s'adapter à l'utilisation de l'approche descendante et ne convient pas pour le style de gestion ascendante.

Ce logiciel est axé sur le chef de projet et le place au centre des communications sur le projet. Bien souvent, les membres de l'équipe ont uniquement accès au plan du projet en lecture seule, et ne peuvent pas y apporter des modifications ni leur contribution.

Les employés envoient leurs mises à jour au chef de projet par courrier électronique, dans des fichiers distincts. Le chef de projet doit alors recueillir l'ensemble des données et intégrer ces informations manuellement au plan du projet. Ensuite, il doit communiquer les changements effectués aux cadres de l'entreprise.

Toutes ces procédures de routine empêchent le chef de projet d'exercer sa fonction et de mettre ses compétences à contribution. Cette quantité pharaonique de travail de contrôle mécanique et de synchronisation ne laisse que très peu de place à la gestion des équipes, la fonction première du chef de projet.

Heureusement, la situation évolue grâce aux changements qui s'opèrent actuellement dans la façon dont les gens échangent et reçoivent des informations. Plusieurs méthodes visant à intégrer avec succès les meilleures pratiques de la gestion ascendante voient le jour. Parmi ces méthodes, on retrouve les Technologies d'entreprise 2.0, à savoir les wikis, blogs, réseaux sociaux, outils collaboratifs, etc.

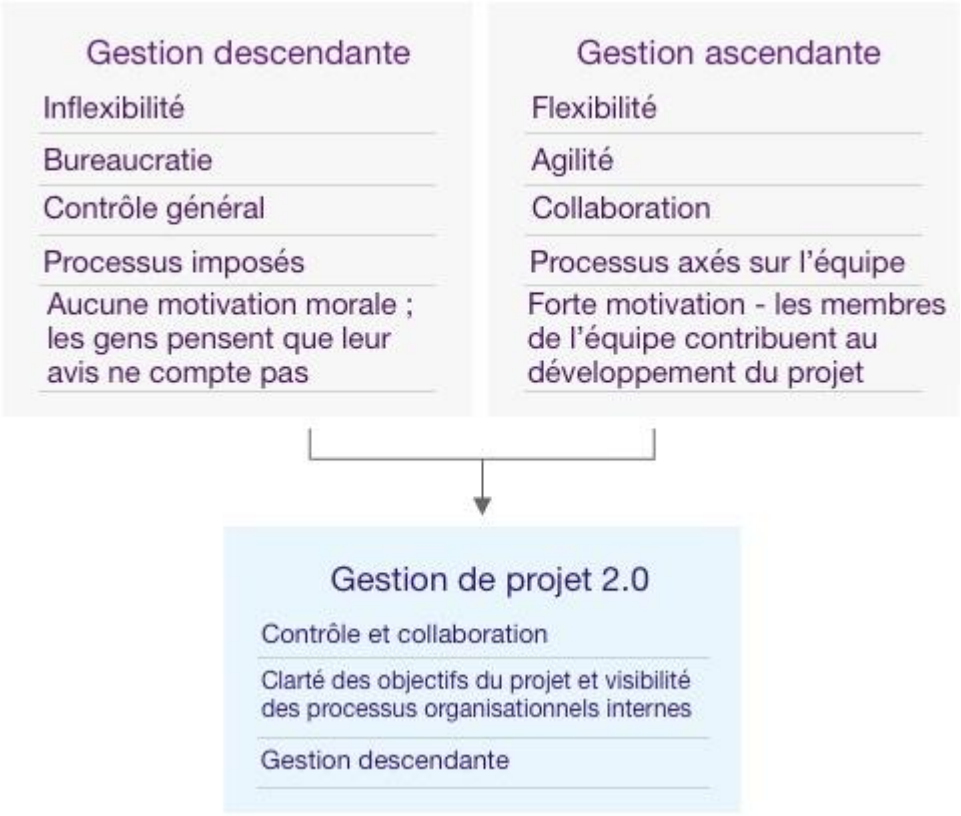
Elles sont de plus en plus présentes dans les organisations et changent leur façon de réaliser des projets. Elles transforment la gestion de projets traditionnelle en gestion de projet 2.0 et apportent de nouveaux modes de collaboration, qui sont basés sur l'intelligence collective, c'est-à-dire l'ensemble des connaissances détenues par chaque membre de l'équipe du projet dans différents domaines.

Il est désormais possible de recueillir ces informations et de les partager au sein d'un environnement collaboratif et flexible grâce à un logiciel de gestion de projets de deuxième génération. Le chef de projet est chargé de diriger les travaux de son équipe et détermine la direction que doit prendre le projet en fonction des renseignements communiqués par chaque employé.

Ainsi, le chef de projet ne joue plus le même rôle dans le projet. Le logiciel de gestion de projets 2.0 facilite la délégation des tâches. Cela signifie que les gens dépendent moins du chef de projet en tant que générateur de tâches. Son rôle ne consiste plus à attribuer des tâches, mais à coordonner le projet. Il doit faciliter la communication entre les membres de l'équipe,

fournir un environnement de travail créatif et guider l'équipe. Il devient alors un visionnaire capable de tirer parti des forces et des faiblesses de l'équipe et d'ajuster le développement du projet, en fonction de divers changements externes. Chaque membre de l'équipe reste libre d'atteindre un objectif clé comme il l'entend.

Grâce aux outils de gestion projets de deuxième génération, les chefs de projets sont en mesure de combiner les avantages des deux approches de gestion. Ces outils les aident à combiner contrôle et collaboration, clarté des objectifs du projet et visibilité des processus organisationnels internes.



5.3.1.1. Outils et Techniques

Parmi les outils et techniques de collecte des données pouvant être utilisés pour ce processus figurent notamment les éléments suivants :

- Brainstorming : Technique utilisée pour répertorier une liste d'idées en peu de temps. Il est organisé dans un environnement de groupe sous la conduite d'un animateur. Il comprend deux parties, la génération d'idées et l'analyse. Le brainstorming peut être utilisé pour collecter les données, les solutions ou les idées des parties prenantes, des spécialistes et des membres de l'équipe pendant la réalisation de la charte de projet.

- Groupe de discussion : Les groupes de discussion réunissent les parties prenantes et les spécialistes dans le but de définir la perception des risques, les critères de réussite et d'autres thèmes de façon plus conversationnelle que des entretiens individuels.

- Entretiens : ils sont utilisés pour obtenir des informations sur les exigences de haut niveau, les hypothèses, les contraintes, les critères d'approbation et d'autres informations auprès des parties prenantes en parlant directement avec elles.

- Questionnaires et enquêtes : ce sont des ensembles de question qui permettent de recueillir rapidement des informations à partir des réponses d'un grand nombre de personnes. Les questionnaires et enquêtes conviennent surtout dans le cas de groupes hétérogènes et lorsqu'un résultat rapide est nécessaire.

- Benchmarking : Il consiste à comparer les produits, les processus et les pratiques à ceux d'organisations comparables dans le but d'identifier les bonnes pratiques et de trouver des idées d'améliorations.

- Jugements d'experts : Une telle expertise peut être fournie par un groupe ou une personne ayant suivi des formations adéquates et possédant une connaissance, une compétence.

  Ces points peuvent notamment toucher les domaines suivants :
  - Stratégie organisationnelle
  - Gestion des bénéfices
  - Connaissances techniques du secteur d'activité et du domaine d'intérêt spécifique du projet
  - Estimation de durée et budgétaire
  - Identification des risques

6. Annexes

6.1. Exemple de modèle de spécifications des exigences

Types d'informations associables à une exigence	Signification
Identificateur	Identificateur unique
Exigence	Décrire la demande de la partie prenante
Emetteur	Qui est à la source de cette exigence ?
Méthode de vérification et validation	Comment peut-on vérifier que l'exigence est satisfaisante pour le demandeur ?
Justifications	Éléments permettant de démontrer (ou prouver) le bien-fondé ou les raisons du choix de l'exigence.
Relations	Donner les liens avec d'autres exigences s'il en existe.
Catégorie (pour information dans le cadre de ce cours)	**Exemples de catégories** : Exigences : Business : besoins généraux de l'organisation Des parties prenantes : exigences des PP De la solution : décrivent les propriétés, les fonctions et les caractéristiques du produit, du service ou du résultat qui satisferont aux exigences business et à celles des PP. Parmi celles-ci nous retrouvons : *les exigences fonctionnelles* qui décrivent les comportements du produit comme par exemple les processus, les données et les interactions que le produit devrait exécuter. *Les exigences non-fonctionnelle* : elles complètent les exigences fonctionnelles et décrivent les conditions environnementales ou les qualités requises pour que le produit soit efficace comme par exemple, la fiabilité, la sûreté, la performance, la facilité d'emploi…

	Exigences du projet : comme par exemple le respect des jalons, les contraintes… Exigences Qualité : rassemblent toute les conditions et critères nécessaires à la validation de l'achèvement comme par exemple les tests, certifications, validations.
Flexibilité	Capacité de l'exigence à être négociée
Priorité	Niveau d'importance pour la prise en compte de l'exigence par rapport aux autres
Risques	Niveau de risque ou liste des risques engendrés par la prise en compte de l'exigence exprimée telle quelle.
État (statut)	État d'avancement de l'exigence : énoncée, officialisée, implémentée, satisfaite
Confidentialité	Niveau de confidentialité de l'exigence
Historique	Ensemble de repères permettant d'identifier les modifications qu'a subi l'exigence : date, auteur, identification et contenu de la modification, type de modification (ajout, suppression, …), justification ou raison de la modification.
Commentaires	Zone texte libre

Remarque importante :
Pour rappel, PMBOK est un guide des bonnes pratiques en gestion de projet, qui peut être utilisé dans différents domaines de l'entreprise et non pas seulement pour les projets IT. Pour une entreprise donnée, ce guide doit être adapté aux besoins de l'entreprise et ne doit pas être suivi à la lettre, chacun prend ce qu'il estime nécessaire dans ce guide pour gérer ses projets en interne et acquérir une certaine maturité à ce niveau. Il en est donc de même pour tous les composants du guide, comme le modèle proposé ci-dessus, vous devez l'adaptez à vos besoins en supprimant des informations inutiles pour votre organisation, voire en ajouter.

6.2. Exemple d'éléments de charte de projet :

1. Titre
2. Documents de référence
3. Présentation du projet
4. L'équipe de projet
5. Les attentes auxquelles doit répondre le projet
6. Les objectifs généraux du projet
7. La justification du projet et les bénéfices attendus
8. Les critères de succès du projet
9. Les risques identifiés
10. Les hypothèses
11. Les contraintes
12. Les exclusions
13. Les facteurs clés de succès
14. L'organisation du projet

6.3. Exemple de registre des parties prenantes

ID	Nom	Entreprise ou organisation	Titre	Rôle dans le projet	Influence Positive, négative ou neutre	Impact sur quelle phase	Priorité	Commentaire

6.4. Exemple d'une checklist simple

Document	Fichier	Statut	Commentaire / Réserves
Charte de projet	<nom> et <version>	<approuvé ou non>	
Registre des parties prenantes	<nom> et <version>	<approuvé ou non>	
Présentation du lancement de projet	<nom> et <version>	Approuvée en séance	Présentation des résultats de la phase de lancement
DECISION			**GO ou NO GO**

6.5. Exercice 1 – Etablir une charte de projet

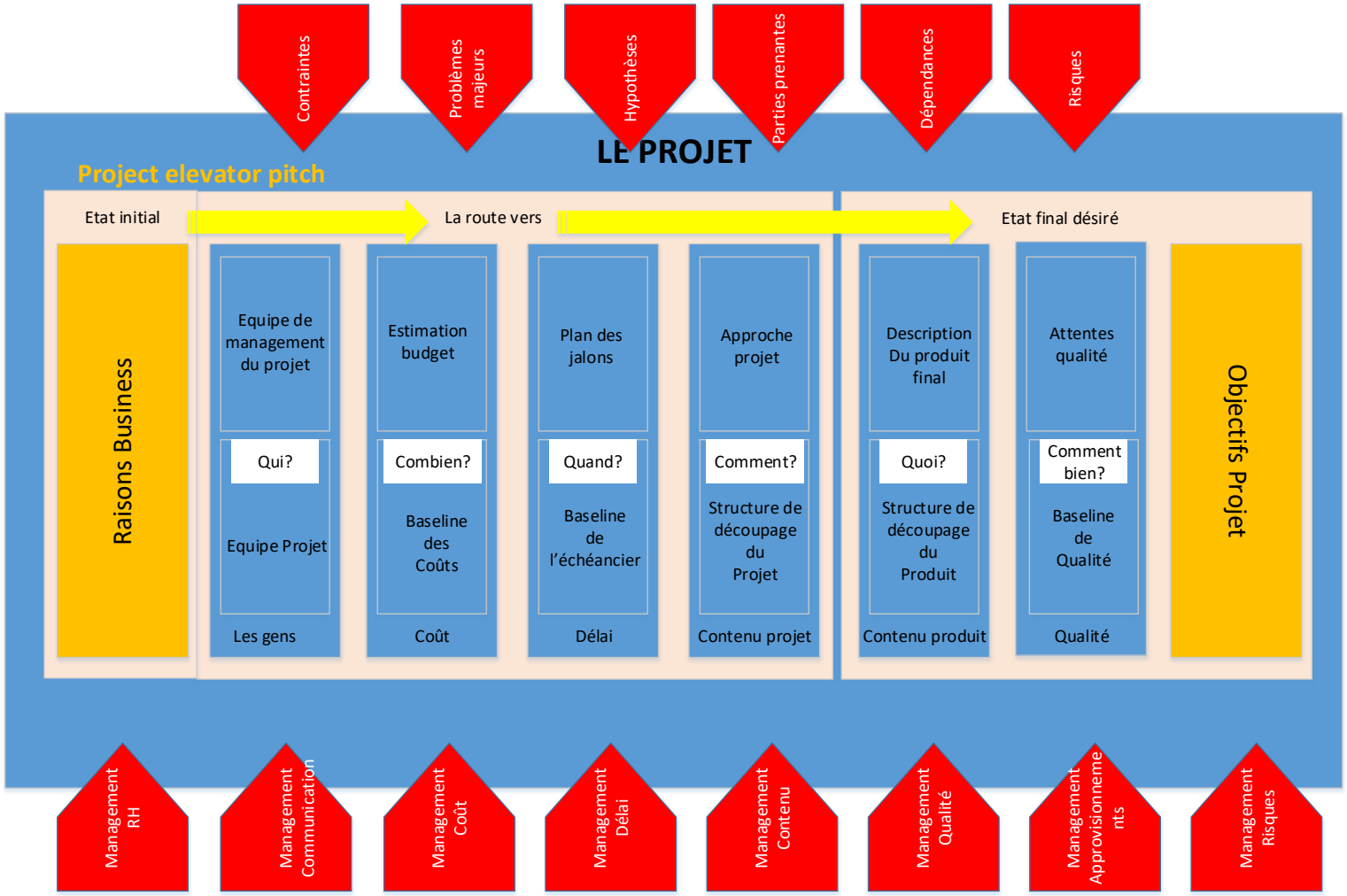
7. Plan de projet

7.1. Introduction

Votre charte de projet a été présentée au comité de pilotage et est à présent validée, ce qui signifie que vous avez reçu le 'Go' officiel pour passer à la phase suivante de votre projet, la phase de planification dont le livrable final est le plan de projet.
Dans celui-ci nous allons détailler plus en avant les différentes rubriques mentionnées dans la charte de projet et en ajouter de nouvelles relatives à l'exécution du projet.
Ceci permettra de pouvoir exécuter notre projet de façon ordonnée, tant au niveau de sa gestion que de son exécution.

Pour démarrer l'explication, nous allons repartir de la feuille de route du projet et y ajouter les éléments complémentaires nécessaires pour constituer un plan de projet.

La feuille de route du projet

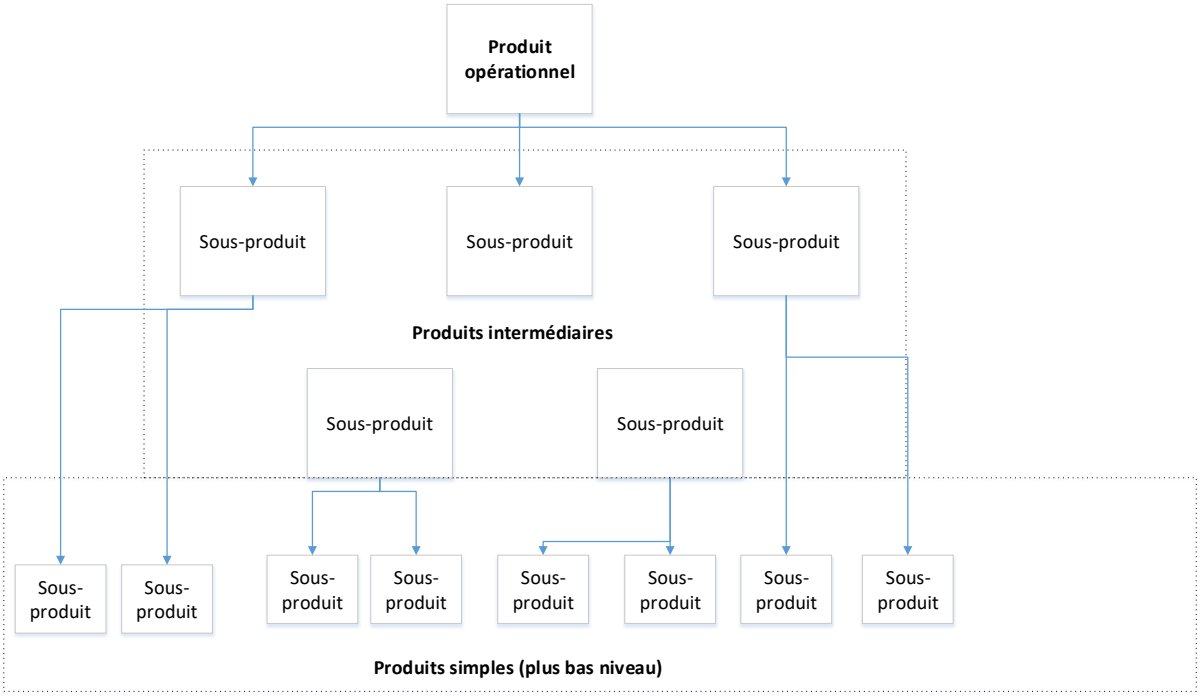


7.2. Structure de découpage du produit (PBS)

Le ‘Product Breakdown Structure’ (PBS) ou Structure de découpage du produit est un diagramme 'Top-to-Bottom' (du haut vers le bas) qui montre la décomposition de tous les produits qui doivent être développés (Prince 2).
Décomposition du produit en sous-produits qui le constituent (les composants produit).

Pourquoi cette décomposition ?

- Elle identifie les produits requis par le client
- Elle identifie les produits additionnels nécessaires pour construire et supporter les produits du client
- Crée un consensus sur le groupage des produits
- La planification à partir du produit du projet est un des principes clefs de Prince 2.

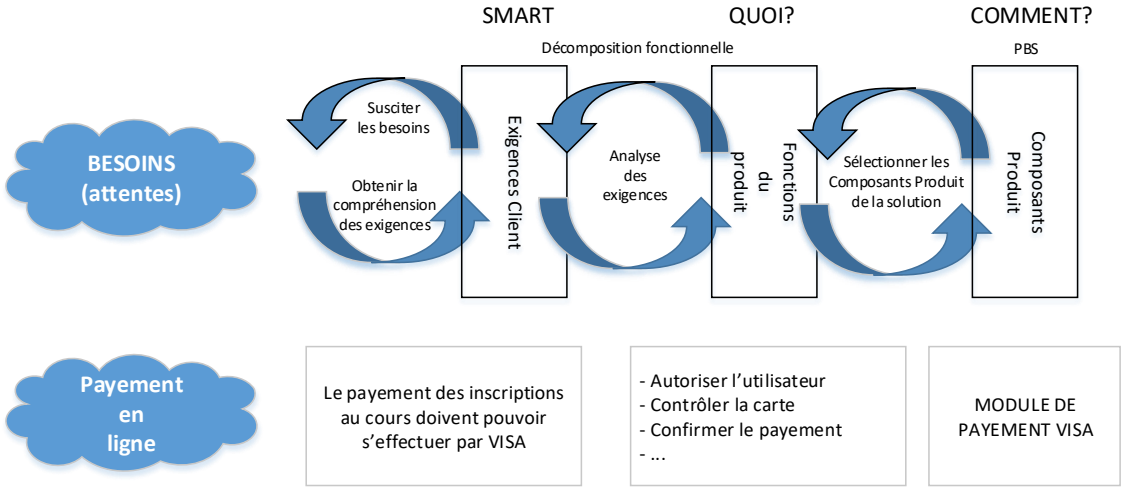


Remarques:

- Le niveau de découpage du produit dépend du produit en lui-même (ou du sous-produit).
  Le principe de découpage est le suivant :
  - Nous décomposons le produit jusqu'à obtenir ce que nous appellerons des 'Lots de travaux' qui doivent pouvoir être exécutés par une entité spécialisée dans le domaine concerné par ce lot de travail (un/des programmeurs, un/des analystes fonctionnels ou techniques, un/des gestionnaire(s) de base de données…).
- Le but de cette décomposition en lots plus petits est d'affiner les estimations de temps nécessaires à l'exécution du projet et les coûts du projet.
- Le principe d'élaboration progressive intervient à ce stade. Certains composants devront dès le début du projet être détaillés, d'autres composants ne seront détaillés que plus tard dans le projet.

7.3. Flux des exigences dans des projets informatiques

Pour un projet informatique, une décomposition du produit (PBS) peut être constituée de fonctions du produit.



Le management et l'analyse des exigences est un processus très important dans le cas de projets informatiques. Ces processus ne font, de façon stricte, pas partie du contenu de PMBOK (ou Prince 2). Nous fournissons, dès lors, un aperçu général des flux typiques des exigences dans les projets basés sur CMMI (Capability Maturity Model Integration).

Remarque :
Le CMMI est un modèle d'évaluation du niveau de maturité d'une organisation concernant le développement de systèmes, de produit et/ou de logiciels. Il a pour objectif la maîtrise des processus d'ingénierie et par conséquent celle de la qualité des produits et des services issus de ces processus. Il propose un

référentiel des meilleures pratiques (best practices) en matière de développement logiciel.
Source : Le CMMI: objectif, modèle, bénéfices et processus clés (fimarkets.com)

Processus CMMI :

- Susciter les besoins / Obtenir la compréhension des exigences :
  Les besoins/attentes sont constituées de tout ce que les parties prenantes (utilisateurs, management...) ont en tête concernant le produit futur : que doit-il faire, quel sera son look, quelle sera la performance?... Les besoins sont souvent exprimés à différents niveaux et exprimés via différents canaux (email, conversation, réunion...). Il faut dès lors passer le temps nécessaire pour transformer les besoins en exigences utilisateurs SMART et documentées.
  Le chef de projet doit s'assurer, avant de créer une *baseline*, qu'il y a une compréhension commune des exigences.

- Analyse des exigences:
  Les exigences du client seront, une fois bien comprises et finalisées (*baseline*), analysées plus en profondeur afin de déterminer les fonctions du produit. On répondra surtout à la question : "que devra faire le produit final? ". Une décomposition fonctionnelle peut être utilisée à cette étape afin de visualiser toutes les fonctions exigées. Il est également très important de maintenir la traçabilité entre les exigences du client et les fonctionnalités du produit.

- Sélectionner les composants produit de la solution:
  Les fonctions seront élaborées par le biais de composants produit. Quand nous avons une bonne vision de la fonctionnalité requise, les composants produits appropriés peuvent être sélectionnés en répondant à la question …Comment seront exécutées les fonctionnalités ?". Les contraintes budgets, délais et la qualité sont des facteurs qui doivent être pris en considération lors du choix du composant produit approprié.

Notez que, durant le processus, les exigences sont allouées à partir des fonctions produit vers les composants produit. Il est très important de maintenir une traçabilité complète.

Exemple: Structure de découpage du produit – Décomposition fonctionnelle

(Voir page suivante)

Astuce: PBS – Méthode des post-it (brainstorming)

Etape 1: Brainstorming
- Noter les produits et composants produits (un par post-it)

Etape 2: Rassembler
- Coller tous les post-it sur le mur

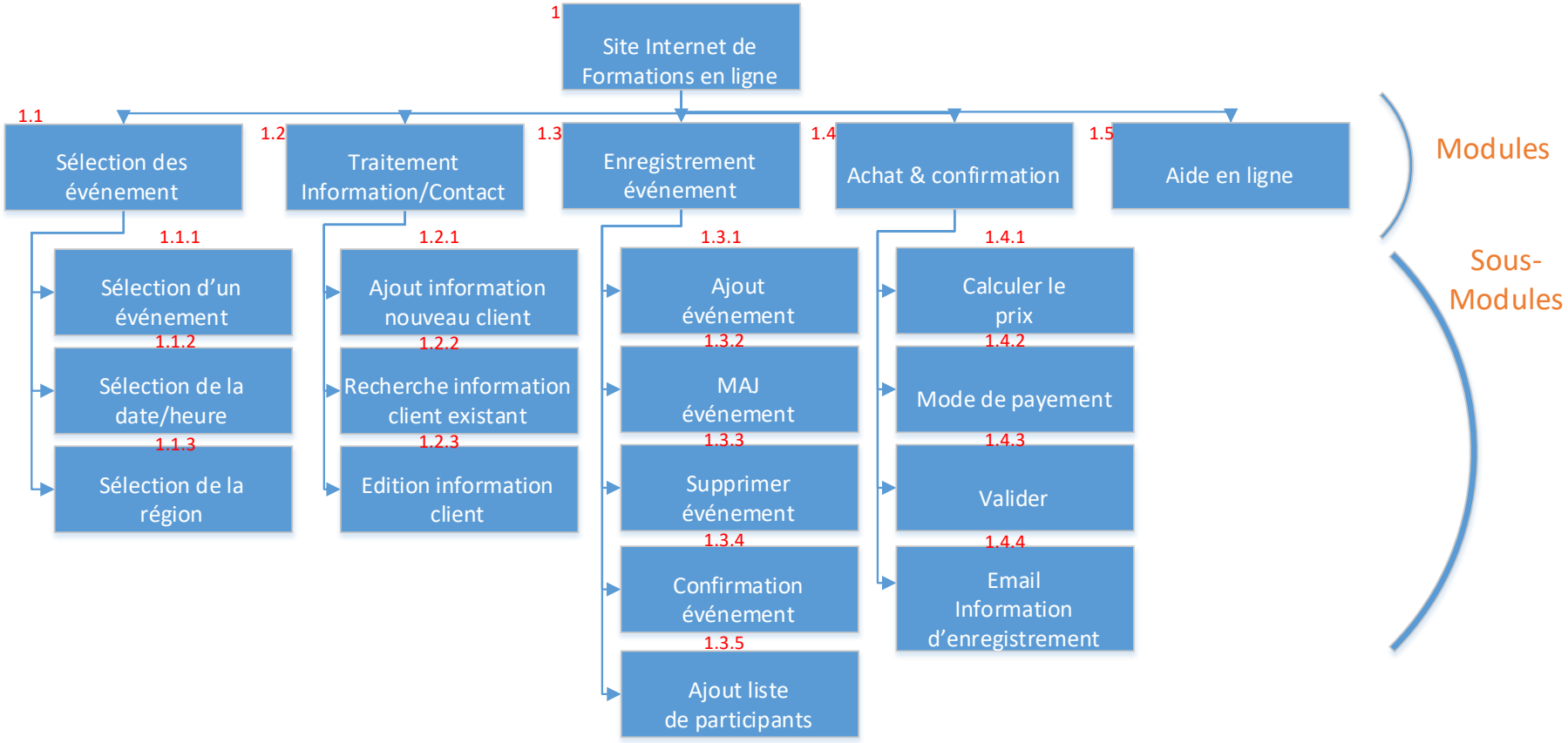
Etape 3: Grouper
- Grouper les post-it de façon naturelle en vue d'un groupement par composant

Etape 4: Documenter
- Documenter le PBS (ex: sur flip chart/PPT)

Etape 5: Vérifier
- Vérifier si le PBS est complet et ajouter des composants si nécessaire

Exemple d'outil qui peut-être utilisé : MIRO (collaboration, brainstorming, mind mappaing, Kanban…) → https://miro.com/

Le développement d'une bonne découpe de produit est plus efficace en utilisant une méthode interactive et itérative. La méthode des post-it en est un bon exemple.


1
Site Internet de Formations en ligne
1.1
Sélection des événement
1.2
Traitement Information/Contact
1.3
Enregistrement événement
1.4
Achat & confirmation
1.5
Aide en ligne
Modules
1.1.1
Sélection d'un événement
1.1.2
Sélection de la date/heure
1.1.3
Sélection de la région
1.2.1
Ajout information nouveau client
1.2.2
Recherche information client existant
1.2.3
Edition information client
1.3.1
Ajout événement
1.3.2
MAJ événement
1.3.3
Supprimer événement
1.3.4
Confirmation événement
1.3.5
Ajout liste de participants
1.4.1
Calculer le prix
1.4.2
Mode de payement
1.4.3
Valider
1.4.4
Email Information d'enregistrement
Sous-Modules

7.4. Baseline de qualité

Critères de qualité et méthode de mesure de qualité par lots de travail

- Critères de qualité:
  - Définir les spécifications de qualité auxquelles le produit devra répondre et quelles mesures seront appliquées par ceux qui inspecteront le produit final (Prince 2).
  - Il peut s'agir d'une simple référence vers un standard ou d'une explication exhaustive relative à ces points.
- Méthodes de mesure de la qualité:
  - Quel type de contrôle de qualité devra être utilisé pour contrôler la qualité ou les fonctionnalités du produit? Par exemple des tests, inspection, revue complète... (absence d'erreurs/d'inexactitudes, de disparités, routes sans issue...)
- Remarques :
  - Le manque d'une définition appropriée de la méthode de mesure de la qualité pourrait mener à des délais additionnels
  - Il faut inclure les critères de qualité et la méthode de mesure de la qualité dans la description des lots de travail

EXEMPLE	Critères de Qualité	Méthodes de Qualité
0. Gérer le projet		
1. Site internet de formations en ligne		
LT 1.1. Sélection de l'événement	**Les événements devront être facilement accessibles et triés par date** Après sélection, l'élément sélectionné s'affichera sur l'écran et le curseur passera automatiquement au champ de sélection de date et heure	**Tests unitaires sur le formulaire d'encodage et contrôle visuel**
LT 1.2 Sélection de la date et l'heure	**Les dates et heures devront être formatées au format européen** Après sélection, l'élément sélectionné s'affichera sur l'écran et le curseur passera automatiquement au champ de sélection de la région	**Tests unitaires sur le formulaire d'encodage et contrôle visuel**
LT 1.3 Sélection de la région	**Les régions devront être facilement accessibles et triés par ordre alphabétique** **Après sélection, l'élément sélectionné s'affichera sur l'écran**	**Tests unitaires sur le formulaire d'encodage et contrôle visuel**
2. Traitement informations/Contact		
. . .	. . .	. . .

7.5. Structure de découpage du projet – Structure de découpage du travail

7.5.1. Buts de ces structures

Décomposition hiérarchique, axée sur les livrables du travail.
La structure de découpage du projet (SDP) subdivise le travail en entités gérables.

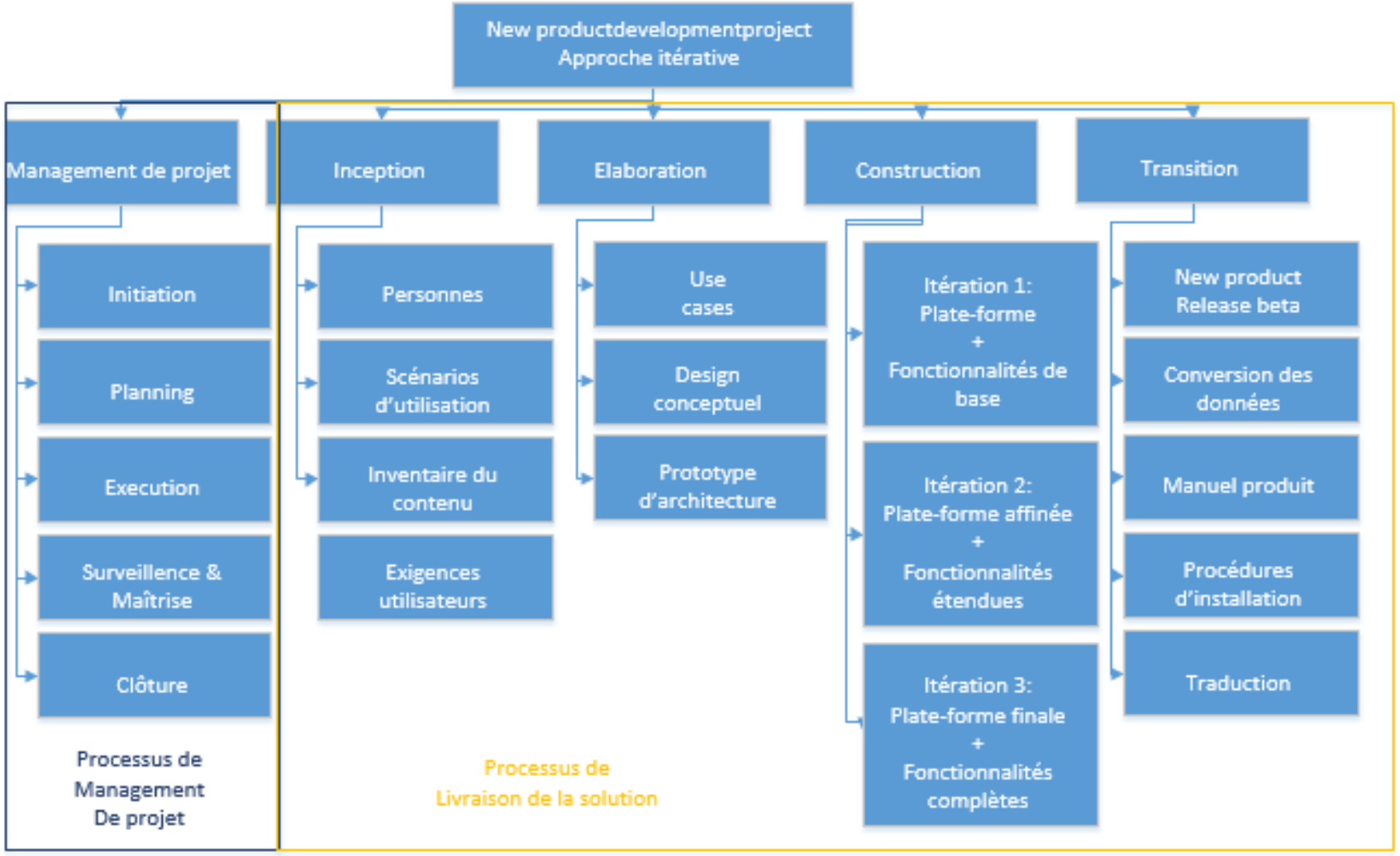
- Le SDP organise et définit le contenu complet du projet
- Pourquoi une SDP ?
    - Définir le contenu du travail et le cycle de vie du projet
    - Pour donner une base pour l'état d'avancement du projet et le rapportage de progression
    - Faciliter la communication entre le chef de projet et les parties prenantes (échéances, risques, performance, dépendances et budget)
    - Entrée principale pour les autres processus de management de projet et livrables
- Critères de regroupement dans la SDP
    - Processus projet (cycle de vie, cycle itératif...)
    - Résultat projet (sous-projets, livrables par spécialiste...)
    - Unité organisationnelle
- Le niveau le plus bas de la SDP sera utilisé pour
    - Les estimations des échéances et des coûts
    - La surveillance de l'avancement et du rapportage

Le SDP est donc un outil qui est utilisé pour définir le contenu du projet. Le PMBOK le définit comme tel: …Décomposition hiérarchique, axée sur les livrables que l'équipe doit exécuter pour atteindre les objectifs du projet et produire les livrables voulus. La SDP partage le projet en morceaux gérables, elle est le cœur du projet et tout ce qui n'y est pas repris est considéré comme hors contenu.
Il existe plusieurs façons de découper le travail projet comme, par exemple, par phase, par sous-projet, par unité organisationnelle. En général nous utiliserons la découpe par phases. Les unités les plus basses de découpe au niveau de la gestion du projet sont les lots de travail (PMBOK).
Les lots de travail sont donc les points de contrôle gestionnaire du projet. Les coûts à un moment donné et la progression du projet seront rapportés à ce niveau. Ce pourquoi le chef de projet gère le projet au niveau des lots de travail.

Remarque

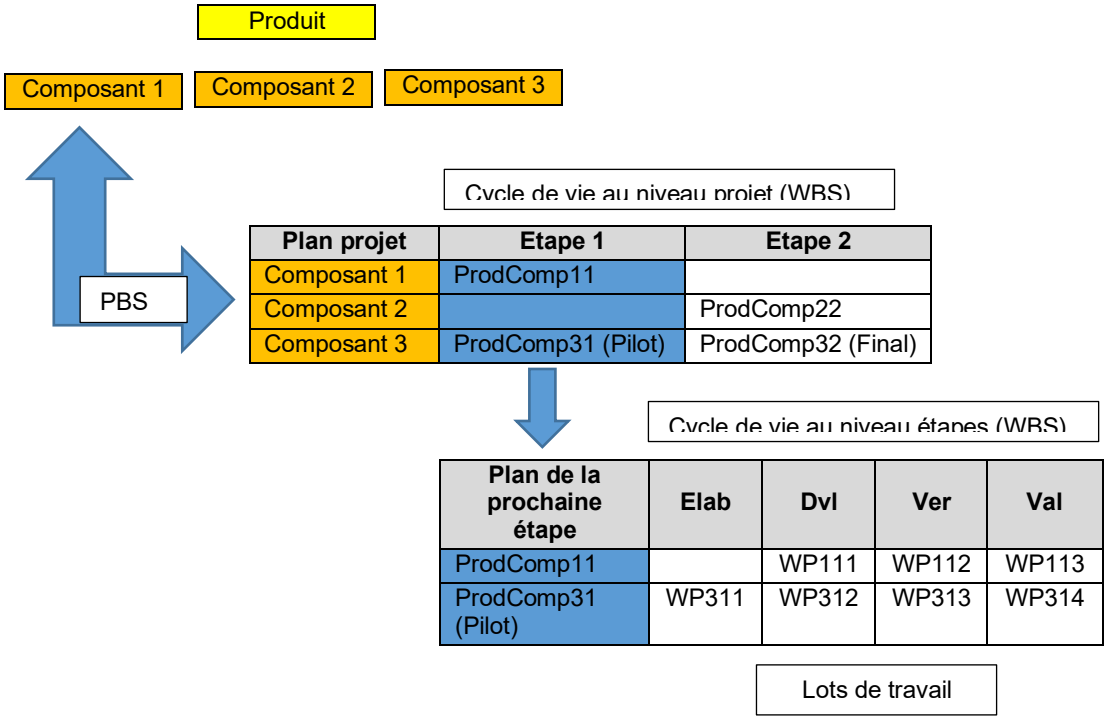
Les lots de travail seront découpés plus tard pour définir et planifier les activités de l'échéancier. Ceci sera réalisé par le propriétaire du lot de travail mais n'entrera donc pas en compte dans le cadre de la gestion du projet.

7.5.2. SDP: Exemple – SDP pour le développement d'un produit en utilisant une approche itérative



7.6. Relation entre la découpe du produit, le cycle de vie et la SDP

7.6.1. Explication



Un moyen de développer un bon SDP est d'implémenter une sorte de planification basé sur les produits :

- Développez un PBS qui définit les différents composants du produit qui devra être livré.
- Inclinez le PBS de 90° et formez une matrice avec les étapes du cycle de vie du projet
- Décrivez dans les cellules de la matrice quelles parties des composants seront livrées durant chaque phase du projet
- Vous pouvez maintenant lire le SDP de haut en bas.

7.6.2. Dictionnaire de la SDP

Faire pour chaque lot de travail une description

- Document qui décrit chaque composant de la Structure de découpage du projet
- Le dictionnaire de la SDP est joint à la SDP

Exemple de contenu du document:

- Propriétaire du LT
- Contenu du LT
    - Livrables
    - Activités
- Délai
    - Date de début et de fin
    - Jalons
- Coût
    - Exigences au niveau ressource
- Qualité
    - Critères de qualité
    - Méthode de qualité
- Dépendances
    - Output utilisé par...
    - Lots de travail éventuels qui doivent être terminés pour démarrer celui-ci
- Risques
    - Risques liés à ce lot de travail

7.7. Estimation coûts : méthode descendante

Le chef de projet commencera à effectuer une estimation avec une méthode descendante (Top-Down) basée sur la SDP

- Ceci sera la première amélioration de notre estimation après le démarrage
- Techniques
    - Jugements d'experts
    - Estimation par analogie : Utilisation d'informations historiques (projets précédents similaires)
    - Estimation paramétrique : par exemple, je développe en moyenne un formulaire en une journée, donc pour 10 formulaires je mettrai 10 jours
- Exemple: voir ci-dessous

	TEMPS-HOMME		MATERIEL
	Heures/homme	Euro	
0. Gérer le projet			
1. Site internet de formations en ligne			
LT 1.1. Sélection de l'événement	2	**160**	
LT 1.2 Sélection de la date et l'heure	2	**160**	
LT 1.3 Sélection de la région	**2**	**160**	
2. Traitement informations/Contact			
. . .	. . .	. . .	
TOTAL	6	480	

7.8. Estimation des coûts : Méthode ascendante

Le propriétaire du lot de travail définit les activités nécessaires pour fournir le lot de travail.

Pour cela il analyse le LT plus en détail:

- Quelles activités sont requises ?
- Quelle est la séquence des activités ?
- Quelle est la durée de chaque activité ?
- Qui peut effectuer l'activité ?
- Quel est le coût (par activité) ?

Ces estimations de LT sont additionnées en vue de déterminer le coût du projet (effort total). Il s'agit d'une méthode ascendante car elle est exécutée par le responsable d'un LT.

Exemple

LT 1.4: Sandwiches – Gestion caisse (jour/homme)	**6**
Formulaire ajout/retrait €	2
Module de comptabilisation dans la base de données (code sous un bouton 'enregistrer')	2
Mail automatisé de confirmation de versement/retrait vers le client et le gestionnaire	2

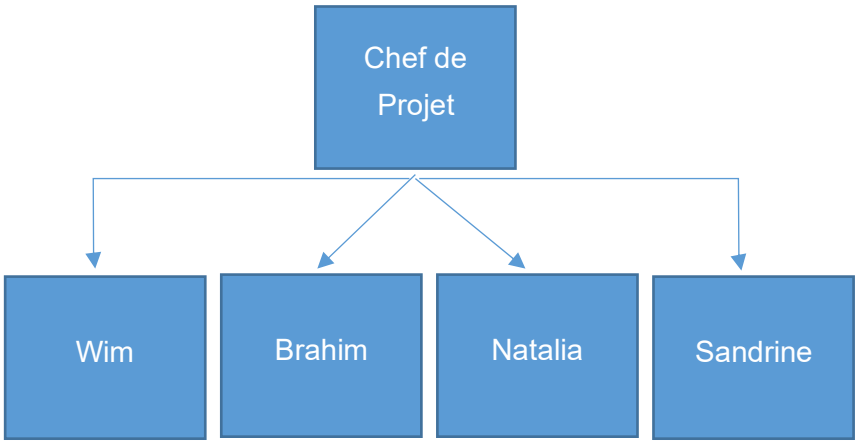
7.9. Equipe projet

4.10.1. Organizational Breakdown Structure

Développer un diagramme organisationnel du projet.

Structure de décomposition organisationnelle (= Organizational Breakdown Structure = OBS)

- Une description hiérarchique de l'organisation projet de telle façon que les lots de travail soient arrangés par unité organisationnelle (départements, unités, équipes)



7.9.1. Illustrer les connexions entre les LT (SDP) et l'équipe (OBS)

Diagramme d'affectation des responsabilités (Responsibility Assignment Matrix = RAM)

- Lier des activités à des ressources pour s'assurer que le contenu de chaque composant est assigné à un individu ou à une équipe.
- Ce type de diagramme peut exister à différents niveaux
- Format RACI :
    - Responsable (exécution): ce rôle va prendre en charge le travail.
    - Autorise : ce rôle approuve le travail terminé et est totalement responsable pour ce travail. Il ne peut y avoir qu'un A par activité.
    - Consulté : Le ou les **C** sont les participants qui doivent être **consultés**. Sur consultation du A et des R, ils **donnent leur avis sur les sujets où ils sont experts**. Les C **n'ont pas autorité**. C'est le **A qui décide de prendre en compte ou non l'avis** des C.
    - Informé : Les **I** sont les personnes qui **doivent être informées**. Elles sont classiquement indirectement **impactées par le projet**, comme les utilisateurs, les responsables de projets périphériques...

Exemple

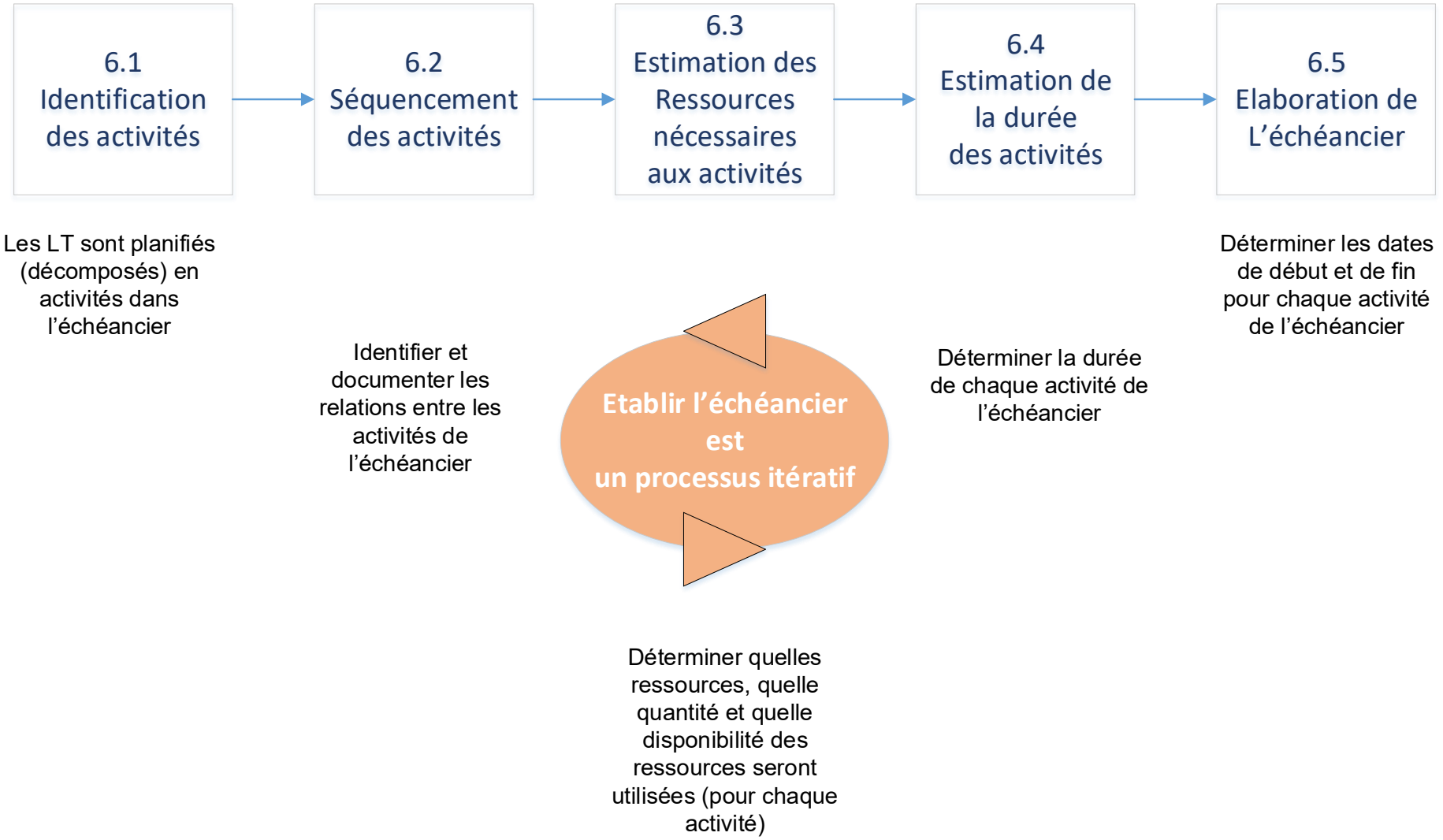
LT	Wim	Brahim	Natalia	Sandrine	John
Définit		R	I	I	A
Design	I		R	C	A
Développe	R		I	C	A
Teste		I	I	R	A

4 Lots de Travaux

Le modèle RACI est un exemple, parmi d'autres, de diagramme d'affectation des responsabilités

Un exemple au format .xlsx est donné via ce site → https://www.webstrat.fr/blog/raci-definir-roles-responsabilites-acteurs-projet (voir lien en bas de cette page WEB)

7.10. Processus d'établissement d'un échéancier détaillé



Gestion des délais du projet

Ce processus est de type itératif, ce qui signifie qu'il faut le parcourir plusieurs fois (dans les 2 sens) pour aboutir à un échéancier correct et réaliste. Souvent un outil de planification est utilisé pour réaliser ces opérations.

4.11.1. Les processus d'établissement de l'échéancier détaillé

1. Identification des activités: Identification des activités de l'échéancier qui doivent être réalisées pour produire les différents livrables du projet.
2. Séquencement des activités: Identification et documentation des dépendances entre les activités de l'échéancier.
3. Estimation des ressources nécessaires aux activités: estimation des types et des quantités de ressources nécessaires à l'exécution de chaque activité de l'échéancier.
4. Estimation de la durée des activités: Estimation du nombre de périodes de travail nécessaires pour réaliser complètement chacune des activités de l'échéancier.
5. Elaboration de l'échéancier: Analyse des séquences des activités et de leur durée, des ressources nécessaires et des contraintes en vue de créer l'échéancier du projet.

Ces différents processus sont expliqués dans les points qui suivent.

Processus 6.1: Identification des activités:
Les lots de travail (LT) existent au niveau le plus bas de la structure de découpage du projet. Pour rappel, c'est donc le niveau le plus bas auquel s'intéressera le chef de projet.
Cependant, le LT sera encore à un niveau de détail insuffisant que pour effectuer une estimation correcte du temps nécessaire à sa réalisation. Pour chacun de ceux-ci un responsable est désigné qui aura, entre autres, la tâche de décomposer les lots de travail en unités plus petites, chiffrables au niveau de l'effort nécessaire, les activités. Ce seront donc les activités qui seront nécessaires pour accomplir le LT.

Exemple:

Pour un module d'acquisition des données des clients il nous faudra:

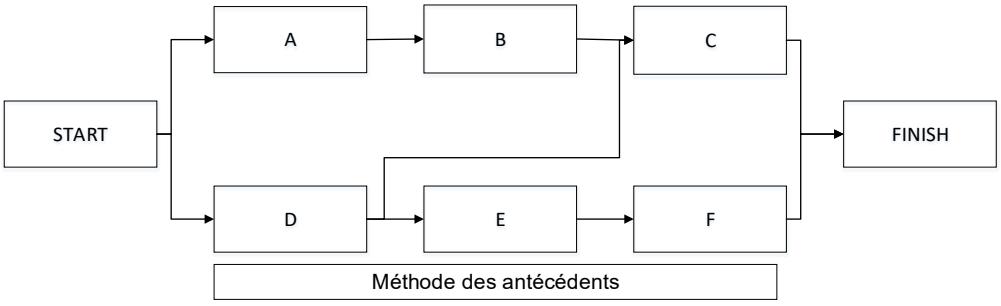
- Un formulaire 'Client' comprenant
    - des éléments de design (visuel)
    - des champs à remplir qui permettront la création, la modification, la suppression des informations concernant le client.
    - des vérifications automatisées sur les informations entrées dans ce formulaire
    - une interface avec une base de données qui devra contenir bien évidemment une table client
    - des interfaces avec d'autres systèmes éventuels (ex: lecteur e-ID/ lecteur de carte d'identité électronique...)
    - ...

Le niveau de détails (décomposition) sera déterminé par le responsable du LT. Il arrivera de cette manière à un niveau suffisamment bas que pour estimer le temps nécessaire pour réaliser chacune des activités.

Processus 6.2, 6.3, 6.4: Séquencement des activités, estimation des ressources et estimation de la durée

Dès que les activités de l'échéancier ont été définies, elles doivent être mises en séquence en tenant compte des dépendances. Pour réaliser ceci nous allons voir la méthode des antécédents est de loin la plus utilisée dans les outils de planification.

Illustration



Nous pouvons distinguer 4 types de dépendances:

- Liaison fin-début: le démarrage de l'activité successeur dépend de la fin de l'activité antécédente (le plus courant!)
  Exemple: les champs ne peuvent être créés sur le formulaire que si celui-ci a déjà été créé.

- Liaison fin-fin: l'achèvement de l'activité successeur dépend de celui de l'activité antécédente.
  Exemple: je ne peux terminer mon formulaire client que si ma base de données et mes tables clients ont été créées (et que ce travail est terminé)

- Liaison début-début: Le démarrage de l'activité successeur dépend de celui de l'activité antécédente
  Exemple: mesurer la performance d'un logiciel ne peut démarrer qu'après que les tests unitaires aient démarré

- Liaison début-fin : l'achèvement de l'activité successeur dépend du début de l'activité antécédente (rare!)
  Exemple: si la phase d'installation de mon serveur est décalée alors l'approvisionnement (livraison) de ce dernier peut être décalé également pour éviter le stockage inutile (coûts, garantie...)

Qualité des estimations des LT/activités

A. Décomposition

La technique de décomposition, telle qu'elle est appliquée à la définition des activités, consiste à subdiviser les lots de travail du projet en composants plus petits et plus faciles à gérer, appelés activités. Les activités constituent l'effort nécessaire à l'achèvement du lot de travail. Le processus *Définir les activités* définit les données de sortie finales comme étant des activités et non des livrables, comme cela est le cas pour le processus *Créer la structure de découpage du projet.* La liste d'activités, la SDP et le dictionnaire de la SDP peuvent être développés séquentiellement ou simultanément, la SDP et le dictionnaire de la SDP servant de base à l'établissement de la liste finale d'activités. Chaque lot de travail de la SDP est décomposé en activités nécessaires à la production des livrables du lot de travail. La participation des membres de l'équipe à cette décomposition peut permettre d'obtenir de meilleurs résultats, plus précis.

B. Planification par vagues

La planification par vagues est une forme de planification par élaboration progressive dans laquelle le travail prévu à court terme est planifié en détail, tandis que le travail à long terme est planifié à un niveau plus élevé de la SDP. Par conséquent, le travail peut être présenté de manière plus ou moins détaillée selon sa position dans le cycle de vie du projet. Par exemple, au cours de la période initiale de planification stratégique, alors que l'information est moins détaillée, les lots de travail peuvent n'être décomposés qu'au niveau du jalon. Une meilleure connaissance des événements qui vont se produire à court terme permettra une décomposition en activités.

C. Modèles

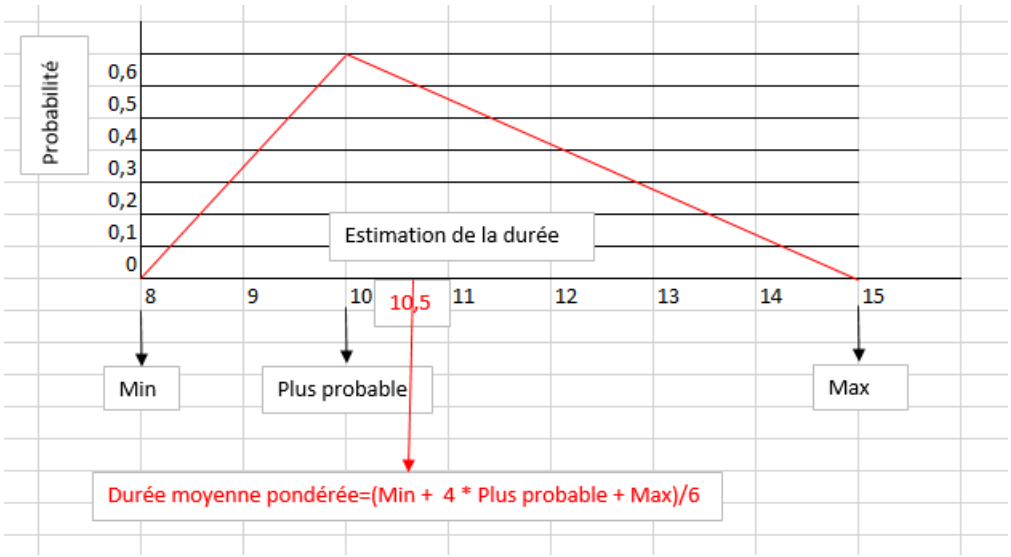
Une liste d'activités standards ou une partie de la liste d'activités provenant d'un projet antérieur, peut souvent servir de modèle à un nouveau projet. Dans les modèles, les informations correspondantes sur les attributs des activités peuvent également comprendre d'autres informations descriptives utiles à la définition des activités. Les modèles permettent également d'identifier les jalons typiques de l'échéancier.

D. Jugement d'expert

Lorsqu'ils possèdent l'expérience et la compétence nécessaire à l'élaboration des énoncés détaillés du contenu du projet, de la SDP et des échéanciers du projet, les membres de l'équipe de projet ou d'autres experts peuvent apporter leur expertise lors de la définition des activités.

Techniques d'estimations:

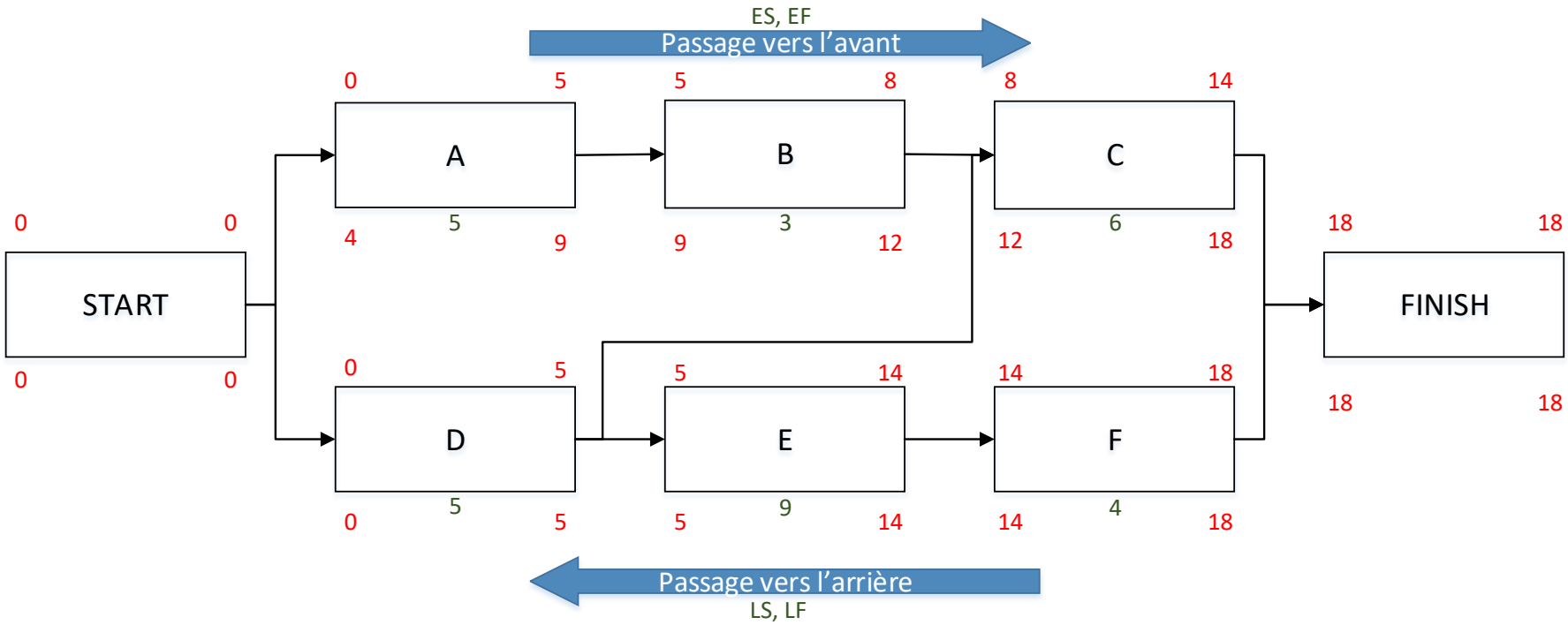
- Jugements d'experts
- Impliquez les personnes qui effectueront le travail
- Estimations par analogie (informations historiques)
- Estimations paramétriques
    - Quantité de travail * Productivité = durée
- Réserve de contingence
    - Eventualités, incertitudes
- Estimation à 3 points (PERT – voir ci-dessous))



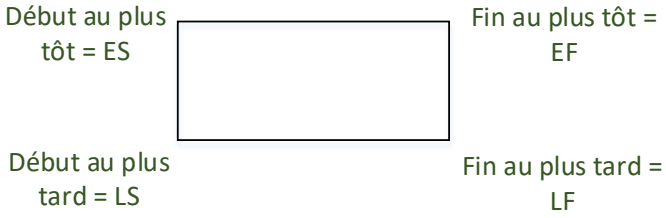
Processus 6.5: Elaboration de l'échéancier

L'élaboration de l'échéancier est un processus très répétitif. Le but est de définir une date de début et de fin à chaque activité, pour finalement déterminer la durée totale du projet. L'output majeur est la référence de base de l'échéancier (baseline).

Dans le cadre de ce cours nous nous limiterons à aborder l'analyse du diagramme de réseau avec la méthode du chemin critique (critical path).



Chemin critique



Explications du schéma ci-dessus

Questions:

- Quelle est la flexibilité de l'échéancier sur les différents chemins du réseau?
- Quelle est la durée minimale du projet?

Méthode:

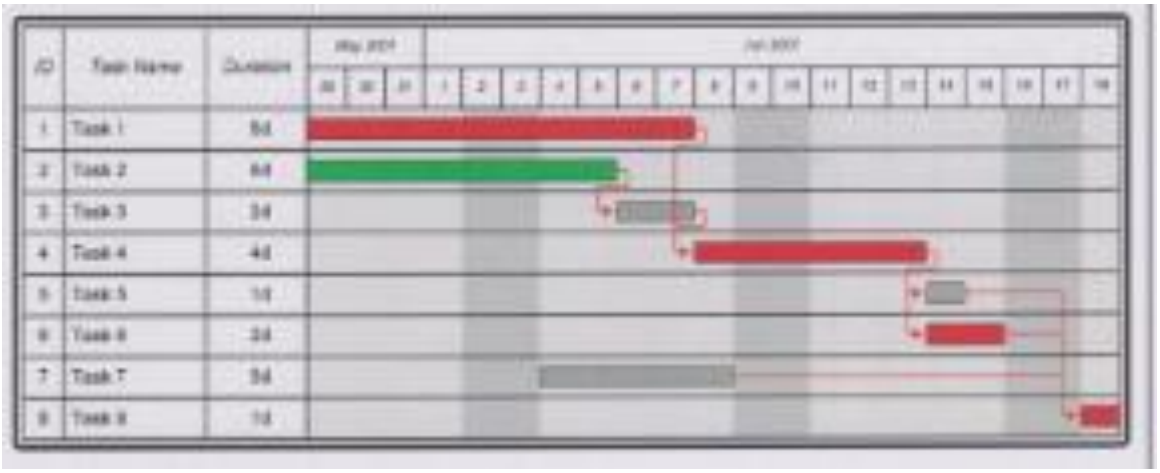
- Démarrer sur base d'un diagramme de réseau, utilisant un modèle d'échéancier.
- Documenter la durée pour chacune des activités (en vert sur le schéma)
- Faites un passage en avant pour chacune des branches en démarrant de la première où vous initialisez ES=0 pour la première activité (pour autant qu'il n'y ait d'antécédents avant cette première activité).
- Calculez ensuite EF de cette même activité en ajoutant simplement: EF=ES+NB jours activité (qui est repris en vert dans notre schéma ci-dessus). Pour l'activité A, par exemple, ceci nous donnera EF=0 + 5 = 5.
- Pour l'activité suivante nous initialisons le ES avec le EF (sauf si activité antécédente qui provoquerait une durée supérieur à EF). L'activité suivante sur cette même branche dans notre exemple est la B. Donc, puisque pas d'activité antécédente, ES = 5. On calcule EF: EF = 5 + 3 = 8. On continue ensuite la même opération pour l'activité C. Celle-ci a une activité antécédente, mais il faut un intervalle de temps de 5 pour la réaliser, celle-ci n'aura donc aucun impact sur le démarrage de l'activité C (puisque 5<8) et le processus reste donc le même: EF = 8 + 6 = 14. C'est la dernière activité de la branche, il faudra au total un intervalle de temps de 14 au minimum pour finir les activités de la branche A, B, C.
- Nous faisons ensuite la même opération avec la deuxième branche et nous obtenons un EF final égal à 18.
- La branche dont l'EF final est le plus élevé sera celui qui apparaîtra sur la case Finish.
- Si nous nous limitons à cette échéance (18 = durée du projet), alors cette branche sera le chemin critique, c'est-à-dire le chemin où nous avons une marge de 0 au niveau de la flexibilité, et donc le moindre problème au niveau d'une des activité dans la branche D,E,F engendrera un retard de l'échéance du projet.
- Pour la branche A, B, C, où nous avons une marge de manœuvre, nous pouvons faire un passage en arrière (de la droite vers la gauche en démarrant du Finish). Nous savons que la durée minimale du projet est de 18, donc LF de l'activité C est initialisée à 18. Ceci nous permet de calculer le LS → LS = 18 – 6 = 12. Ceci signifie que l'activité C devrait démarrer au plus tard à un intervalle de temps = 12! (donc on démarrera au plus tôt à 8 et au plus tard à 12 pour maintenir les échéances)

En répétant les mêmes opérations pour la branche A, B, C, nous constaterons que nous pourrions postposer la première activité de cette branche d'un intervalle de temps = 4, c'est la 'marge libre'.

Attention toutefois de ne pas faire de toutes les branches des chemins critiques car au moindre problème rencontré nous risquons d'avoir du retard sur nos échéances!

Illustration:

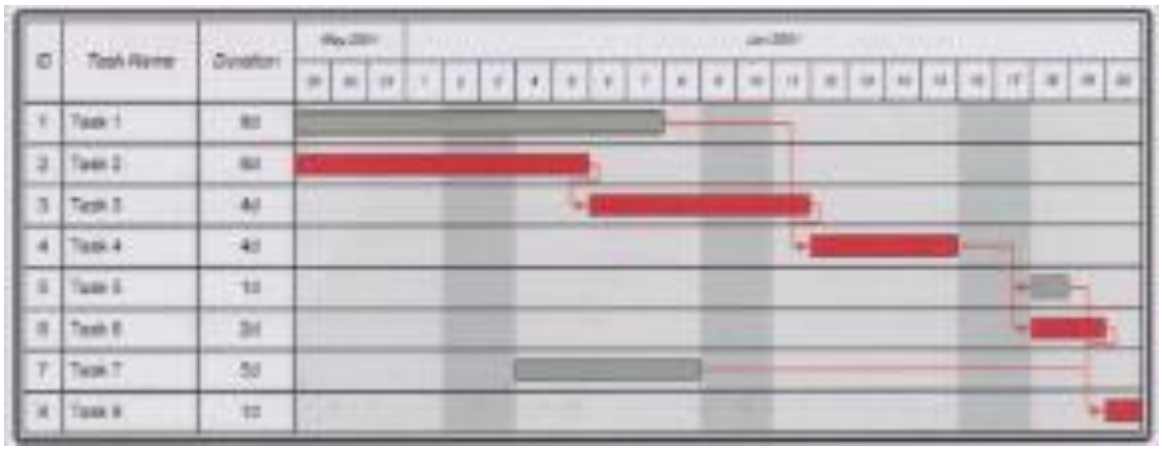
Le chemin critique est le chemin le plus long qui dicte la durée du projet (la marge est de 0 ou négative.



Les activités qui ne se trouvent pas sur le chemin critique ont une marge libre. Par exemple la tâche 2 a une marge de 2 jours

... et la tâche 3 a une marge de 2 jours

Supposons que l'activité 2 et l'activité 3 utilisent toutes 2 leur marge libre de 2 jours, ceci aura pour conséquence que la réalisation du projet sera retardée de 2 jours!



Ce qui se passe en ce cas, c'est que le chemin critique s'est déplacé...

La dernière étape dans ce processus 6.5 (Elaboration de l'échéancier) sera de le réviser et de voir comment gagner du temps, si nécessaire, pour diminuer la durée totale du projet ou prendre une marge sécurité au niveau des délais annoncés.

Conseils pour réduire la durée totale:

- Exécution des activités en parallèle
- Allouer plus de ressources
- Séparer les activités en différentes parties pour qu'elles puissent être exécutées en parallèle par plusieurs ressources
- Réallouer les ressources sur des activités critiques plutôt que sur des activités non critiques
- Permettre les heures supplémentaires, travailler week-end et/ou jours fériés (avec compensation pour les ressources bien entendu!!)
- Réduire la qualité, les ambitions ou le contenu du projet

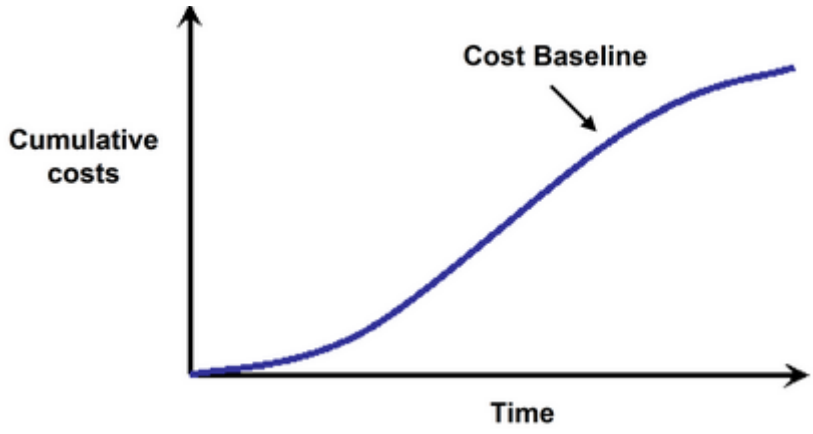
7.10.1. Exercice

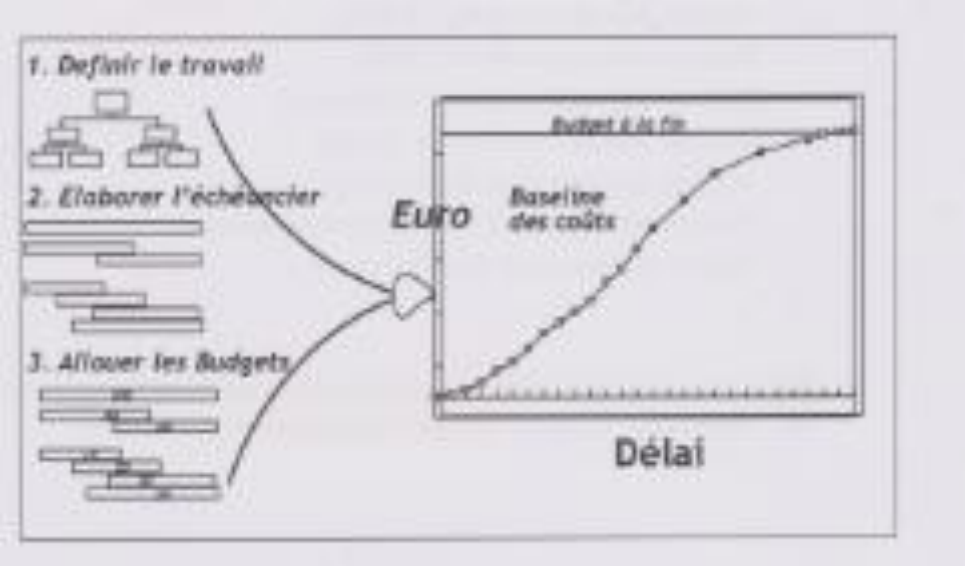
Créer un échéancier pour le Ticketing system (par exemple avec l'outil 'OpenProj' qui peut être téléchargé gratuitement sur le site:
http://www.commentcamarche.net/download/telecharger-34056854-openproj)

7.11. Budgétisation des coûts – Baseline des coûts

Le budget sera étalé dans le temps, ceci est une nécessité pour mesurer la performance du projet.

- La ligne de référence des coûts (cost baseline) doit nous donner une vue claire sur ce qui sera dépensé et quand
- L'état du budget (coût actuel vs coût planifié) devra pouvoir être mesuré à n'importe quel moment du projet
- C'est l'addition des coûts estimés par période





Un processus de planning est un processus fortement itératif.

Le processus est principalement composé des étapes suivantes:

- Définir le contenu du travail (CONTENU): créer la structure de découpage du projet avec les lots de travail (LT) au niveau le plus bas
- Définir l'échéancier du projet (TEMPS): élaborer l'échéancier des activités qui doivent être exécutées pour livrer les LT
- Budgéter le travail (COÛTS): agréger les estimations des coûts d'activités individuelles ou des LT afin de fixer une référence de base des coûts.

8. Gestion des risques du projet

8.1. Liste des processus liés à la gestion des risques

La gestion des risques du projet est un groupe de processus dans le PMBOK qui comprend **6 processus**:

- Planification de la gestion des risques (11.1): Décider comment approcher, planifier et exécuter les activités de management des risques du projet
- Identification des risques (11.2): déterminer quels risques pourraient avoir un impact sur le projet et documenter leurs caractéristiques
- Analyse qualitative des risques (11.3): Hiérarchiser les risques pour une analyse ou une action ultérieure en évaluant et en combinant leur probabilité et leur impact
- Analyse quantitative des risques (11.4): Effectuer l'analyse chiffrée des effets des risques identifiés sur l'ensemble des objectifs du projet
- Planification des réponses aux risques (11.5): Etablir des procédures et des méthodes destinées à augmenter les opportunités favorables aux objectifs du projet et à réduire les menaces à leur encontre
- Surveillance et maîtrise des risques (11.6): Suivre les risques identifiés, surveiller les risques résiduels, identifier les risques nouveaux, exécuter les plans de réponses aux risques et évaluer leur efficacité au long du cycle de vie du projet

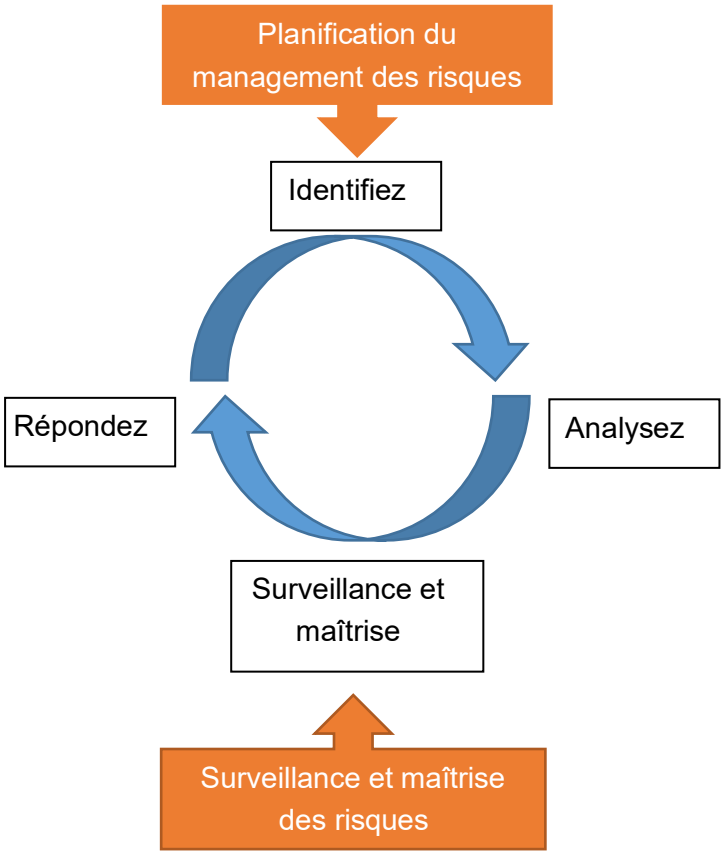
8.2. Définition d'un risque (PMBOK)

"Un risque est un événement ou une situation dont la concrétisation, incertaine, aurait un impact positif ou négatif sur les objectifs du projet, comme les délais, le coût, le contenu ou la qualité"

Que pouvons-nous déduire de cette définition?

- Le risque peut être une menace ou une opportunité. Par exemple, l'engagement et l'intégration d'un nouveau membre dans l'équipe de développement est une incertitude
- Il peut y avoir une ou plusieurs causes et un ou plusieurs impacts possibles (délai, coût, contenu, qualité)
- Nous ne sommes pas certains que cela se passera

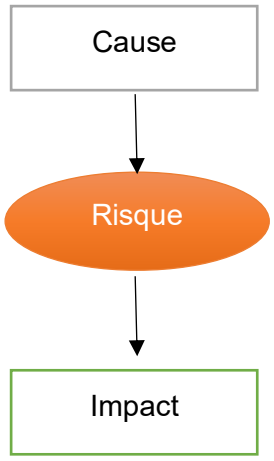
Un exemple classique de projet à risques est une construction immobilière où il est difficile de prévoir les délais pour l'obtention des permis nécessaires à la construction, la bonne coordination entre les différents corps de métier, la météo qui peut influencer la bonne exécution des travaux...



8.3. La gestion des risques

Puisque tout projet construit et livre des produits/services uniques, des risques y sont inhérents. Nous pouvons en déduire que 'Le risque zéro n'est pas une option!'
Sans prise de risque on ne peut pas progresser. Le chef de projet doit connaître et analyser les risques afin d'isoler les risques critiques et de prendre des actions en tenant compte des coûts pour atténuer ces risques.

La description d'un risque doit être précise et non ambigüe

	Parce que nous n'avons jamais travaillé avec ce fournisseur Il pourrait ne pas nous donner un service optimal et nous délivrer les ressources nécessaires quand nous en aurons besoin Ce qui pourrait nous mener à des retards dans notre projet

Pour être certain que tous les risques soient interprétés de la même façon, il faut appliquer une approche méthodique pour décrire les risques. Chaque risque devrait être identifié en décrivant la cause et l'impact éventuel. De cette façon nous pouvons éventuellement avoir une cause qui implique plusieurs risques et un seul risque qui peut causer plusieurs impacts.

Un exemple de risque

Parce que le projet ne peut pas être développé en étroite collaboration avec le client final,

Le résultat pourrait ne pas correspondre aux attentes des utilisateurs,

Ce qui pourrait causer la non-acceptation du produit.

8.4. Exercice

Risque ou pas ?

- Nous n'avons jamais fait ce type de projet auparavant
- Les spécifications techniques seront livrées une semaine en retard
- Le client pourrait ne pas avoir fourni les vraies exigences de l'utilisateur
- La nouvelle ressource pourrait ne pas avoir les compétences requises
- Le management est très occupé
- Le management pourrait ne pas avoir le temps de participer au comité de pilotage
- Il y aura une fuite dans le toit de la maison

8.5. Plan de gestion des risques

Il s'agit du premier processus du management des risques

Exemples de points de réflexion pour un plan générique :

- Quelle est notre tolérance aux risques ?
- Qui est responsable de l'exécution de la gestion des risques ?
- Quelle approche allons-nous avoir ?
- Quand allons-nous mettre à jour les risques ?
- Quelle technique allons-nous utiliser pour analyser les risques ?
- Qui va rapporter et à qui ?
- ...

Le plan de management des risques est un plan subsidiaire qui fait partie du plan général du projet. Il décrit comment le management des risques sera effectué durant le projet.

Dans les sociétés à maturité élevée, un processus de management des risques (avec plan et *templates* nécessaires) existe au niveau organisationnel. Le chef de projet peut, lors du démarrage du projet, adapter ce plan de management des risques aux besoins de son propre projet.

8.6. Identification des risques

Il s'agit du second processus du management des risques

8.6.1. Introduction

Comment découvrir les risques ?

Nous devons parler avec les parties prenantes (interview, brainstorming...) et revoir les documents projets.

Les analyses à réaliser:

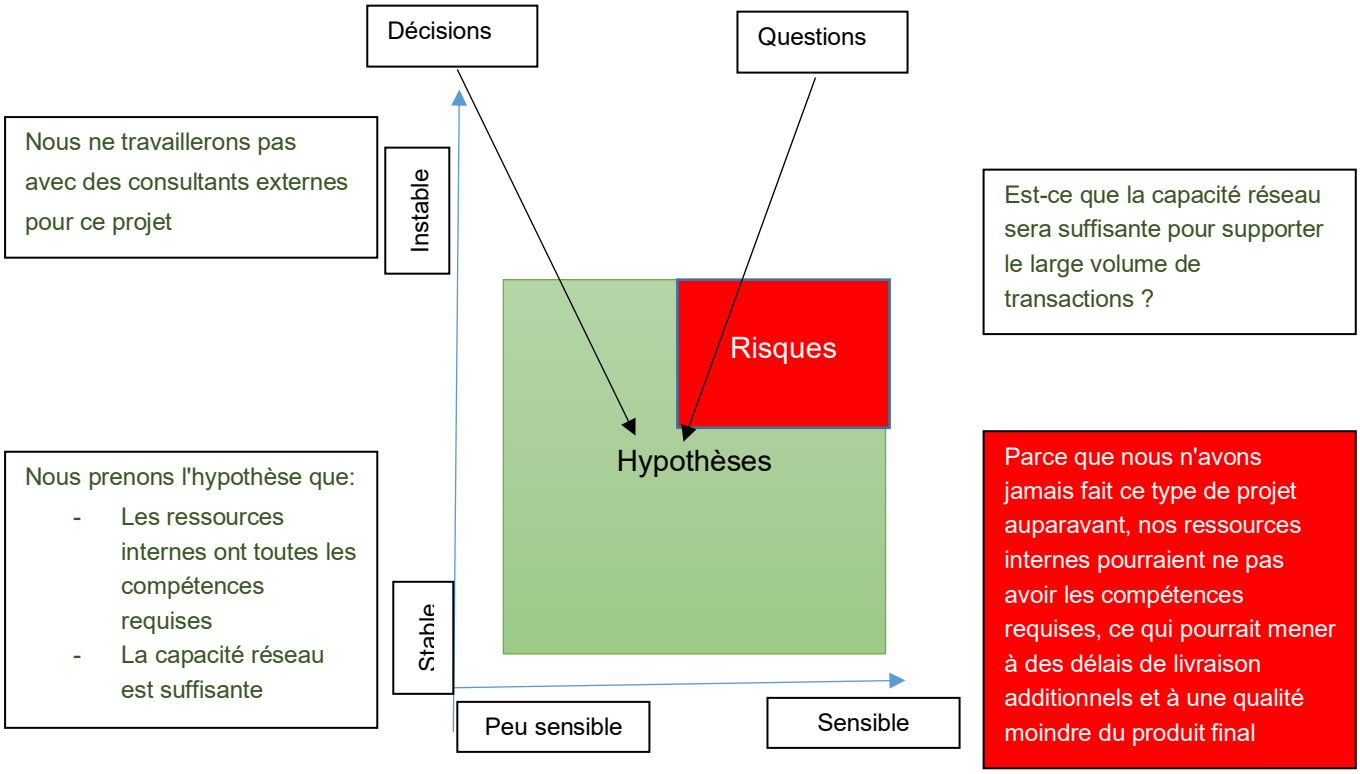
- Problèmes majeurs
  - Les questions auxquelles nous n'avons pas encore de réponse
- Décisions
  - Sélection d'une alternative parmi d'autres sans avoir toutes les informations en main
- Hypothèses
  - Des choses que nous supposons vraies pour que le projet soit une réussite
- Dépendances
  - Autres lots de travail/Projet/Fournisseurs : par exemple quelqu'un doit nous fournir quelque chose ou nous devons fournir quelque chose à quelqu'un
- Parties prenantes clefs

Il est important d'identifier, dès que possible, les incertitudes dans le projet et de les analyser. Ceci permettra d'identifier les risques critiques et de déterminer ce qu'il faut faire pour s'assurer que l'impact reste le moins important possible. Pour l'identification des risques nous allons analyser quelles hypothèses nous faisons, quelles décisions sont prises et quels sont les problèmes majeurs.

De nombreuses check-lists existent pour aider les chefs de projets à identifier les risques de leur projet. Exemple : https://espaces-numeriques.org/wp-content/uploads/2019/01/I22p20.pdf (simple avec de bons points de réflexion !). De nombreuses autres sources peuvent être trouvées sur Internet!

Identification des risques par l'analyse des hypothèses: exemple

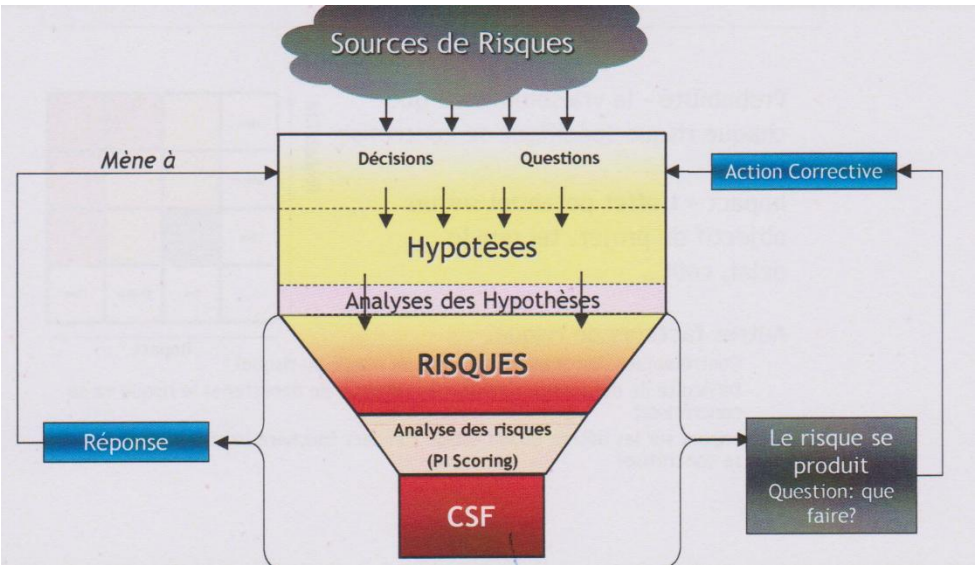
8.6.2. Analyse des hypothèses



Une des techniques pour identifier les risques s'appelle l'analyse des hypothèses. Des décisions et des problèmes majeurs sont convertis en hypothèses. Ces hypothèses sont ensuite mesurées au point de vue stabilité (l'hypothèse est-elle stable ou ne va pas tenir?) et au point de vue sensibilité (dans quelle mesure notre projet est sensible au fait que l'hypothèse soit vraie ou fausse?).

Dans l'exemple ci-dessus, quand nous prenons la décision de ne pas travailler avec des ressources externes, nous partons de l'hypothèse que nous aurons toutes les compétences nécessaires pour réaliser le projet en interne. Quand nous connaissons, par exemple, les techniciens qui doivent assurer le travail ainsi que leur degré de connaissance de la nouvelle technologie, nous pourrions conclure que ce n'est pas une hypothèse stable (via leur historique par exemple). En plus, cette technologie peut avoir une grande importance pour notre projet. De là, disposer de cette compétence ou ne pas en disposer, fera la différence pour notre projet.
Nous avons donc à faire une hypothèse qui est instable et qui, en plus, est sensible pour notre projet. Elle représente donc un risque important pour notre projet.

8.6.3. Le silo des risques



Des sources de risques, il y en a beaucoup comme par exemple: peu de disponibilité du management, définition vague des rôles, utilisation de nouvelles technologies...

Certaines questions sont résolues par la prise de décision, d'autres restent ouvertes et évoluent vers des problèmes majeurs. Décisions et problèmes majeurs mènent implicitement à des hypothèses (nous supposons que ces éléments sont réels pour pouvoir avancer).

Bien que les hypothèses soient des incertitudes, et donc des risques, elles ne seront pas toutes enregistrées comme risque. Par la technique de l'analyse des hypothèses (sensibilité, stabilité), les risques réels sont identifiés et documentés.

Ensuite, en utilisant l'analyse des risques (p.e. PI score), les risques sont analysés et l'importance en est déterminée (faible, moyen, important). Le top-3 des risques avec l'importance la plus élevée, sont les facteurs de succès critiques (CSF = Critical Factor Success).

Dans le schéma ci-dessus nous voyons également:

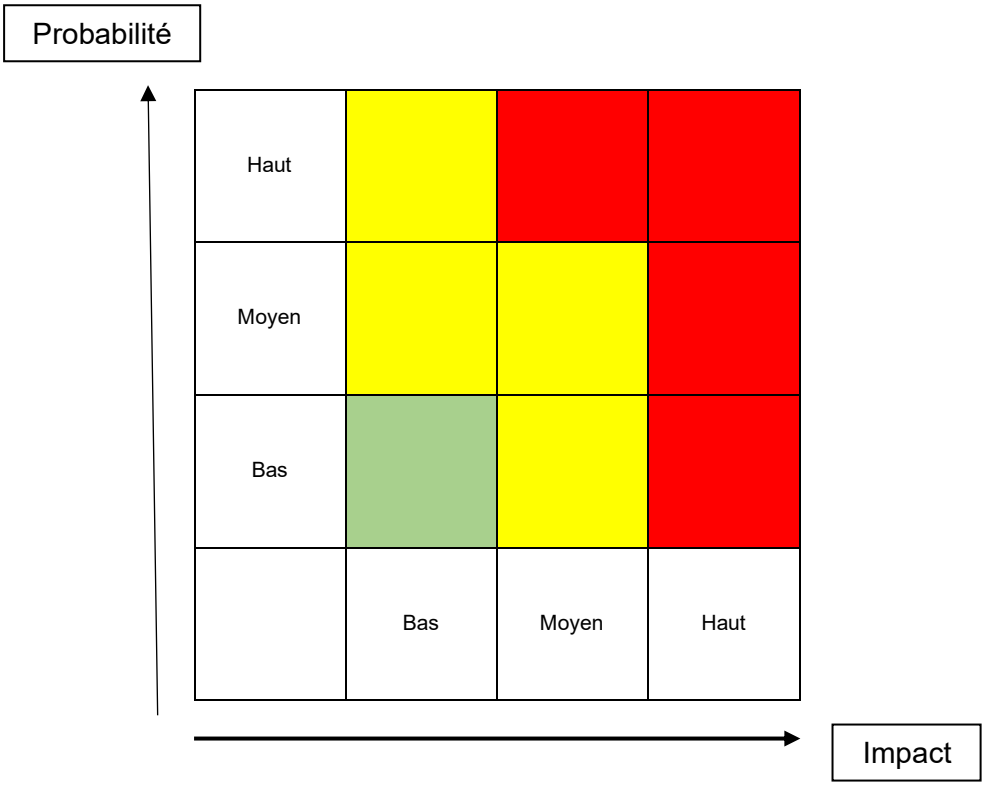
- Les réponses aux risques sont développées pour mitiger les risques, mais les réponses aux questions peuvent, à leur tour), entraîner des décisions, des problèmes majeurs et donc des hypothèses
- Quand un risque se produit, des actions correctives (ou de contingences=plan B) doivent être mises en place. A leur tour des actions correctives peuvent entraîner d'autres nouveaux risques.

8.7. Analyse qualitative des risques

Il s'agit du troisième processus de management des risques.

8.7.1. Introduction

L'importance du risque = f(impact, probabilité)



- La probabilité = vraisemblance qu'un risque spécifique se concrétise
- Impact = effet potentiel sur un objectif du projet tel que le délai, le coût, le contenu...

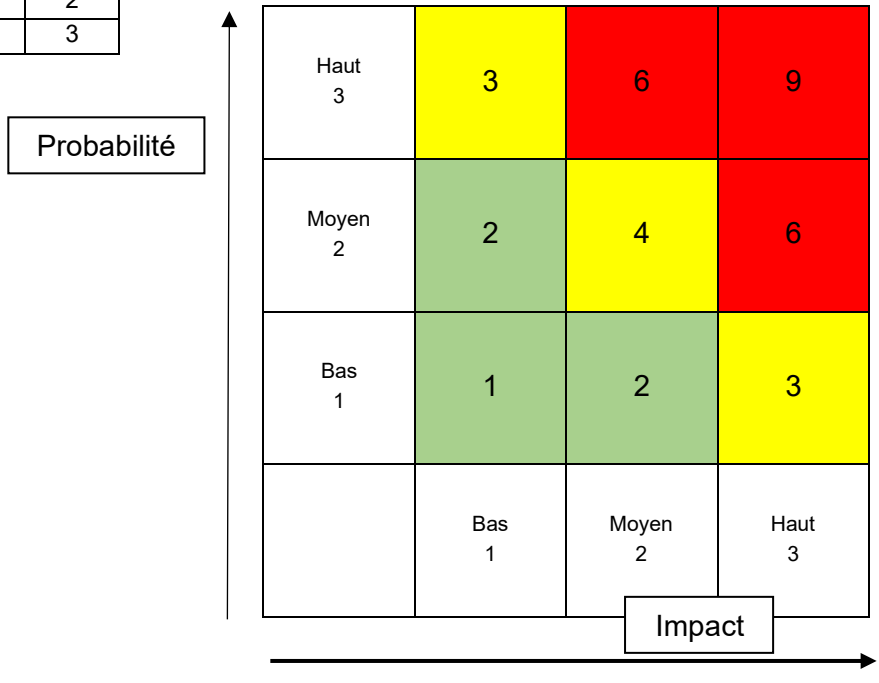
Outre ces deux facteurs, d'autres interviennent également comme:

- La contrôlabilité: dans quelle mesure pouvons-nous intervenir dès que le risque se produit?
    - Exemple:
        - Une personne se trouve sur des rails de train, le train arrive, quand il est à 20 mètres il saute sur le côté
        - Même topo mais dans un tunnel... → même impact, même probabilité mais différence dans la contrôlabilité.
- Difficulté de détection: dans quelle mesure est-il facile de détecter si un risque s'est produit ou pas?
    - Exemple:
        - Il est beaucoup plus difficile de détecter que les attentes du client ont été mal définies que de constater que quelqu'un est malade (et indisponible)
- Délai d'impact: quand le risque va-t-il impacter notre projet s'il se produit?
    - Exemple:
        - Les attentes utilisateur mal définies ne deviendront visibles que vers la fin du projet

8.7.2. La matrice probabilité/Impact

Une façon qualitative de prioriser les risques.

Probabilité		Impact	
LOW	1	LOW	1
MED	2	MED	2
HIGH	3	HIGH	3

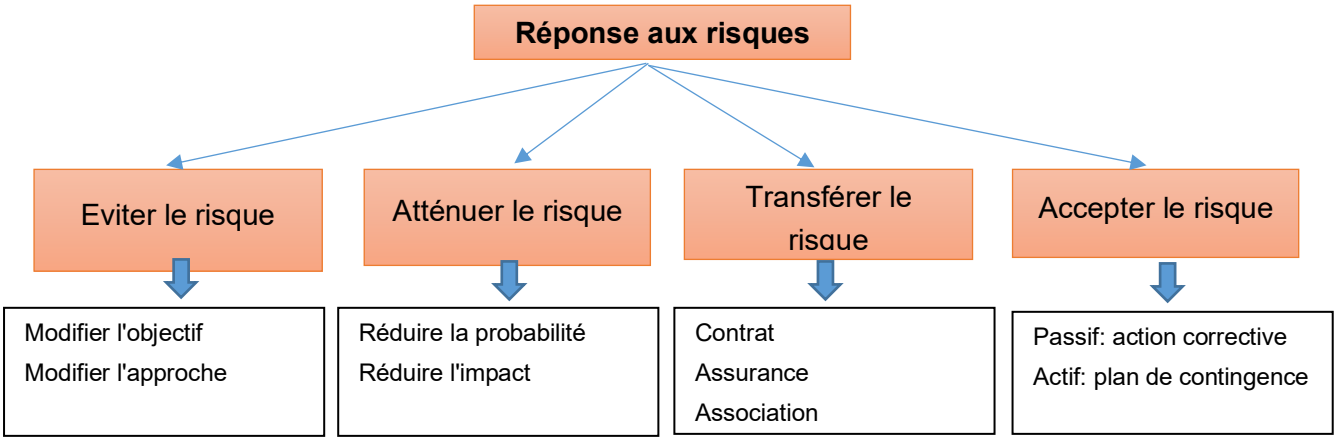


Interprétation des résultats de la matrice représentée ci-dessus:

- Comme mentionné ci-dessus: (Probabilité x Impact) = Sévérité (Criticité) du risque.
- Nous estimons par nous-même les paramètres Probabilité et impact:
  - soit estimation après avoir discuté avec l'ensemble des membres concernés par le projet et ce point en particulier
  - soit via des standards de l'entreprise si elle a une certaine maturité au niveau de la gestion de projet en ce domaine.
- Nous calculons ensuite l'indice de criticité du risque (P x I) et nous déterminons si ce risque est faible, critique ou inacceptable. Pour cela, nous déterminons un seuil de l'indice de criticité au-delà duquel le risque devient critique et un autre au-delà duquel le risque est inacceptable. Comme nous pouvons le voir ci-dessus: si l'indice <= 2 alors le risque est acceptable, entre >2 et <6 il est critique, et s'il est >=6 il est inacceptable.
- Ceci est réalisé pour chaque risque identifié, et ensuite nous devrons prendre les mesures nécessaires pour pallier aux risques critiques et inacceptables.
- Généralement nous ne nous soucierons pas des risques faibles, nous surveillerons ou prendrons action pour les risques critiques et traiterons en priorité bien sûr les risques inacceptables.
- Cette matrice peut être représentée pour chacune des contraintes du projet, à savoir:
  - impact sur le coût: quelle sera l'augmentation de coût si le risque se produisait?
  - le délai: de combien de temps le projet pourrait-il être retardé si le risque se produisait?
  - le contenu: le produit final répondra-t-il encore aux attentes du client et sera-t-il utilisable?

8.8. Planning des réponses aux risques

Il s'agit du quatrième processus de management des risques.



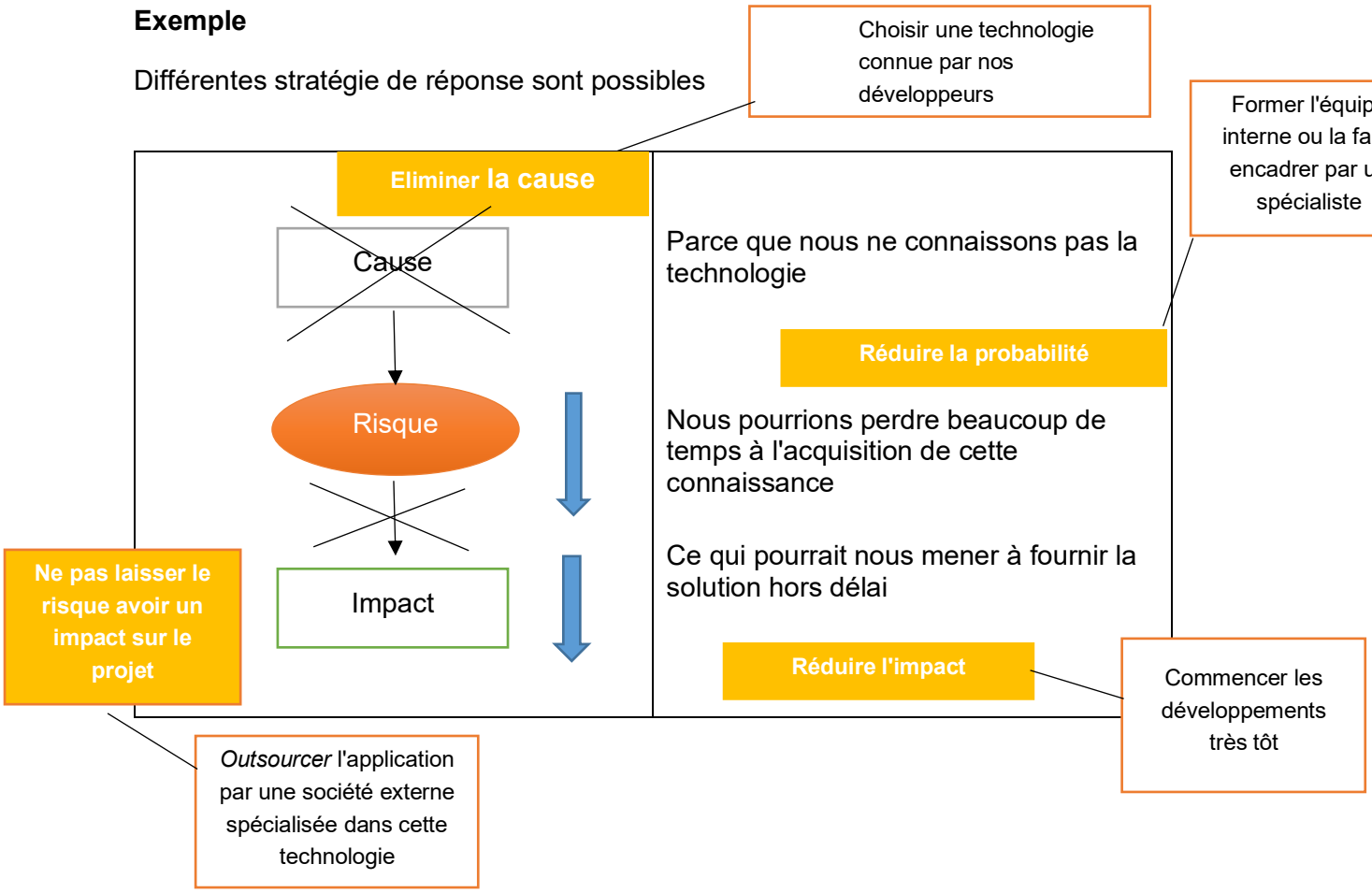
Des réponses aux risques ou des contremesures doivent être prises, alignées avec les priorités définies. Prendre des mesures a un coût, il est donc important de veiller au coût qui doit être en balance avec l'importance du risque (le coût ne peut pas être supérieur au coût qui serait engendré si nous ne faisions rien et que le risque se produisait).

Il existe différentes stratégie de réponse aux risques:
Eviter: Le plan de projet est modifié de telle façon que le risque est éliminé, il ne pourra donc plus se produire. Nous pouvons, par exemple, utiliser une technologie un peu plus ancienne pour éviter le manque de ressources ayant des compétences concernant la nouvelle technologie.
Transférer: L'impact du risque est transféré vers une tiers instance. Si notre fournisseur n'a pas les compétences nécessaires pour livrer un produit spécifique, il peut faire appel à un sous-traitant spécialisé en la matière (avec l'accord du client bien entendu).
Atténuer: Des mesures sont prises pour diminuer la probabilité ou l'impact du risque. Nous pouvons, par exemple, si nous n'avons pas les connaissances dans une technologie, donner des formations ou faire appel à un spécialiste pour les développements (coach des membres de l'équipe)
Accepter: Le risque est accepté, stipulant qu'aucune mesure ne vaut la peine d'être prise. L'acceptation peut être passive où nous ne faisons rien ou active où nous acceptons le risque mais nous veillons à avoir un budget/du temps (de réserve) ou nous prévoyons un plan de contingence séparé.

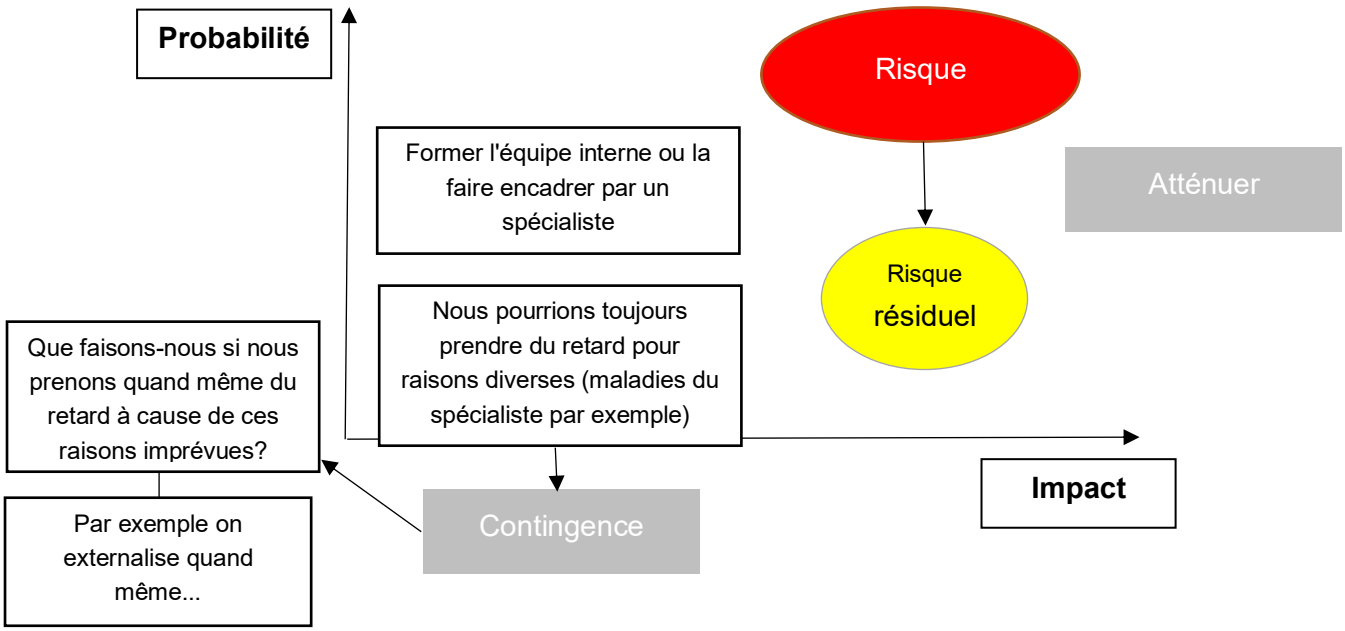
Pour chaque risque, son suivi et l'exécution des contremesures, il faut dédier un responsable. Tous ces risques sont documentés dans un risque-log qui est en permanence mis à jour (concept d'élaboration progressive!)

Exemple

Différentes stratégie de réponse sont possibles



Atténuation et contingence



8.9. Analyse quantitative des risques

Il s'agit du cinquième processus de management des risques.

Dans certains cas, dont les études d'impact, il peut être nécessaire de compléter cette démarche par une approche quantitative visant à estimer le niveau de risque(chiffrer) des situations mises en lumière.

Cette notion ne sera pas vue dans le cadre de ce cours, ce qu'il faut simplement savoir par rapport à l'analyse qualitative:

- Elle est plus précise et moins subjective mais plus complexe,
- Elle nécessite la connaissance détaillée de tous les paramètres d'évaluation
- Elle fait appel à des calculs statistiques

8.10. Surveillance et maîtrise des risques

Il s'agit du sixième et dernier processus de management des risques.

La probabilité et l'impact des risques peuvent changer dans le temps:

- Evaluez chaque risque identifié régulièrement pour pouvoir décider s'il est devenu plus ou moins probable ou si les effets du risque ont changé (mise à jour du registre des risques de façon régulière)
- Evaluez l'efficacité probable des actions sélectionnées
- Chaque risque devrait être discuté durant les réunions d'avancement avec le management (Liste Top 3 des risques)

9. Management des communications du projet

La communication proactive est importante dans tous les projets. La communication est un moyen vital afin de gérer les attentes des parties prenantes durant le projet.

9.1. Management des communications dans PMBOK

Quand nous parlons de management des communications dans un projet, nous parlons du domaine de connaissance qui emploie les processus nécessaires pour assurer, en temps voulu et de façon appropriée, la génération, la collecte, la diffusion, le stockage, la récupération et le traitement final des informations du projet.

Les processus:

- Planification des communications (10.1): ce processus détermine les besoins d'information et de communication des parties prenantes du projet
- Diffusion de l'information (10.2): ce processus permet de mettre l'information nécessaire à la disposition des parties prenantes du projet en temps voulu
- Etablissement du rapport d'avancement (10.3): ce processus permet de collecter et de diffuser l'information sur la performance. Ceci englobe les rapports d'état, la mesure de l'avancement et les prévisions
- Management des parties prenantes: ce processus concerne le management des communications afin de satisfaire les exigences des parties prenantes du projet et de résoudre les problèmes majeurs avec elles.

Nous n'entrerons pas dans les détails pour ce chapitre, les lignes qui suivent sont données à titre purement informatif.

9.2. Plan de management des communications

Le plan devrait comprendre les éléments suivants:

- Informations à communiquer (quelles informations communiquer aux parties prenantes?)
- But: la raison de la distribution de l'information (ex: rapporter l'avancement du projet au Comité de pilotage)
- Fréquence: de distribution de l'information (ex: toutes les deux semaines)
- Dates de début et de fin: le délai durant lequel l'information sera distribuée (ex: toutes les deux semaines)
- Format/Moyen: un fichier PPT par exemple
- Responsabilité: qui distribuera l'information?

9.3. Management des parties prenantes

Pour rappel, les parties prenantes sont toutes les personnes ou groupe de personnes qui:

- Collaborent au projet ou
- Ont un pouvoir de décision sur l'avancement du projet ou
- Subissent les conséquences du résultat du projet

Manager les parties prenantes signifie les identifier et gérer leur niveau d'engagement dans le projet (quelles sont les parties prenantes 'clef'?)
Le temps à allouer à cette partie du projet n'est pas à négliger et dépendra de la taille, de la complexité du projet et de la résistance que pourraient avoir certaines parties prenantes face au projet (face au changement!). Les parties prenantes doivent être impliquées le plus tôt possible dans le projet afin d'obtenir, dans la mesure du possible, un engagement collectif. Nous aurons, comme dans tout projet, des alliés et des ennemis (les deux positions extrêmes) au projet, une grille peut être établie à ce niveau. Il faudra analyser ceci et déterminer leur 'pouvoir'. Concernant les opposants il faudra essayer de les motiver en leur expliquant la plus-value que pourrait avoir le produit final pour l'organisation et pour eux.

Pour obtenir l'adhésion des parties prenantes, il faudra les impliquer, les responsabiliser et les tenir régulièrement informé sur l'état d'avancement du projet.
Les fonctions attribuées à chacune d'elle devront être clairement établies et non équivoques.

10. Exécution du projet

Durant le processus d'exécution du projet il faudra "Guider l'équipe", c'est-à-dire:

- diriger et gérer l'exécution du projet
- produire les livrables requis et collecter l'information concernant la réalisation du projet

Même si PMBOK a scindé l'exécution et la surveillance et maîtrise au niveau des processus à suivre, tout ceci se fera en parallèle...

10.1. Management de la livraison des Lots de Travail (LT)

Autoriser et accepter les LT, y compris:

- La description détaillée et les critères d'acceptation
- Dates de début et de fin planifiée, coûts planifiés (jours/homme)
- Risques et problèmes relevant
- Prescriptions au niveau du rapportage de l'avancement
- Parties prenantes impliquées
- Procédure de configuration

Identifier les problèmes (changements, risques, soucis, questions)

- Entrer les problèmes majeurs dans le registre des problèmes dès qu'identifiés
- Evaluer l'impact sur la triple contrainte et les risques
- Faire remonter, si nécessaire, avec des recommandations

Evaluer et rapporter l'avancement selon le plan

- Prendre les actions correctives s'il y a lieu

Accepter la livraison des LT (Valeur Acquise = Earned Value)

- Transférer vers le management de la configuration

10.2. Exemple de registre des problèmes majeurs

Pbms	Description et impacts	Priorité (L/M/H)	Date	Rapporté par	Assigné à	Date de résolution	Etat	Résolution
1	Une acquisition en suspend a exigé de l'effort de la part des ressources. Ceci pourrait avoir un impact de 2 semaines sur les délais	H	20/10/2016	Philippe	Thierry	2/11/2016	F	Le sponsor a libéré les ressources clefs comme prévu
2	Un consultant externe a quitté le projet sans prévenir. Si nous ne trouvons pas rapidement un remplaçant, nous pourrions avoir des retards sérieux	H	24/10/2016	Patrick	Jack	31/10/2016	F	Le fournisseur a pu rapidement livrer un remplaçant. Le délai nécessaire à la formation est d'environ une semaine. Pas de délai complémentaire
3	La livraison du serveur est postposée de 2 semaines. Les tests pourraient être retardés de 2 semaines	H	10/11/2016	Brigitte	Thierry		R	Nous cherchons des serveurs alternatifs

10.3. Assurance qualité vs Contrôle de Qualité

Quelle est la différence?

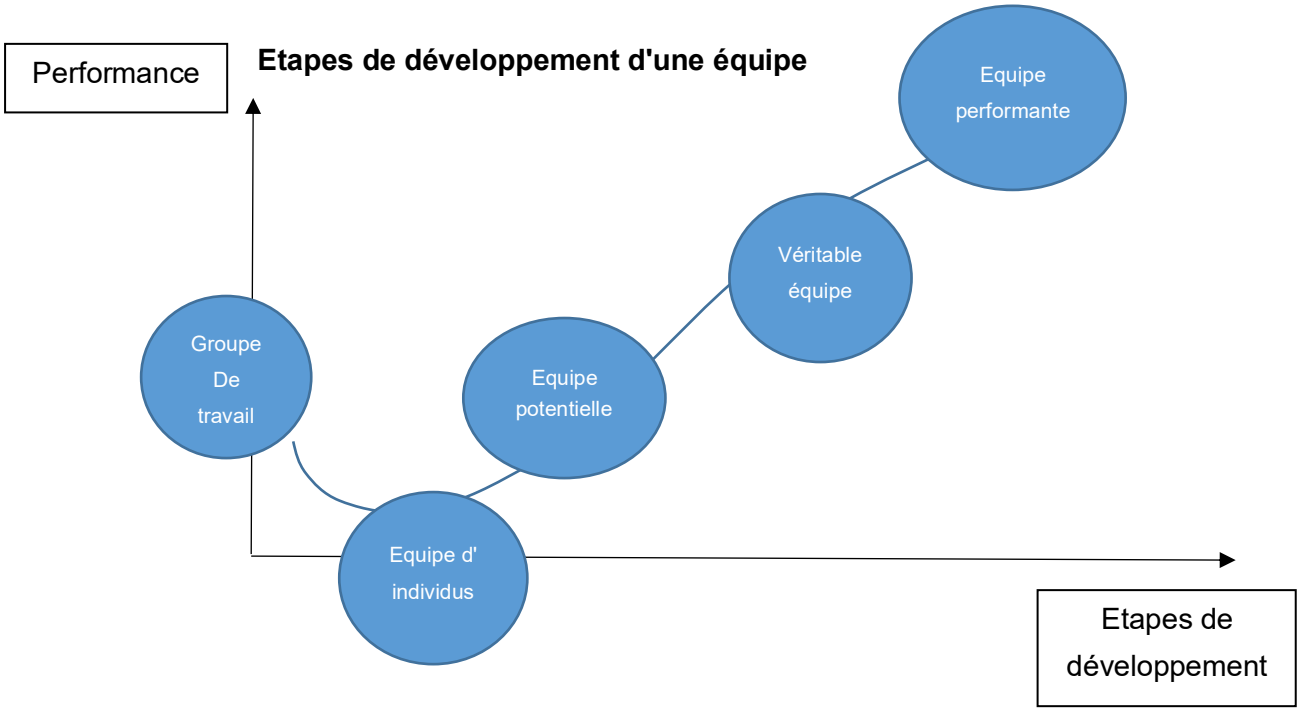
L'assurance qualité <u>évalue les processus</u>: <u>Contrôler si les processus fonctionnent</u> Le processus est-il bien suivi? Les contrôles CQ sont-ils appliqués? Les contrôles CQ sont-ils suffisants? Le processus cause-t-il des problèmes de qualité? Le processus fonctionne-t-il bien dans L'organisation? <u>Outils et techniques</u> Audits Evaluations	**Les contrôles de qualité évaluent les produits:** <u>Contrôler la qualité du ou des produits</u>: Le produit est-il dans les tolérances? Le produit (ou le LT issu du cycle de vie) est-il de qualité suffisante? <u>Identifiez les moyens pour éliminer les causes?</u> <u>Outils et techniques</u> Revues/Tests Inspections
Comment allez-vous assurer la qualité du travail que vous allez effectuer? Exemple: Décrire le processus qui est utilisé pour créer un livrable	Comment contrôlez-vous la qualité du travail effectué? Exemple: Faites une révision structurée des Livrables pour trouver des inconsistances, des bugs...
Référence CMMI Process and products quality assurance	Référence CMMI : Vérification (VER) Validation (VAL)
Définitions du PMBOK	
Effectuer l'Assurance Qualité (AQ): application des activités systématiques et planifiées ayant trait à la qualité afin de s'assurer que le projet requis pour répondre aux exigences	**Effectuer le Contrôle Qualité (CQ):** Surveillance de résultats spécifiques du projet pour déterminer s'ils sont conformes aux normes de qualités applicables, et identification des moyens d'éliminer les causes de performance insatisfaisante.

10.4. Efficacité de l'équipe projet

Management des équipes de projet efficace:

- Comment un groupe de personnes devient-il une équipe efficace?

Evoluer d'un groupe d'individus vers une équipe efficace	
Groupe de travail - Les gens travaillent ensemble - Les sentiments ne font pas partie du travail - Les conflits sont accommodés - La confiance et l'ouverture aux autres sont calculées - L'information est livrée sur une base de 'j'ai besoin de savoir' - Les buts et objectifs sont plutôt personnels et flous	Equipe projet - Les gens se font confiance - Les sentiments sont exprimés ouvertement - Les conflits sont résolus - Les gens s'aident mutuellement - L'information est partagée - Les objectifs sont partagés



D'autres points existent mais ne serons pas abordés dans le cadre de ce cours (ex: charte de l'équipe, check-list de l'efficacité de l'équipe...)

11. Processus de surveillance et de maîtrise

Prendre les actions correctives en temps opportun

Le but de la surveillance et de la maîtrise du projet est de permettre des actions correctives en identifiant les déviations du travail effectué par rapport à ce qui avait été prévu dans le plan. Ceci est réalisé en mettant en place un système précis de mesure de projet pour s'assurer que les déviations par rapport au plan soient détectées le plus tôt possible.

Un rapportage approprié des performances à des intervalles réguliers crée la transparence désirée pour le Comité de Pilotage, ce qui n'est pas toujours le cas.

La gestion des modifications au plan, de façon contrôlée, fait également partie de la maîtrise et surveillance du projet.

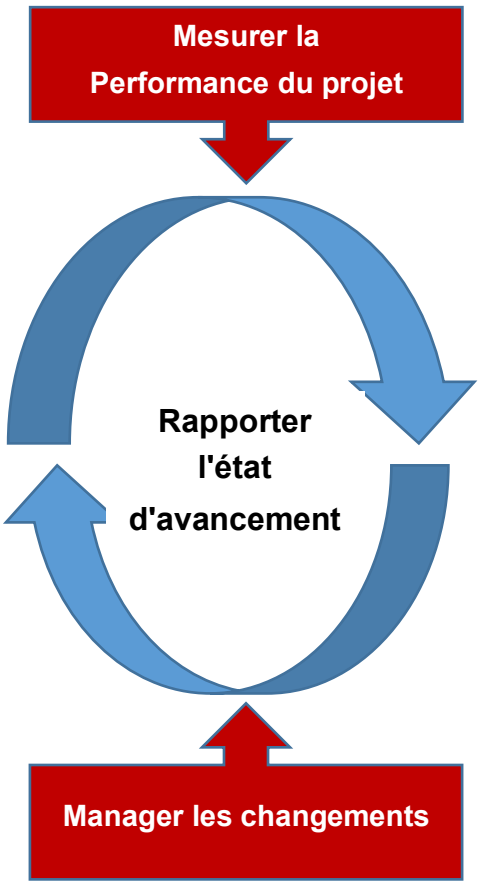
Les processus sont repris dans la colonne encadrée à la feuille suivante.

Domaines de connaissance	Groupes de processus de management de projet				
	Groupe de processus de démarrage	Groupe de processus de planification	Groupe de processus d'exécution	Groupe de processus de surveillance et de maîtrise	Groupe de processus de clôture
4. Management de l'intégration du projet	4.1 Élaborer la charte du projet	4.2 Élaborer le plan de management du projet	4.3 Diriger et piloter l'exécution du projet	4.4 Surveiller et maîtriser le travail du projet 4.5 Mettre en œuvre la maîtrise intégrée des modifications	4.6 Clore le projet ou la phase
5. Management du contenu du projet		5.1 Recueillir les exigences 5.2 Définir le contenu 5.3 Créer la SDP		5.4 Vérifier le contenu 5.5 Maîtriser le contenu	
6. Management des délais du projet		6.1 Définir les activités 6.2 Organiser les activités en séquence 6.3 Estimer les ressources nécessaires aux activités 6.4 Estimer la durée des activités 6.5 Élaborer l'échéancier		6.6 Maîtriser l'échéancier	
7. Management des coûts du projet		7.1 Estimer les coûts 7.2 Déterminer le budget		7.3 Maîtriser les coûts	
8. Management de la qualité du projet		8.1 Planifier la qualité	8.2 Mettre en œuvre l'assurance qualité	8.3 Mettre en œuvre le contrôle de qualité	
9. Management des ressources humaines du projet		9.1 Élaborer le plan des ressources humaines	9.2 Constituer l'équipe de projet 9.3 Développer l'équipe de projet 9.4 Diriger l'équipe de projet		
10. Management des communications du projet	10.1 Identifier les parties prenantes	10.2 Planifier les communications	10.3 Diffuser les informations 10.4 Gérer les attentes des parties prenantes	10.5 Rendre compte de la performance	
11. Management des risques du projet		11.1 Planifier le management des risques 11.2 Identifier les risques 11.3 Mettre en oeuvre l'analyse qualitative des risques 11.4 Mettre en oeuvre l'analyse quantitative des risques 11.5 Planifier les réponses aux risques		11.6 Surveiller et maîtriser les risques	
12. Management des approvisionnements du projet		12.1 Planifier les approvisionnements	12.2 Procéder aux approvisionnements	12.3 Gérer les approvisionnements	12.4 Clore les approvisionnements

La surveillance et la maîtrise couvrent les processus suivants dans le PMBOK:

- **Surveiller et maîtriser le travail projet (4.5)**: surveiller et maîtriser les processus utilisés pour le démarrage, la planification, l'exécution et la clôture du projet afin d'atteindre les objectifs de performances définis dans le plan de management du projet
- **Maîtrise intégrée des modifications (4.6):** Effectuer le suivi de toutes les demandes de modifications. Les approuver, et maîtriser les modifications apportées aux livrables et à l'actif organisationnel.
- **Vérification du contenu (5.4)**: processus de formalisation de l'acceptation des livrables achevés du projet.
- **Maîtrise du contenu (5.5)**: Processus de maîtrise des modifications apportées au contenu du projet.
- **Maîtrise de l'échéancier (6.6):** maîtrise des modifications relatives à l'échéancier du projet.
- **Maîtrise des coûts (7.3)**: processus utilisé pour influencer les facteurs générateurs d'écarts des coûts et maîtriser les modifications de budgets du projet.
- **Mettre en œuvre le contrôle qualité (8.3)**: surveillance de résultats spécifiques pour déterminer pour déterminer s'ils sont conformes aux normes de qualité applicables, et identification des moyens d'éliminer les causes de performances insatisfaisantes.
- **Diriger l'équipe projet (9.4)**: suivre la performance des membres de l'équipe, effectuer des retours d'informations, résoudre les problèmes et coordonner les modifications afin d'améliorer la performance du projet.
- **Etablissement du rapport d'avancement (10.3)**: ce processus permet de collecter et diffuser les informations sur la performance. Il englobe les rapports d'état, la mesure de l'avancement et les prévisions.
- **Manager les parties prenantes (10.4)**: ce processus concerne le management des communications afin de satisfaire les exigences des parties prenantes du projet et résoudre les problèmes majeurs avec elles.
- **Surveillance et maîtrise des risques (11.6)**: ce processus consiste à suivre les risques identifiés, surveiller les risques résiduels, identifier les risques nouveaux, exécuter les plans de réponse aux risques et évaluer leur efficacité au long du cycle de vie du projet.
- **Administration du contrat (12.5):** effectuer le management du contrat et des relations entre l'acheteur et le fournisseur, passer en revue et documenter les performances passées ou présentes d'un fournisseur afin de définir les actions correctives nécessaires et établir une base de relations futures avec lui, maîtriser les modifications concernant ce contrat et, au besoin, gérer les relations contractuelles avec l'acheteur extérieur du projet.

La surveillance et maîtrise de projet a au moins un processus pour chaque domaine de connaissances.



Trois activités majeures peuvent être distinguées dans la surveillance et maîtrise du projet:

1. Mesurer la performance du projet: mesurer le plus exactement possible les paramètres suivants:
    - Etat: décrivez où en est le projet, la phase ou le lot de travail pour l'instant
    - Progression: décrivez ce que les équipes ont accompli en comparaison avec ce qui était planifié
    - Prévision: prédire sur quoi la performance future du projet sera basée, tenant compte de notre connaissance actuelle
2. Rapporter la progression:
    - Les données de performance du projet devront être rapportées aux personnes appropriées pour obtenir des décisions sur la suite, les modifications ou la clôture du projet. Ceci doit être fait à des intervalles réguliers afin que le bon effet soit obtenu et que les décisions soient prises au bon moment

3. Manager les modifications:
    - Tout projet est susceptible de changer par rapport au plan original. Le chef de projet ne doit aucunement s'assurer que le plan actuel soit réalisé coûte que coûte et résister à chaque remise en question de celui-ci. Le chef de projet gère les activités projet pour s'efforcer d'atteindre les objectifs du projet. Si ceci était (éventuellement) mis en danger, le chef de projet aura la responsabilité de déterminer, avec l'approbation du comité de pilotage, si le projet doit être modifié, voire clôturé prématurément.

Mesure de la performance d'un projet

La valeur acquise (Earned value ou EV) est le standard international pour mesurer l'état d'avancement d'un projet.

Qu'est-ce que la valeur acquise?

- Une unité de mesure de l'état d'avancement d'un projet
- Une méthode conforme pour analyser l'avancement et la performance du projet
- La base de l'analyse de la performance des coûts d'un projet

Historique de la valeur acquise:

- Début des années 1900 – Ingénieurs dans les usines
    - Approche 3-dimensionnelle pour évaluer la "performance des coûts"
- Année 60: Standard du département de la défense
    - 1962: PERT/Coût
    - 1967: C/CSCS
    - 1996: *Earned Value Management System* (EVMS)
- A présent: gain de popularité au sein de la gestion de projet
    - Il y a aussi bien des partisans que des opposants!

Management de la valeur acquise (EV)

- Exemple:
  A un certain moment dans le projet

Planned	Actual	Variance
600	400	200
Valeur du travail exécuté à ce jour = 300€		

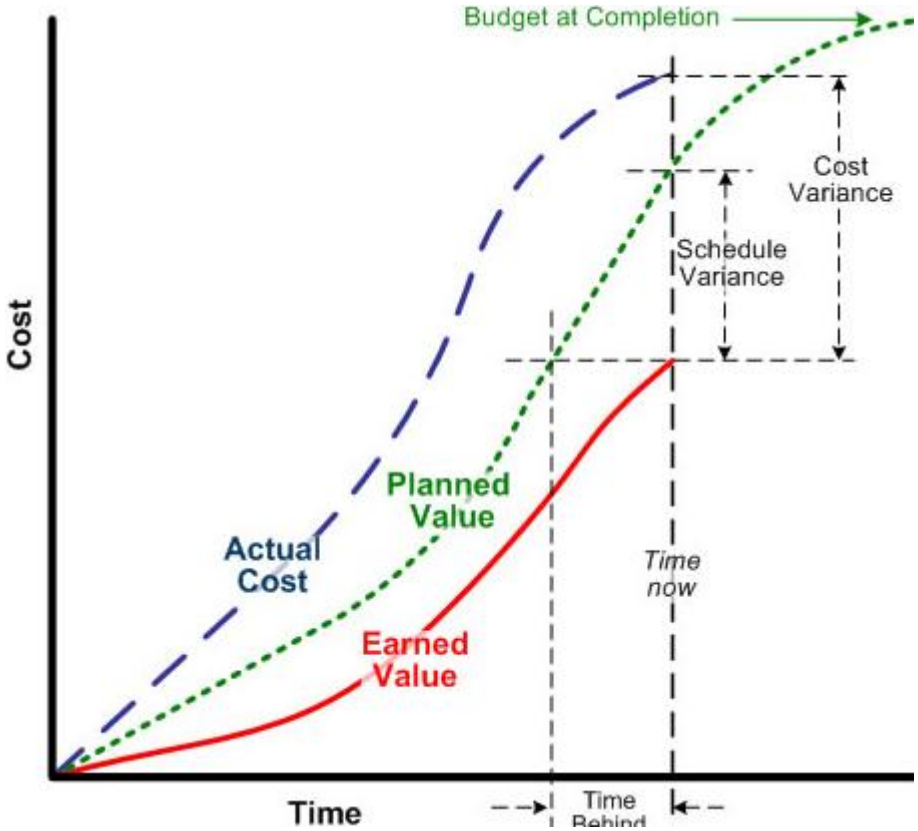
Si nous nous référons uniquement à la deuxième ligne du tableau, nous avons une variance de 200€ par rapport à ce qui avait été planifié, soit 600€. Nous pourrions croire que nous avons gagné 200€... si nous ne tenions pas compte du travail réalisé, ce qui est erroné bien sûr! C'est précisément cette information qui est fournie par la méthode de la 'EV'.

Mesurer et analyser la 'EV' à intervalles de temps réguliers durant le cycle de vie du projet, nous fournira de l'information au sujet du coût actuel du projet avant qu'il soit terminé

Exemple
Le projet a démarré le mois dernier. Un budget de 600€ a été planifié pour le premier mois. Nos dépenses sont actuellement de 400€ (à la fin du mois). Le projet a donc requis 200€ de moins que prévu.
"Est-ce une bonne nouvelle?". La valeur acquise répond à cette question: "cela dépend?"

Voyons ce qui s'est réellement passé durant le mois. Le client ne veut payer que 300€ car le travail est en retard. Nous sommes donc hors budget et hors délai!

***Earned Value* – Valeurs clefs**



- Planned Value (PV) = Valeur Planifiée = Coût Budgété du Travail Planifié
  Coût budgété du travail qu'il est prévu de réaliser pour une activité ou un composant de la structure de découpage du projet jusqu'à une date donnée (coût budgété du travail prévu)
- Coût réel (AC): Coût total encouru pour effectuer le travail de l'activité ou du composant de la structure de découpage du projet au cours d'une période donnée (coût réel du travail effectué)
- Valeur acquise (EV): montant du travail budgété effectivement réalisé au cours d'une période déterminée (coût budgété du travail effectué)
  Combien de travail avez-vous réellement effectué?

Mesure de performance

Variances:

- Ecart de coût (Valeur Acquise – Coût Réel) → **CV** = EV – AC
- Ecart de délais (Valeur Acquise – Valeur Planifiée) → **SV** = EV – PV

Des résultats positifs (>=0) indiquent une situation positive, tandis que des valeurs négatives (<0), montrent une déviation négative par rapport à la situation désirée.

Indices:

- Indice de performance des coûts: **CPI** = EV/AC (<1 est négatif)
- Indice de performance des délais: **SPI** = EV/PV (>=1 est positif)

Dans l'hypothèse que le processus d'exécution continue avec la même performance, nous pouvons calculer le coût estimé pour l'achèvement en utilisant les formules suivantes:

- Coût estimé pour achèvement: **ETC** = (BAC – EV) / CPI (BAC=Budget à la fin du projet)
- Coût final estimé: **EAC** = (AC + ETC)

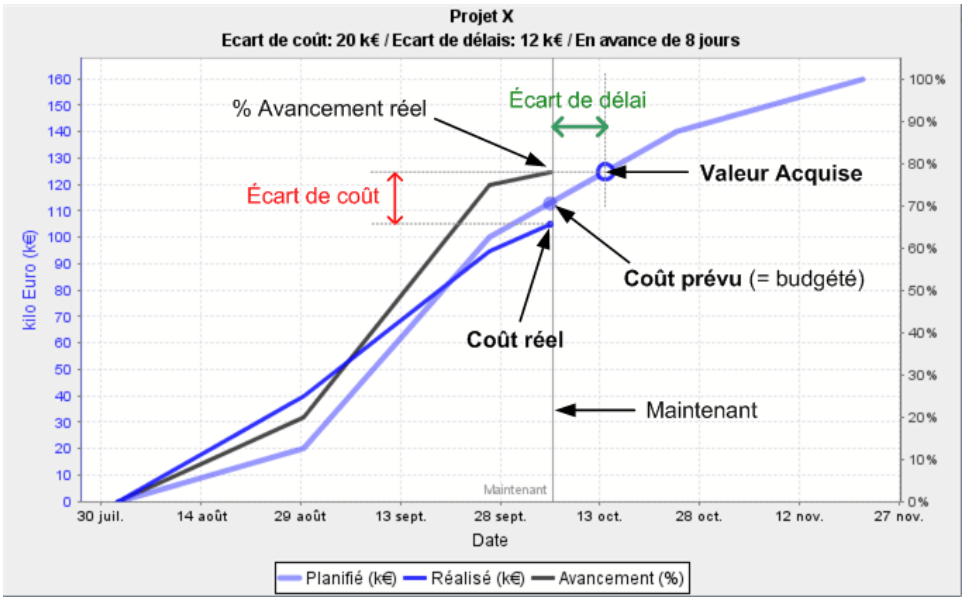
Remarque:

- L'avancement réel valorisé en euros s'appelle la valeur acquise, donc la valeur de ce qui a été fait, et donc considéré comme acquis.
- L'interprétation du diagramme est simple avec 4 configurations possibles:
  - **Tout va bien**, lorsque la valeur acquise est supérieure au coût réel et au plan.
  - **Rien ne va plus**, lorsque la valeur acquise est inférieure au coût réel et au plan.
  - **Trop cher mais plus vite que prévu,** lorsque la valeur acquise est inférieure au coût réel mais supérieure au plan.
  - **En retard mais moins cher que prévu,** lorsque la valeur acquise est supérieure au coût réel mais inférieure au plan.
  - Représentation: voir ci-dessous

Représentation de ces différents cas

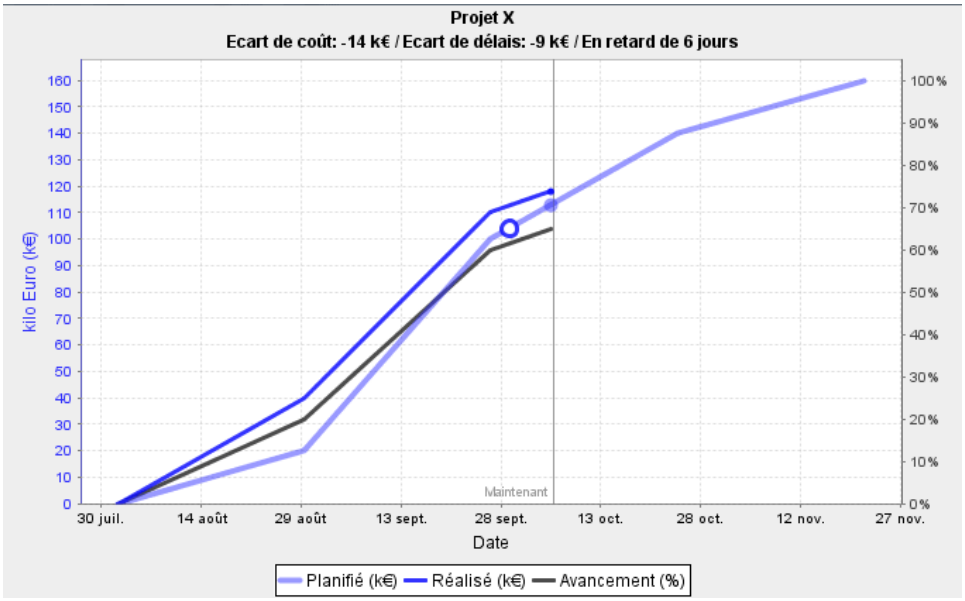
Cas n°1 : Tout va bien

Dans ce cas, le projet est en avance et dépense moins que budgété. La valeur acquise est au-dessus du coût prévu et du coût réel.



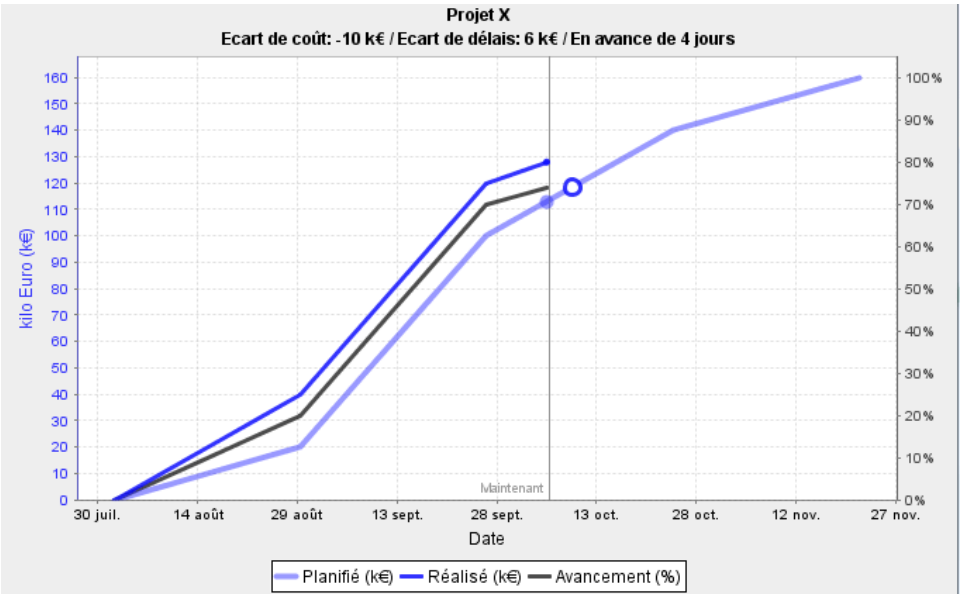
Cas n°2 : Rien ne va plus

Ici, le projet est en retard et en surcoût. La valeur acquise est en-dessous du coût prévu et du coût réel.



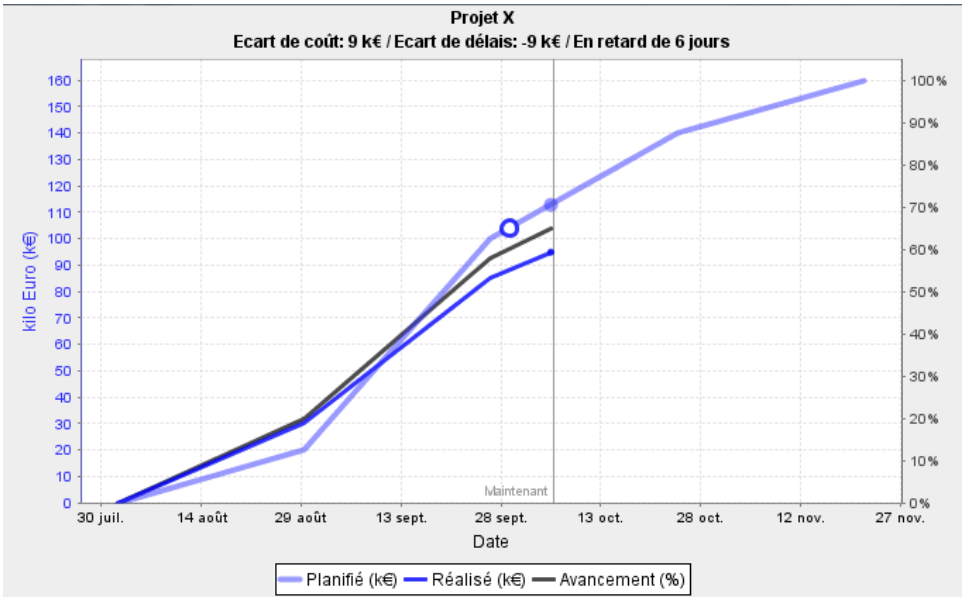
Cas n°3 : Trop cher mais plus vite que prévu

Dans ce cas, le projet est en surcoût bien qu'il soit en avance. La valeur acquise est en dessous du coût réel mais au-dessus du coût prévu.



Cas n°4 : En retard mais le projet coûtera moins cher que prévu.

Dernier cas, le projet est en retard mais réalise des économies par rapport au plan. En effet, la valeur acquise est en-dessous du coût prévu mais au-dessus du coût réel.



Source: http://blog.timeperformance.com/les-secrets-du-diagramme-de-la-valeur-acquise-en-4-images/

Exemple

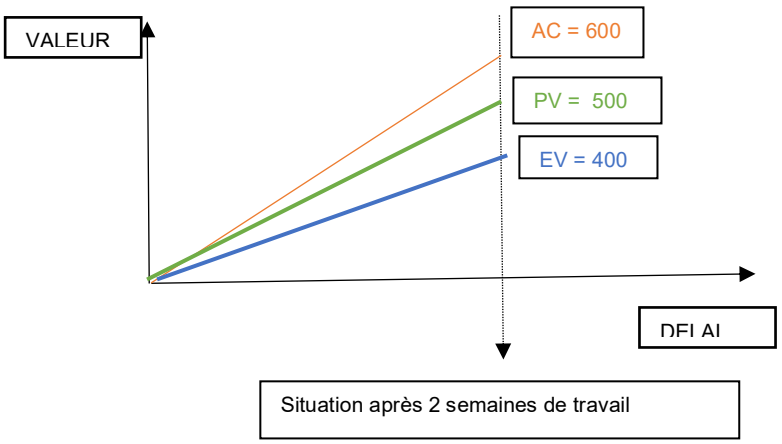
Supposons que nous ayons un projet où nous devons faire l'upgrade de 10 ordinateurs. Le temps estimé au total est de 1000h/homme (100h/homme par ordinateur). Nous avons 4 semaines pour exécuter le travail, soit 250heures/homme par semaine).

La situation fin de semaine 2 est la suivante: nous avons 4 ordinateurs installés et 600h/homme consommées.

Question: quel est l'état d'avancement du projet?

Solution

- Nous avons 4 ordinateurs installés au bout de 2 semaines, donc **EV = 400**
- Nous avons consommés 600h/homme, donc **AC = 600**
- Nous sommes à deux semaines de travail, la moitié du temps initialement prévu, donc **PV = 500**.
- Représentation graphique



Calculs des variances:

Au niveau financier:

CV = EV – AC = 400 – 600 = -200h/homme

CPI = 400/600 = 2/3 (donc pour chaque € dépensé nous accomplissons 0,67€ de travail)

Le CV est négatif et le CPI < 1, ce qui signifie que c'est mauvais au niveau financier!

Au niveau du travail accompli:

SV = EV – PV = 400 – 500 = -100h

SPI = 400/500 – 4/5 (donc pour chaque heure prestée nous avons fourni 80% du travail planifié initialement)

Le SV est négatif et le SPI <1, ce qui signifie que c'est mauvais au niveau du planning!

Coût estimé pour achèvement:

ETC = (BAC – EV)/ CPI = (1000 – 400) / (400/600) = 900h/homme

Coût total estimé:

EAC = (ETC + AC) = 900 + 600 = 1500 h/homme

Ecart à l'achèvement:

1500h/homme – 1000h/homme = 500 h/homme (soit un dépassement budgétaire de 50%!

5 étapes pour mettre *l'Earned Value* management en pratique:

1. Etablir une SDP (WBS = structure de découpage du travail)
2. Etablir une *baseline* des coûts (budget planifié dans le temps par LT ou pour plusieurs LT si certains sont trop petits)
3. Sélectionner une méthode d'EV par lots de travail (méthode de mesure du travail accompli par rapport à ce qui est attendu de nous- voir ci-dessous *)
4. Rassembler les données et calculer les indicateurs
5. Analyser et rapporter les données

(*) Sélectionner une méthode d'EV par Lot de Travail:

La valeur acquise (EV) est mesurée en fonction des réalisations actuelles et en déterminant leurs valeurs. Il est important de définir, lors du processus de planification, comment la valeur du travail évoluera au courant de l'exécution. Le choix de la méthode dépendra du type de travail qui sera effectué.

Exemples de méthodes de mesure de l'EV:

- **0 – 100**: EV reste à zéro jusqu'au moment où le travail est accepté par le client. Par exemple un lot de travail intitulé 'Approuver le rapport' est fait ou pas fait.
- **50 – 50**: 50% de la valeur du travail est allouée lorsque le travail du livrable démarre et les 100% sont acquis après acceptation du client. Par exemple, lors du démarrage du lot de travail 'Document d'exigences', on note une EV de 50% et lorsque le document d'exigences est accepté par le client on note 100%.
- **Jalons intermédiaires**: des jalons (objectifs) sont définis tout au long du cycle de production des livrables et une certaine valeur est allouée à la réalisation de chaque jalon. Par exemple, pour le lot de travail 'Analyser module X', nous pouvons allouer 5% lors du démarrage du travail, 50% à la soumission de l'ébauche du rapport, 75% après la révision du rapport par le client et 100% après acceptation par le client.
- **Unités équivalentes**: Valeur basée sur la partie terminée d'un grand nombre d'activités similaires. Par exemple, nous avons effectué 2 interviews sur 10 d'une évaluation.
- **Effort proportionnel**: EV est calculée au prorata de l'EV d'une autre unité de travail. Par exemple, l'EV du lot de travail 'Logistiques des interviews' peut suivre l'EV des interviews.
- **% complete**: Quoique utilisé régulièrement dans des projets, cette évaluation subjective devrait être évitée afin de permettre un rapportage objectif. Dans le cas de la rédaction d'un rapport de 50 pages, pouvons-nous dire que l'EV est de 50% après avoir rédigé 25 pages? La partie la plus difficile du document reste peut-être encore à rédiger!!
- **Niveau de l'effort**: EV est basé sur le temps passé sur le travail (pour du travail qui ne peut être défini ou mesuré)

12. Clôture du projet

La clôture de projet est probablement le processus le plus négligé parmi les groupes de processus de management de projet. C'est pourtant une bonne occasion de renforcer la relation avec le(s) client(s) interne(s) et/ou externe(s) et avec les fournisseurs afin d'améliorer l'efficacité des futurs projets.

Les processus concernés:

- Clore le projet: finaliser l'ensemble des activités
- Clôture des contrats: achever et effectuer le règlement final de chaque contrat, y compris la résolution de tout en suspens, et clore chacun des contrats applicables au projet ou à l'une de ses phases.

Clore le projet comprend notamment (sans entrer dans les détails):

- Evaluation de la performance du projet: performances, équipe projet...
- Documentation (historique → retenir les leçons!)
- Archiver toute la documentation du projet
- Evaluation de l'efficacité du chef de projet, des fournisseurs...

Clôture des contrats:

- S'assurer que toutes les conventions contractuelles ont été respectées afin de pouvoir terminer le contrat
- Vérification de tous les produits avant le transfert final vers le client. Nous devons obtenir l'acceptation finale du résultat final du projet de la part du client
- ...

Les négligences souvent constatées dans la clôture du projet sont:

- Temps pour contrôler les chiffres
- Temps pour mesurer la satisfaction du client
- Temps des évaluations finales des sous-traitants
- Temps pour l'évaluation finale des membres de l'équipe
- Temps pour résumer les leçons retenues
- ...

13. Modèles de cycle de vie pour les applications informatiques

13.1. Waterfall

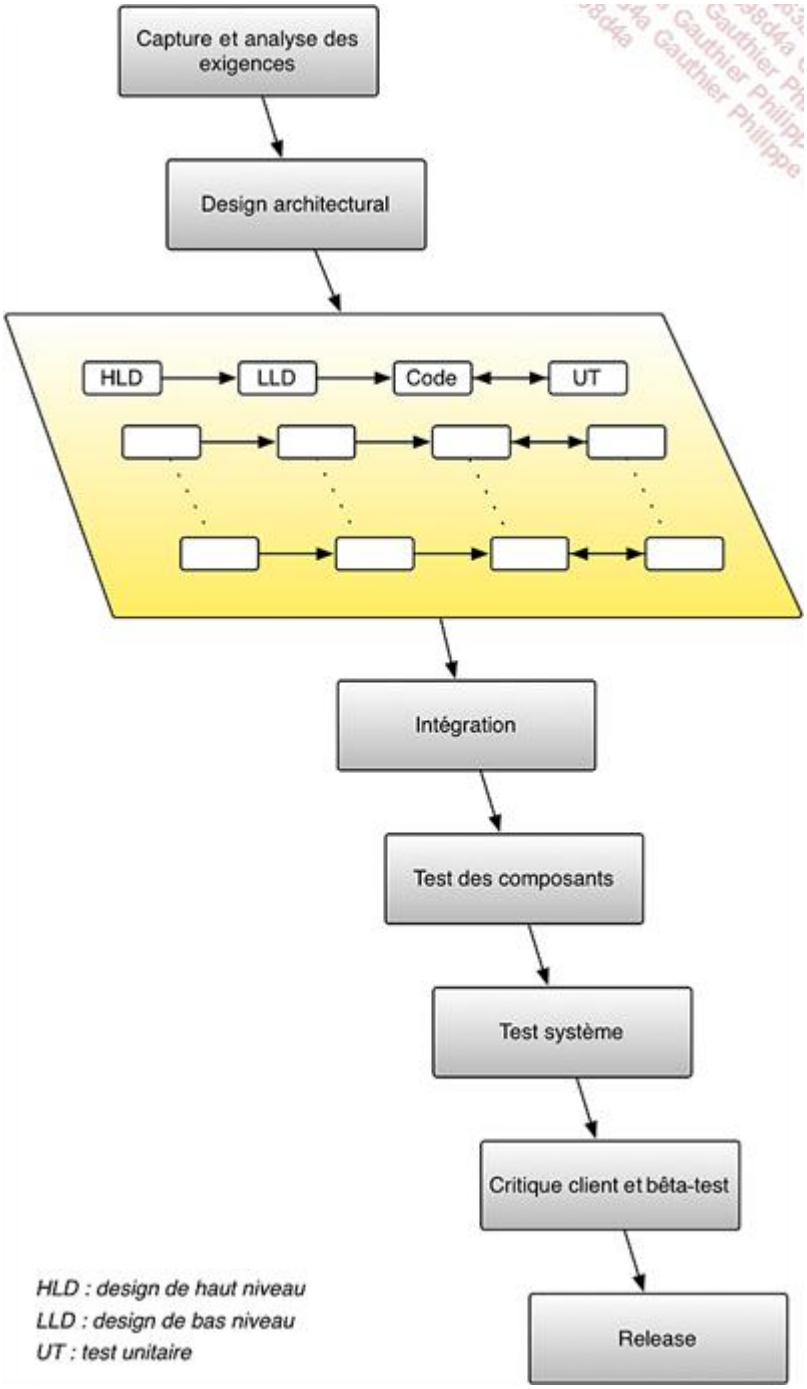
13.1.1. Principe du Waterfall

À la fin des années 70, les développements logiciels se caractérisaient par de sérieux retards et des surcoûts désastreux qui faisaient frémir les chefs de projets. Ceux-ci se consacraient alors essentiellement à la planification et au contrôle de leur production et commençaient à être dépassés par la complexité croissante des développements sous leur responsabilité. L'émergence du processus Waterfall vint à point nommé donner un souffle nouveau aux méthodologies vieillissantes de l'époque.

Ce processus scinde la tâche ardue et monolithique (d'un seul bloc) du développement en une succession d'étapes (activités) plus simples qui s'enchaînent logiquement : spécification fonctionnelle, design architectural, code, intégration, test, bêta-test, release… D'ailleurs, on retrouve encore l'essentiel de ces étapes dans les processus modernes.

L'approche « diviser pour mieux régner » du processus Waterfall a de nombreux avantages : elle permet un suivi plus précis de l'avancement du projet et permet d'identifier assez tôt les éventuels glissements ; elle force l'organisation en charge des développements à être mieux structurée (ce qui est d'autant plus profitable que l'organisation est vaste) ; elle demande à produire un certain nombre de documents qui serviront de référence pour la validation et la maintenance du système, voire même de documents contractuels en cas de sous-traitance d'une partie des étapes.

Ce processus vise à délivrer des produits en temps et en heure, à des coûts maîtrisés.



La phase de design se décompose en deux niveaux de granularité :

- Une couche de design de haut niveau qui définit les interfaces et les composants : interfaces générales pour les utilisateurs (GUI – Graphical User Interface), interfaces de programmation (API – Application Programming Interface), interfaces entre composants, protocoles de communication… et qui s'assure aussi que l'ensemble des exigences est bien couvert par les composants.
- Une couche de design de bas niveau, celui-ci détaillant le précédent en spécifiant plus précisément le contenu des composants : classes, méthodes, packages.

L'étape d'implémentation est itérative. Ce qui permet une certaine forme de parallélisme de la production, si cela a été pensé lors de la phase de design. Elle vise à coder l'ensemble des composants définis dans le design. Les tests unitaires sont couplés à l'implémentation. Il s'agit de modules autonomes et exécutables qui valident le comportement des composants, la fidélité au design, les entrées/sorties, les effets de bord…

À l'issue de la phase de codage, on teste l'interaction des composants entre eux via des tests d'intégration et enfin on valide le comportement général du produit par des tests système qui sont censés couvrir tout le périmètre fonctionnel. Ces tests sont déroulés suivant un document de « test plan » par des équipes de validation.

Vient ensuite une phase de pré-production, nommée bêta-test, où l'on va proposer le produit à une série limitée d'utilisateurs afin de valider définitivement l'usage du système et lever les éventuels bugs d'usage non détectés par les équipes de qualification.

Enfin, lorsque les responsables du produit jugent qu'il est conforme à leurs exigences, ils décident de mettre le produit en production sous forme d'une release.

L'inconvénient majeur de ce processus - et c'est pourquoi d'autres méthodologies plus agiles ont vu le jour depuis - est que les durées de réalisation de chacune des étapes, mises bout à bout, peuvent s'étendre sur plusieurs années avant la mise en production d'un gros projet (*Time To Market*, proportionnel à la taille du produit).

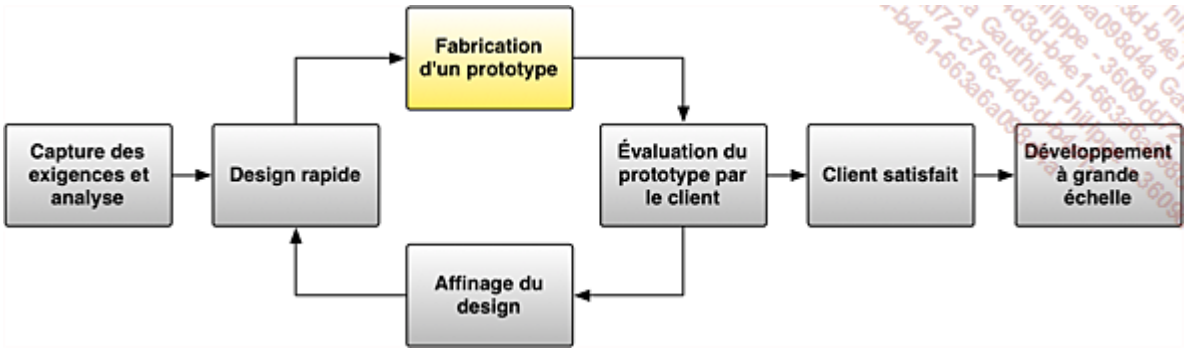
13.1.2. Prototype

La première étape du modèle Waterfall est la capture des exigences utilisateur. Une fois que l'on sait ce que le système doit faire, on peut passer à la phase de design et d'implémentation. Un problème épineux survient dès lors que les spécifications changent avant la fin du cycle de développement (ce qu'elles feront indubitablement, car c'est peut-être l'unique certitude de la création informatique, sauf bien entendu dans des projets où le périmètre fonctionnel demeure intangible depuis longtemps, bien connu, maîtrisé, mais c'est chose rare).

Parfois même il n'y a pas de spécification : un beau matin, un membre de l'équipe marketing vient vous voir et vous dit : « j'ai une super idée, un logiciel qui doit faire ci et ça, les utilisateurs vont adorer ». Alors, en tant qu'architecte, vous mettez toute votre attention au service de l'expert métier et vous commencez à l'abreuver de questions concernant les fonctionnalités et l'usage de son nouveau produit. Dans un cas comme celui-ci, une approche trop formelle, partant sur une analyse fonctionnelle dans les règles avec tout l'arsenal, serait tout à fait contre-productive, car l'expert métier vient vous voir non pour exprimer son besoin, mais pour le comprendre. C'est un peu comme une thérapie, les réponses ne sont pas ailleurs qu'en nous, mais il faut quelqu'un pour nous aider à les trouver. C'est aussi l'un des rôles de l'architecte, et l'un de ses outils magiques est le prototype.

Le prototype n'a pas l'usage pour but, mais il permet de le cerner ; c'est un outil de spécification qui sert souvent de preuve de concept (POC). La plupart du temps il est jetable, mais il arrive aussi qu'un prototype évolue jusqu'à ce qu'il réponde à la demande de l'utilisateur voire parfois devienne un produit. Ce cas-là est d'ailleurs problématique, car certains managers ne comprennent pas toujours que ce qui a été produit en moins d'une semaine soit tout à coup estimé à plusieurs mois de développement. De plus, le design est souvent négligé, voire absent de ces lignes de code esquissées, il faut donc faire attention à tout reprendre depuis le début pour ne pas partir sur des bases architecturales fragiles qui conduiraient inévitablement à une refonte lourde et coûteuse.

« Voir c'est croire ! » semble nous dire l'approche prototype. C'est une bonne façon pour les utilisateurs (marketing) et le développeur de parler le même langage et d'éviter les confusions lors de la spécification du produit.



13.1.3. Avantages et inconvénients de cette méthode

Avantages

- Méthode facile à mettre en place, logique et structurée
- Méthode qui s'adapte parfaitement à des projets qui répondent à des objectifs clairement identifiés ainsi qu'aux projets où la qualité prime sur le coût et les délais
- Dès le départ, l'intégralité du projet est définie et planifiée avec précision, ce qui facilite (en théorie) l'estimation précise du budget, du temps et des ressources nécessaires à l'accomplissement de l'ensemble du projet
- Les différentes étapes se succèdent dans un ordre rigoureux en suivant les échéances, ce qui permet au projet d'être livré dans les délais. Cette rigidité simplifie la gestion du projet, tout comme la succession des différentes phases sans aucun chevauchement.

Inconvénients

- L'inconvénient majeur de cette approche est son manque de flexibilité à cause de son déroulement séquentiel. En effet, la méthode waterfall ne laisse aucune place aux changements et aux imprévus.
- Les risques de déception du client sont plus grands. Puisque celui-ci ne voit le produit qu'à la livraison, il se peut qu'il soit déçu du résultat final car ses attentes ont évolué ou le contexte a changé, et le projet ne répond pas aux besoins actuels.
- Tous changements impliquent de revoir le projet dans son intégralité (ou presque) car toutes les phases en seront potentiellement affectées. Cela va alors générer des retards et des coûts supplémentaires importants. Par exemple, cela vous coûtera beaucoup plus cher d'ajouter une pièce supplémentaire à votre maison déjà terminée, que si cette pièce avait été prévue sur le plan initial.
- La méthode en cascade n'est pas adaptée aux projets complexes de grande envergure.

Utilisation

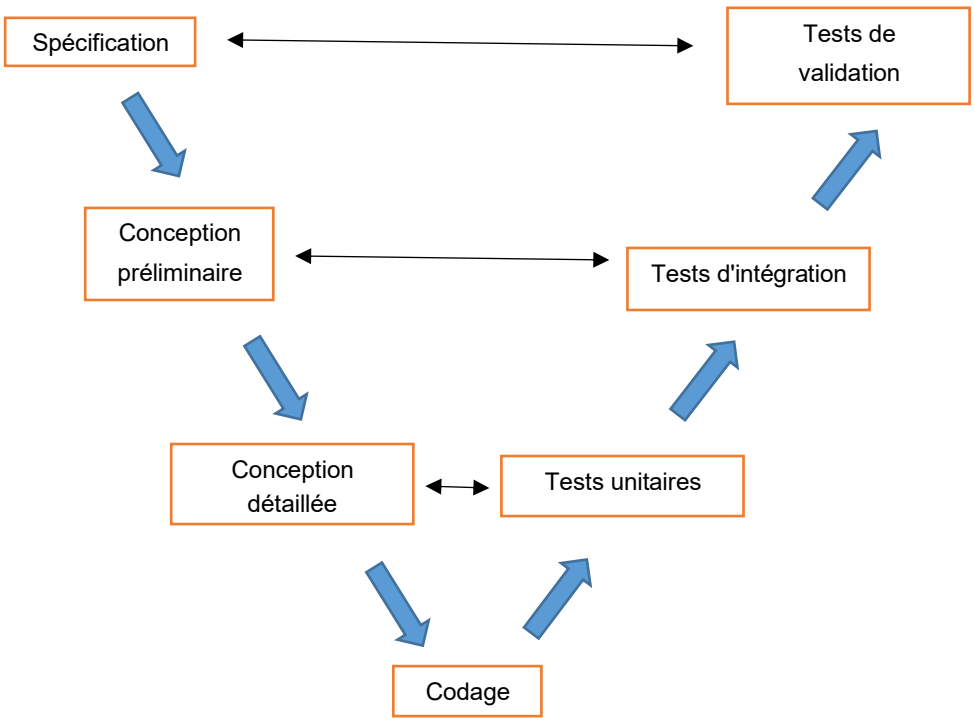
La méthode waterfall reste l'une des méthodologies de gestion de projet très utilisée. Simple et logique, son organisation stricte permet d'établir plus facilement le budget, le temps et le travail nécessaires au projet.

Cependant, cette approche doit être privilégiée lorsque le projet bénéficie d'une vision claire dès le départ et que le client n'a pas la possibilité de modifier l'étendue du projet une fois qu'il a débuté.

Source: https://www.planzone.fr/blog/quest-ce-que-la-methodologie-waterfall

13.2. Modèle en V

13.2.1. Présentation



Le cycle en V décrit les étapes essentielles du développement d'un logiciel, le cycle de vie du projet. Il est représenté par un V dont la branche descendante contient toutes les étapes de la conception du projet, et la branche montante toutes les étapes de tests du projet. La pointe du V, quant à elle, représente la réalisation concrète du projet, le codage.

Chaque étape d'une branche a son pendant dans l'autre branche, c'est à dire qu'une étape de conception correspond à une étape de test qui lui est spécifique. Une étape de test peut donc être élaborée dès que la phase de conception correspondante est terminée, indépendamment du reste du projet.

13.2.2. Spécification

La spécification est souvent le document d'entrée du projet qui est parfois rédigée en commun avec le client (cahier des charges).

La spécification, bien rédigée, est une suite d'exigences, si possible regroupées par thèmes, pour permettre d'esquisser un début d'architecture et aussi un plan de test.

Chaque exigence doit être testable, mesurable, etc...
Attention que nous parlons ici d'une situation idéale, sans changement, avec un périmètre clairement défini au départ et non modifiable, or nous savons que les changements au niveau des exigences sont monnaie courante et donc les tests devront suivre...

Il nous faudra pourtant, avant de démarrer un projet, bétonner au maximum la spécification. Sans cela notre projet pourrait dériver et ne plus correspondre à la demande initiale du client.

13.2.3. La conception

La conception regroupe les activités d'étude qui suivent la spécification, et ce jusqu'au codage. Le codage n'est que la matérialisation de la conception.

La conception englobe:

- **la modélisation**: facultative, assistée ou non d'un outil, elle permet de définir et de visualiser le système ou une partie de celui-ci, aussi bien de manière statique que dynamique. La modélisation n'est pas une étape en soi, elle peut être utilisée à tout endroit du projet. Outre ses qualités techniques, l'aspect visuel de cette pratique provoque un fort impact chez le client, qui peut plus facilement imaginer son produit fini.

- **l'architecture, ou conception préliminaire**: elle définit l'ensemble des briques constitutives de l'application et leurs interfaces. C'est donc la définition des différents modules/applications et des services qu'elle offre. C'est souvent le point faible d'un logiciel, car un défaut à ce niveau, donc probablement détecté en phase d'intégration, sera toujours pénible à corriger de par ses conséquences: débogage long, reprise d'intégration dans un outil de gestion de configuration, régénération de l'exécutable, traitement de la demande de correction, etc.

- **la conception détaillée**: La conception détaillée consiste, souvent, à écrire du *pseudocode* pour définir chaque fonction logicielle (conception des écrans, des interfaces, des différents modules).

13.2.4. Le codage

Le codage consiste à mettre en forme la conception détaillée, donc transformer du pseudocode en code.

13.2.5. Les tests

Les tests sont définis durant la phase de conception, comme le préconise la méthode en V.
Les tests:

- Unitaires: test de chacun des modules indépendamment des autres
- Intégration: Tests de l'ensemble des modules lorsqu'ils sont mis ensemble
- Validation (Acceptation): Tests destinés à contrôler que les fonctionnalités attendues sont bien présentes, ni plus ni moins (le système fait ce que l'on attend de lui et fournit les résultats attendus). Ces derniers tests sont réalisés par les utilisateurs finaux en suivant des scénarios qu'ils auront décrits durant la phase de conception.

13.2.6. En résumé

On utilise d'abord une approche descendante, du plus général au plus détaillé.
On se sert des documents que l'on a écrits dans les phases amont pour la validation du logiciel:

- Oblige à prendre en compte la validation dès le début du projet
- Définit les notions de tests unitaires et de tests d'intégration
- Définit les livrables (documents de spécification, de conception, de validation)

13.2.7. Avantages et inconvénients

Avantages

- Méthode simple et intuitive à utiliser et qui constitue un processus naturel où il faut concevoir avant de réaliser.
- Attitude proactive : Le travail effectué lors des phases de conception permet de limiter les risques et dérives pendant les phases de tests. Il s’agit de mettre en face de chaque phase de spécification un moyen de vérification.
- Clarté des responsabilités
- Netteté du périmètre des étapes
- L'aspect séquentiel de ce type de cycle permet de bien terminer une étape avant de commencer une autre

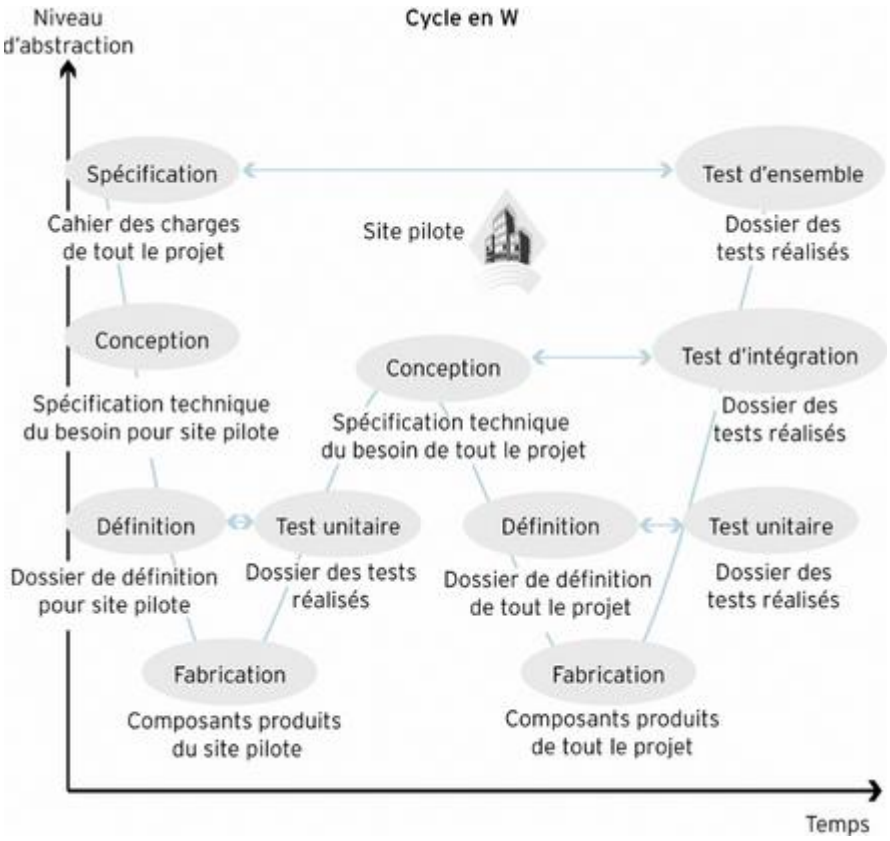
Inconvénients

- Le cycle en V ne tient pas compte de la réalité:
    - Que se passe-t-il si le cahier des charges changeait lors de la construction du système? Il n'y a pas de retour en arrière possible...
    - Que se passe-t-il lorsque l'on s'aperçoit qu'une brique technique ne permet pas de répondre à une exigence technique? Par exemple le JAVA garbage collector et les contraintes temps réel (JAVA garbage collector = gestionnaire de la mémoire)
- Pas de vérification et validation à la fin de chaque étape
- Que se passe-t-il lorsque l'on découvre un gros bug lors des tests?
    - Les équipes de tests/validation attendent (parfois longtemps!)
- Effet tunnel pour les gros projets: on ne voit pas le bout
- Peu ou pas de possibilité de maquettage et/ou de prototypage:
    - Que se passe-t-il si le commanditaire dit "ce n'est pas ça que j'ai demandé " à la fin du projet?
- Comment vérifier que l'on construit le bon système? Ou qu'il est construit correctement?

-

13.3. Modèle en W

13.3.1. Présentation



Source: https://www.google.be/search?biw=1366&bih=659&tbm=isch&sa=1&q=cycle+en+w&oq=cycle+en+w&gs_l=psy-ab.3..0j0i24k1.101962996.101964017.0.101965177.2.2.0.0.0.0.95.173.2.2.0....0...1.1.64.psy-ab..0.2.172...0i67k1.sHA64ldFvQw#imgrc=b0DBPBMYbA7QaM:

Le modèle en W part du modèle en V, en y ajoutant des phases d'élaboration de maquettes permettant de valider la phase de conception. C'est une technique utilisée lorsque, par exemple, nous avons une vision un peu floue du projet ou lorsque nous voulons gagner en qualité pour le livrable. Le " cycle en W " est applicable notamment dans les projets de développement de système d'information et dans les projets de réorganisation, dès qu'il y a volonté de s'assurer de la pertinence de la solution qui va être généralisée. Le premier contenu partiel est réalisé puis testé sur un site pilote et le retour d'expérience de celui-ci définit les ajustements à appliquer sur le premier contenu réalisé, ce qui permet d'affiner la définition du reste du projet.

13.3.2. Avantages et inconvénients

Avantages

- Cette méthode possède les mêmes avantages que la méthode en V
- De plus, la création de prototypes permet d'avoir une meilleure idée de l'aspect final du projet, évitant de tomber dans un effet tunnel

Inconvénients

- Même inconvénients que pour le modèle en V, sauf en ce qui concerne l'effet Tunnel.
- La création d'un premier prototype permet certes de diminuer légèrement les précédents inconvénients mais implique aussi des coûts et une durée plus importante

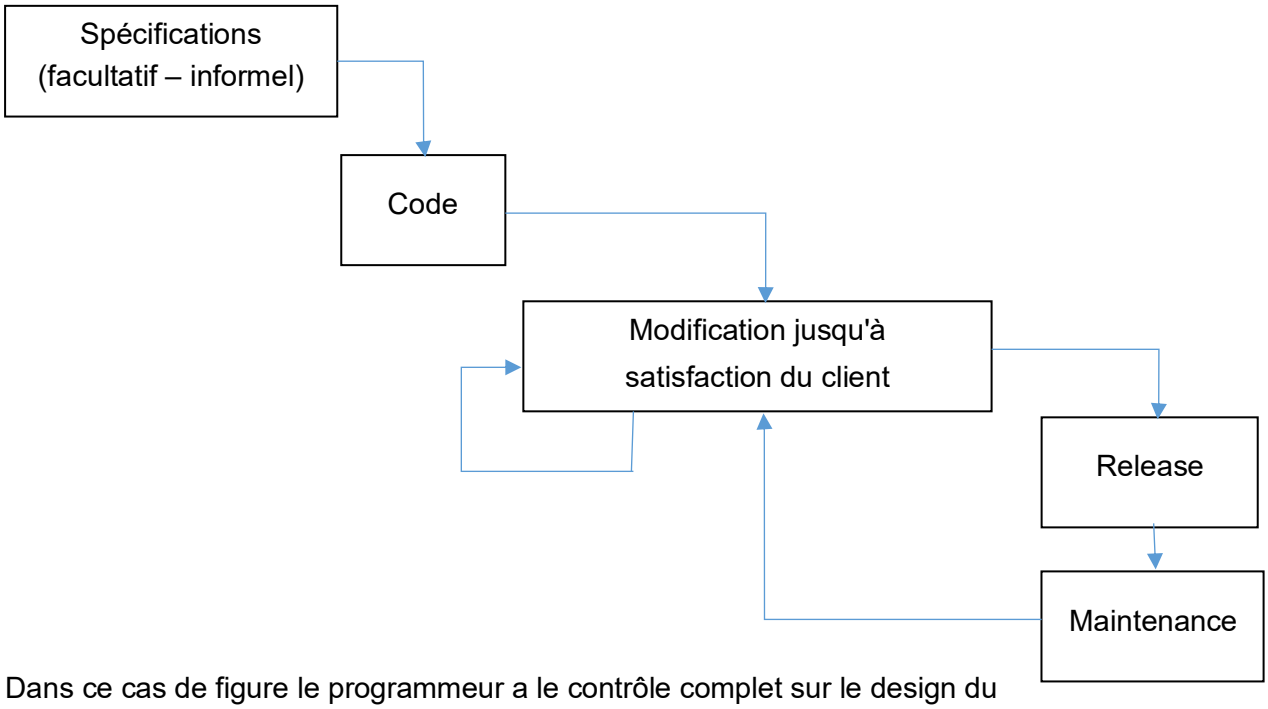
Utilisation de cette méthode

- C'est une technique utilisée lorsque, par exemple, nous avons une vision un peu floue du projet ou lorsque nous voulons gagner en qualité pour le livrable.
- Le " cycle en W " est applicable notamment dans les projets de développement de système d'information et dans les projets de réorganisation, dès qu'il y a volonté de s'assurer de la pertinence de la solution qui va être généralisée.

13.4. Code & Fix

13.4.1. Présentation

Cette méthode est également connue sous le nom du codage en mode 'cow-boy coding'! Le terme "... ne fais pas le cow-boy!" est souvent utilisé quand quelqu'un veut faire quelque chose en dehors des normes (procédures), mais peut-être l'entreprise ne dispose-t-elle tout simplement pas de procédures bien établies...



Dans ce cas de figure le programmeur a le contrôle complet sur le design du système.

13.4.2. Avantages et inconvénients

Avantages

- Pas de gestion de projet (planning, processus...), donc gain de temps
- Livraison rapide
- Peu de ressources nécessaires

Inconvénients

- Les changements sont difficiles et coûteux
- Les objectifs et délais ne sont pas définis clairement (qu'est-ce qui sera délivré et quand?)
- La qualité n'est pas vraiment mesurable (ni définie d'ailleurs)

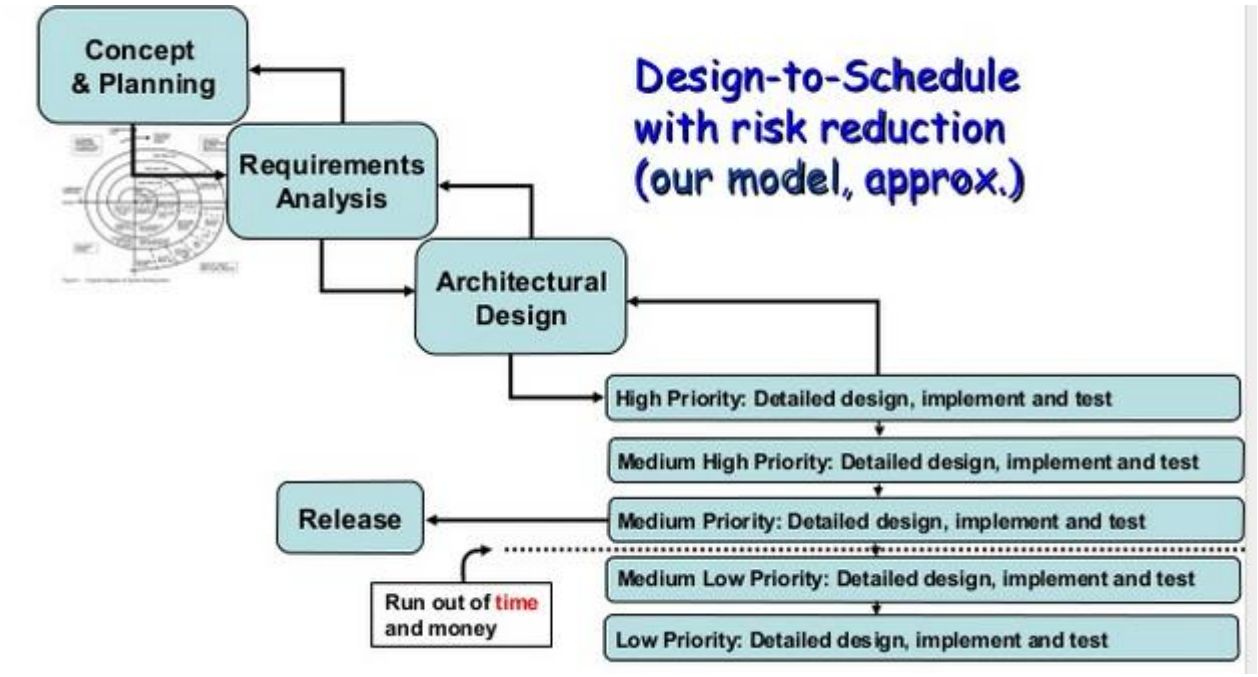
Utilisation de cette méthode

- Résultats exigés dans un laps de temps très (trop) court
- Pas de de budgets/ressources en suffisance que pour pouvoir faire autrement

- Le programmeur n'a pas compris les instructions mais il doit fournir quelque chose

13.5. Design-to-schedule

13.5.1. Présentation



Source: https://www.slideshare.net/AnoushGhamsar/architecture-presentation-8

La livraison des résultats se fait par 'étages' (stages).
Brève description des étapes de cette méthode:

- Définition du concept et du planning:
  - L'équipe projet doit acquérir une vision globale du projet et des premières exigences. Le planning est fixé (une date de fin de projet).
- Nous analysons ensuite les exigences, les objectifs à atteindre. Nous découpons le projet en différents 'étages' (chacun contenant une ou un ensemble d'exigences) et nous affectons à chacun de ces étages une priorité (high – medium – low). Celle-ci déterminera l'ordre dans lequel les étages seront traités (high → low). Cette étape est cruciale, une erreur à ce niveau pourrait mener le projet à un échec.
- Chacun des étages est ensuite exécuté entièrement (design détaillée, implémentation, tests et validation par le client), ce par ordre de priorité décroissante
- La fin du développement sera déterminée, soit par la deadline, soit parce que nous sommes arrivés à la fin du projet (quoique d'autres éléments pourraient marquer la fin du projet comme le budget, l'abandon...)

Quand on est arrivé au bout des développements (planning ou travail terminé), un contrôle est réalisé sur la documentation et le produit final.

13.5.2. Avantages et Inconvénients

Avantages

- Ce modèle diminue le risque d'erreurs durant l'exécution du projet. Des tests constants sont effectués (à chaque étage), un nombre d'erreurs minimum, voire nulle, dans le produit final peut être espérée
- Le produit sera fonctionnel, quel que soit le moment où l'on arrête le projet nous aurons un release du produit
- Les exigences de haute priorité seront développées en premier et feront donc partie des premiers releases du produit

Inconvénients

- Ce modèle particulier indique que la capacité d'une équipe projet à fournir des 'deliverables' est plus important que les coûts engagés et que l'atteinte des objectifs ou des besoins requis
- Toutes les étapes doivent être réalisées entièrement avant de passer à la suivante. Si, par exemple, nous n'avons pas définis correctement certaines caractéristiques du produit final durant la définition des exigences, alors nous allons perdre du temps lors de la phase de conception et risquons de ne pas atteindre nos objectifs.

Utilisation de cette méthode

- La décision de sélectionner cette approche ou pas est souvent due à une contrainte pour l'équipe projet de respecter un planning

13.6. Rapid Application Development (RAD)

13.6.1. Présentation

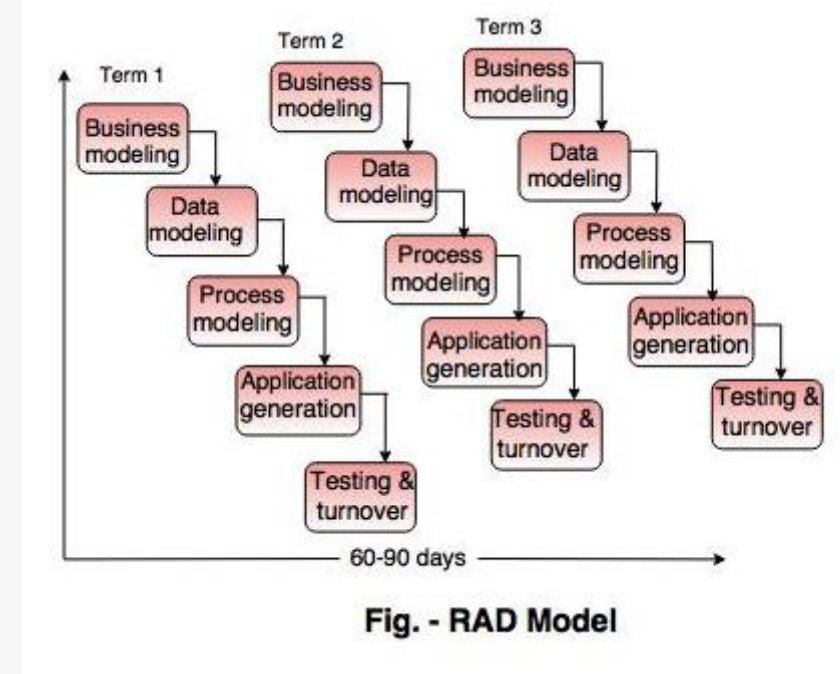


Fig. - RAD Model

Source: http://www.tutorialride.com/software-testing/rad-model.htm

Les principes du RAD sont simples et ils découlent du bon sens. S'il y a quelque chose de révolutionnaire dans la méthode, c'est leur formalisation.

Voici les principes de base

- Tout d'abord, le développement devrait être effectué par de petites équipes, expérimentées et ayant reçu toute la formation nécessaire.
- Le développement doit s'effectuer en faisant appel à une méthodologie formalisée.
- Une équipe de développement RAD doit disposer d'outils puissants qui automatisent le développement dans sa globalité, tant en ce qui concerne les étapes du développement que l'enchaînement de ces étapes.
- L'équipe de développement doit être correctement gérée, par un encadrement encourageant la réutilisation des composants, et attentifs aux aspects humains du management de projet.
- Enfin, les utilisateurs finaux devraient s'impliquer dans le processus de développement.

13.6.2. Les quatre ingrédients de base

- Les outils: de conception, de prototypage, de génération de code, disposant d'un langage de haut niveau (L4G), ces outils étant fédérés par un référentiel et constituant un atelier puissant.
- Les personnes: elles doivent être compétentes, expérimentées, correctement formées aux techniques et aux outils utilisés, et, cela ira mieux en le disant, ...motivées. Ce qui est tout aussi essentiel, c'est que l'utilisateur final soit directement impliqué dans chaque phase du projet.
- Le management: une gestion du projet, en particulier en ce qui concerne les aspects humains, est essentielle. Le pire ennemi du RAD, c'est la bureaucratie. Combien de projets n'a-t-on pas vu dériver ou stagner, ensablés par une bureaucratie toujours plus lourde.
- La méthodologie: Rien ne se fera sans une méthodologie, à la fois bien formalisée et souple. La méthode doit être consignée dans un document électronique partagé et accessible à tous. Il devra être bien structuré, de façon à ce que chacun y retrouve facilement l'information dont il a besoin sans pour cela devoir parcourir l'ensemble de son contenu. Celui-ci devra être régulièrement mis-à-jour afin d'améliorer continuellement la méthodologie (qui évoluera avec la maturité de l'entreprise au niveau de la gestion de projet)

13.6.3. Les quatre phases

Un projet RAD se découpe en quatre phases

1. Phase de définition des besoins: Elle se matérialise par des sessions de travail appelées JRP (Joint Requirements Planning ou définition conjointe des besoins).
2. Phase de "conception utilisateur" (user design): elle se caractérise par les sessions JAD (Joint Application Development ou développement conjoint d'application), qui est un des signes distinctifs du RAD.
3. Phase de construction: mêlant les spécifications détaillées et le codage.
4. Phase de finalisation et mise en place (en anglais: cutover).

Deux points essentiels :

- ces quatre phases constituent un processus itératif. Le "cutover" débouche à nouveau sur une nouvelle définition des besoins, tant que cela s'avère nécessaire
- ces phases peuvent, dans une large mesure, être exécutées en parallèle.

Les JRP
Le principe de la "définition conjointe des besoins" est simple, sélectionnez les meilleurs éléments, à la fois chez le maître d'œuvre et chez le maître d'ouvrage, et faites-les plancher ensemble, dans une salle de réunion sans téléphone et le plus loin possible de toute source de distraction. Ces personnes doivent travailler ensemble, définir ensemble les besoins du système à réaliser.

Remarque : Lorsque nous évoquons les meilleurs éléments, il ne s'agit ni du management seul, ni des "as de la programmation", mais d'une sélection de l'ensemble des profils impliquées dans le projet. Une de ces personnes clés est *l'executive owner*, en d'autres termes celui qui paye pour avoir le produit final (ou son représentant attitré). Il doit s'engager à fond dans le processus.

Les JAD
L'idée derrière les JAD, qui constituent la substance de la phase de conception, est d'augmenter la productivité globale du développement en réunissant, lors d'une même session, les utilisateurs et les concepteurs.

Encore une fois, son efficacité repose sur deux facteurs clés:

- le facteur humain : dynamique de groupe et suppression des barrières organisationnelles
- l'outillage : référentiel puissant et outils de prototypage rapide, les prototypes étant, bien sûr, réutilisables pour le développement du produit final

Les "timebox"
Dans la plupart des projets de développement d'application (si ce n'est dans TOUS les projets), il y a des glissements dans les délais. Cela tient à plusieurs causes. Mais on sait en particulier que 80% des fonctionnalités sont réalisées en 20% du temps, et que les 80% du temps restant sont pris par la réalisation des 20% des fonctionnalités à réaliser. En fait, plus le développement semble s'approcher de la fin, moins on avance vite. Cela ressemble à du polissage de pièces. Plus on avance, plus l'abrasif doit être fin. Et plus l'abrasif est fin, moins on avance!

En RAD, la loi est dure, mais simple: le glissement est interdit!
Pour un habitué du développement classique, c'est une révolution ! Comment peut-on interdire tout glissement dans les délais? Et si je n'ai pas fini de développer, comment faire?

La solution, ici aussi, est simple. Plutôt que d'autoriser des glissements dans les délais, on va tolérer un glissement dans la réalisation: réduisez les fonctionnalités, mais quoi qu'il arrive, vous finirez à temps. Le RAD, et les outils qui sont supposés le supporter, encouragent le développement par affinements successifs. Pour reprendre l'analogie précédente, si on n'a pas le temps d'obtenir un poli parfait, on va se contenter d'une pièce un peu rugueuse. Mais on ne va pas se donner du temps pour polir encore et encore.

Un *timebox* s'achève par une revue, qui décidera s'il faut ou non réitérer une deuxième boucle de la spirale de développement.
Le développement itératif ne rend pas seulement le "*timeboxing"* possible, il le rend nécessaire. En effet, le danger des "fonctionnalités rampantes" (je raffine encore, encore et encore) menace de faire dériver un projet à l'infini. Le management d'un développement de ce style est délicat. Une petite équipe (deux personnes en moyenne) va être mise sous pression pendant une durée fixée (en général 60 jours). Les actions à réaliser alors dans cette 'boîte temporelle' doivent être correctement estimées.

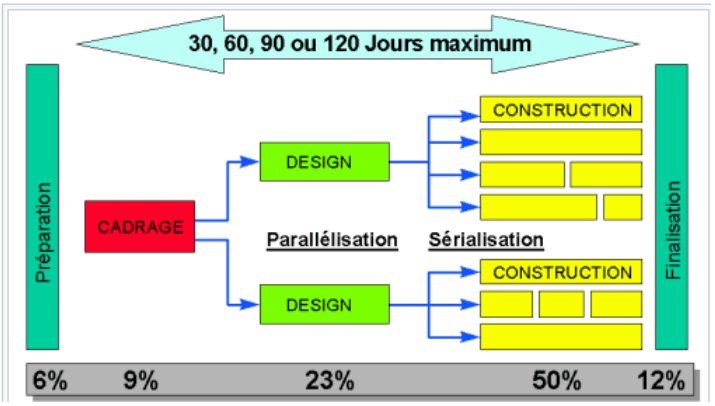
Source: http://www.yves-constantinidis.com/fichiers/yclrad.pdf

13.6.4. Remarques

1. Ce qui vient d'être dit au niveau de la méthode RAD est basé sur un livre de plus de 750 pages datant des années 1990 (James Martin). La méthode a depuis lors évolué. Si l'on se réfère à WIKIPEDIA nous trouverons 5 phases au lieu des 4 phases initiales.

La méthode RAD structure le cycle de vie du projet en 5 phases (dont 3 systématiques) :

1. L'initialisation prépare l'organisation, puis détermine le périmètre et le plan de communication ;
2. Le CADRAGE définit un espace d'objectifs, de solutions et de moyens ;
3. Le DESIGN modélise la solution et valide sa cohérence systémique ;
4. La CONSTRUCTION réalise en prototypage actif (validation permanente) ;
5. La finalisation est réduite à un contrôle final de qualité en site pilote.



Phase 1: Initialisation

Préparation de l'organisation et communication.
Cette phase permet de définir le périmètre général du projet, de structurer le travail par thèmes, de sélectionner les acteurs pertinents et d'amorcer une dynamique de projet.

Phase 2: Cadrage

Analyse et expression des exigences.
La spécification des exigences est du ressort des utilisateurs. Ils expriment leurs besoins lors d'entretiens de groupe. Il est généralement prévu de 2 à 5 jours de sessions par commission (thème).

Phase 3: Design

Conception et modélisation.
Les utilisateurs sont également impliqués dans cette étape. Ils participent à l'affinage et à la validation des modèles organisationnels : flux, traitements, données. Ils valident également le premier niveau de prototype présentant l'ergonomie et la cinématique générale de l'application. Il est prévu entre 4 et 8 jours de sessions par commission. Cette phase représente environ 23 % du projet. À partir de la phase de Design, la parallélisation du travail est possible.

Phase 4: Construction

Réalisation, prototypage.
Durant cette phase, l'équipe RAD (SWAT) doit construire l'application module par module. L'utilisateur participe toujours activement aux spécifications détaillées et à la validation des prototypes. Plusieurs sessions itératives sont nécessaires. Cette phase représente environ 50 % du projet. À partir de la phase de Construction, à la parallélisation du travail peut s'ajouter la sérialisation.

Phase 5: Finalisation

Recette et déploiement.
Des recettes partielles ayant été obtenues à l'étape précédente, il s'agit dans cette phase d'officialiser une livraison globale et de transférer le système en exploitation et maintenance.
Cette phase représente environ 12 % du projet.

Source: https://fr.wikipedia.org/wiki/D%C3%A9veloppement_rapide_d%27applications

2. Les équipes décrites par James Martin s'appellent SWAT (Skilled With Advanced Tools). On parle souvent des outils RAD, mais si on parlait un peu plus des personnes? Ce qui est rare, cher, précieux, c'est une équipe de projet capable de s'embarquer dans une telle aventure. Ce qui est d'une valeur inestimable, c'est un chef de projet capable à la fois de communiquer son enthousiasme, d'animer une équipe, de comprendre le besoin du client, et de secouer sa propre organisation pour y puiser les ressources qui lui manquent. En effet, le RAD c'est avant tout une affaire de personnes et d'organisation. La forme des bureaux et de la salle de réunion y joue un rôle aussi important que la structure du référentiel. La motivation des développeurs y joue un rôle aussi primordial que le générateur d'interface.
3. La méthode RAD est véritablement le précurseur des méthodes AGILE.

13.6.5. Avantages et Inconvénients

Avantages

- Le processus de développement et livraison est rapide
- Ce modèle est flexible au changement (d'exigences)
- Le client est présent tout au long du projet, avec des revues intermédiaires, ce qui diminue le risque de glissement par rapport aux exigences clients.

Inconvénients

- Réutilisabilité des modules (ou objets): celle-ci est vraiment préconisée par les créateurs de cette méthode or, pour qu'un composant soit réutilisable, il doit être utilisable dans des contextes différents par des applications différentes, ou par des modules différents. Ceci implique soit une réflexion en amont de sa construction, soit une "généralisation" de ce composant a posteriori pour l'ouvrir à d'autres utilisations que celle pour laquelle il a été conçu. Des délais très courts, vont faire que c'est bien ce travail de fond qui sera le premier sacrifié. Si un développeur est pris par le temps, on ne voit pas quelle raison le pousserait à chercher à rendre un composant réutilisable à plus long terme, donc à accomplir une tâche dont il ne verra jamais les fruits. La réutilisabilité demande un investissement à long terme.
- Les utilisateurs doivent montrer une grande disponibilité (risque)

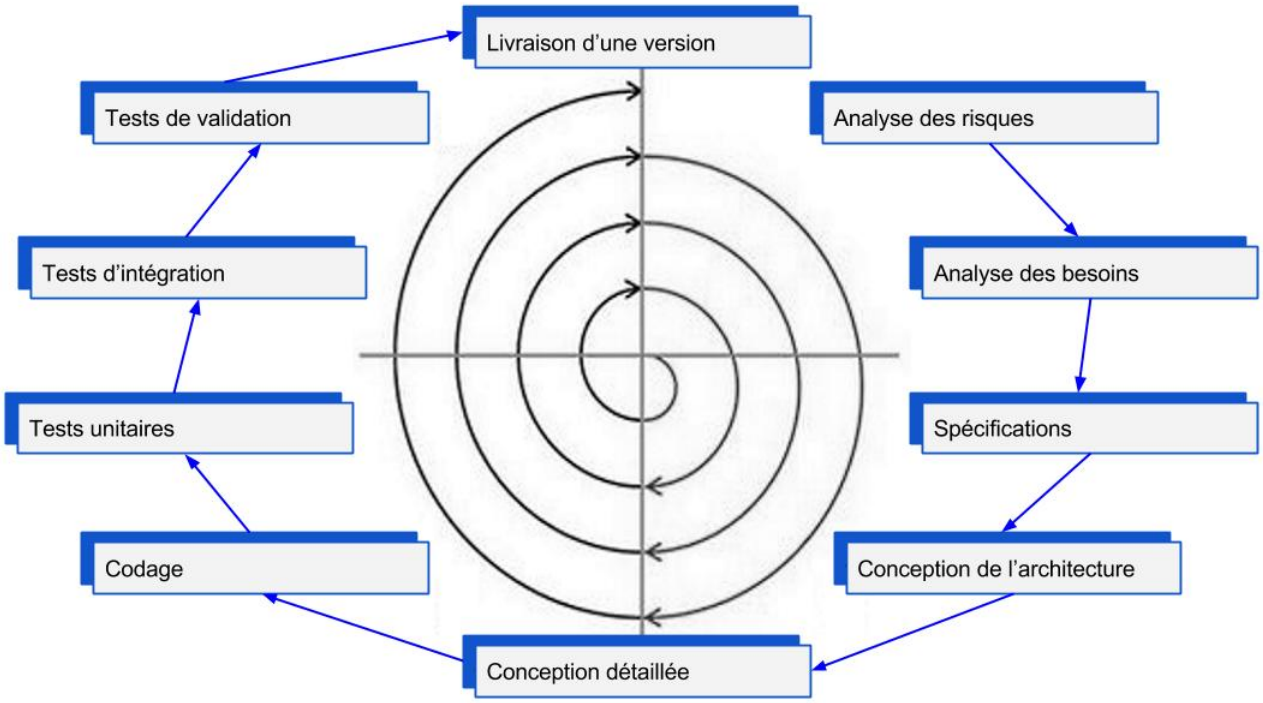
Utilisation

- Petits et moyens projets (de quelques jours à quelques semaines)
- grands projets si l'organisation a vraiment la maturité nécessaire (de quelques mois à quelques années)

13.7. Modèle en spirale

13.7.1. Présentation

Défini par Barry Boehm en 1988 dans son article « A Spiral Model of Software Development and Enhancement », le cycle en spirale reprend les étapes du cycle en V, mais prévoit l'implémentation de versions successives, ce qui permet de mettre l'accent sur la gestion des risques, la première phase de chaque itération étant dédiée à ce poste. A ce point il est nécessaire de définir la notion de prototype. En effet, on ne fait pas des versions successives d'un même produit fini, corriger une liste de bugs permet de passer de la béta à la finale mais pas de la v1 à la v2… Le cycle en spirale prévoit donc la livraison de prototypes, c'est à dire de versions incomplètes du produit. Il peut s'agir d'une simple maquette sans aucune fonctionnalité (on parle alors de prototype horizontal) ou bien de sites partiellement fonctionnels : telle version implémentera la navigation de base, la suivante ajoutera l'espace membres, puis la zone de téléchargements… on parlera alors de prototype vertical.



13.7.2. Avantages et inconvénients

Avantages

- Le but premier de ce modèle étant la gestion des risques, ceux-ci sont logiquement limités.
- L'expertise du client croit à chaque itération du cycle, l'apprentissage se fait par touche et pas d'un seul bloc.
- Ce modèle est très adaptatif : si chaque prototype apporte des fonctionnalités indépendantes, il est possible de changer l'ordre de livraison des versions.

Inconvénients

- Le principal défaut du cycle en spirale c'est qu'il n'est adapté qu'aux projets suffisamment gros
- L'évaluation des risques en elle-même et la stricte application du cycle de développement peut engendrer plus de coûts que la réalisation du logiciel.

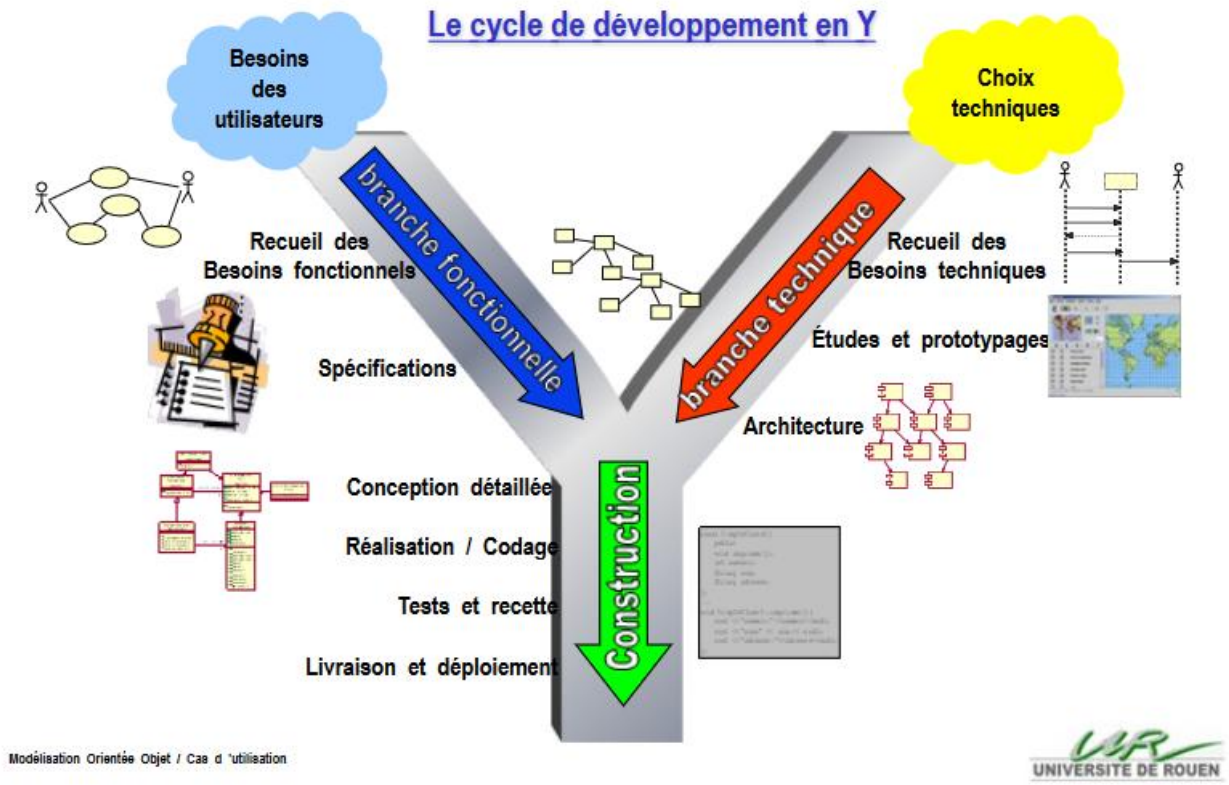
- Ce type de cycle de développement est complexe, entre les étapes prévues en théorie et celles mises en pratique il y a une grande différence.

13.8. Modèle en Y

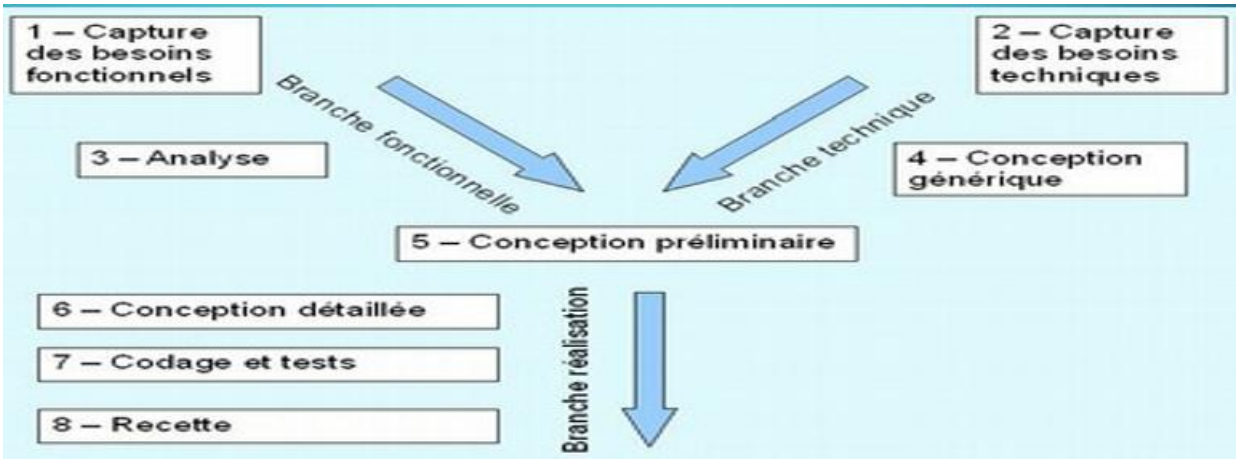
13.8.1. Présentation

2TUP = 2 Track Unifies Process :

- c'est un processus de développement de logiciel qui implémente le processus unifié (UP)
- "2TUP" est un processus qui apporte une réponse aux contraintes de changements continuels imposées aux systèmes d'information de l'entreprise. En ce sens il renforce le contrôle sur les capacités d'évolution et de correction de tels systèmes
- "2TRACK" signifie littéralement que le processus suit deux chemins. Il s'agit des « chemins fonctionnels » et « de l'architecture technique », qui correspondent aux deux axes de changement imposés au système d'information.



Source: http://slideplayer.fr/slide/1289506/



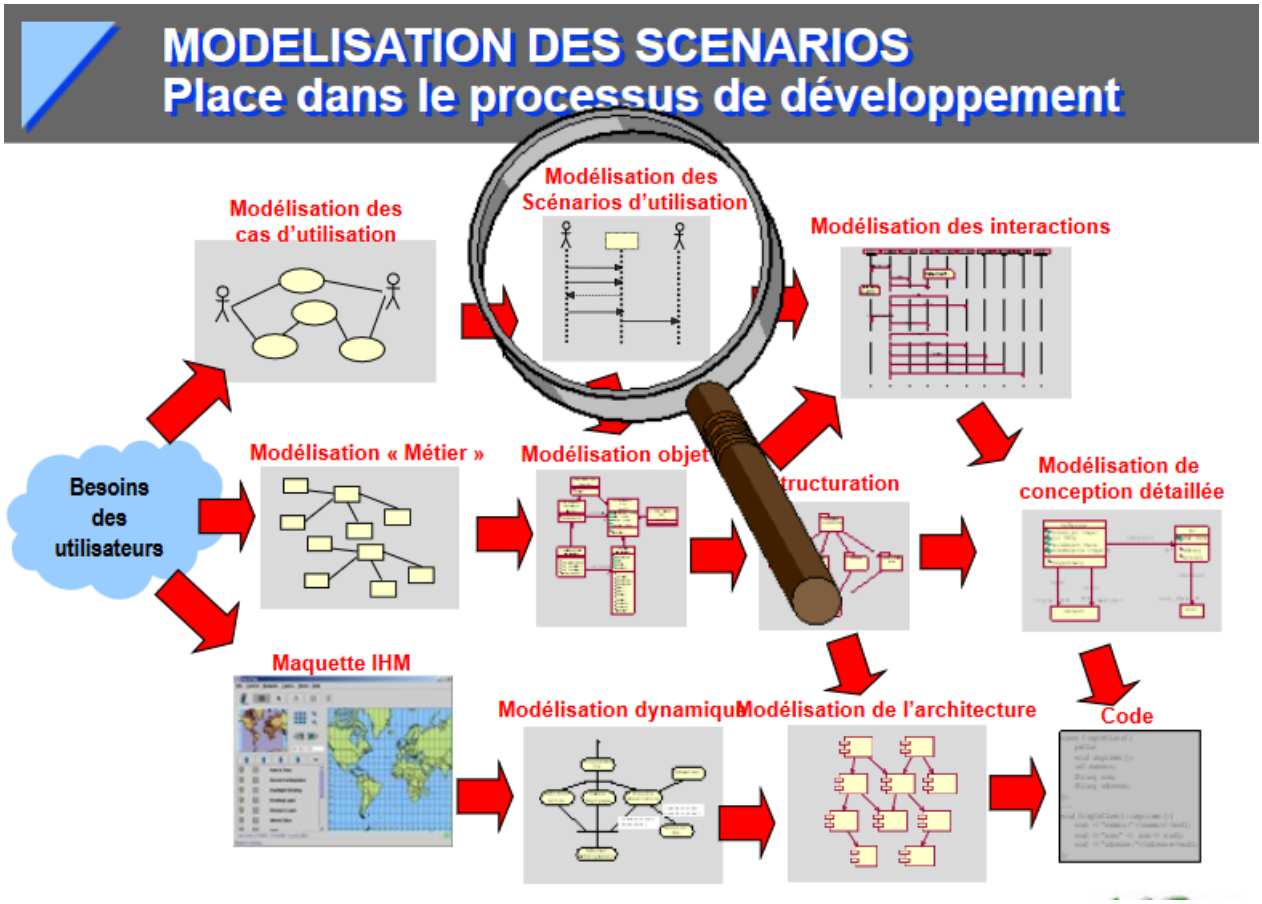
Source: https://fr.slideshare.net/IbenbrahimSoumaya/2-tup?next_slideshow=2

- ***2TUP*** : « 2 Track Unified Process» propose un cycle de développement en Y, qui permet de décomposer le système d'information, suivant un axe fonctionnel et un axe technique, puis fusionner les résultats de ces deux branches formant ainsi la lettre Y.

Principales caractéristiques de 2TUP :

- Il est itératif et incrémental :
  - On avance successivement d'étape en étape en se basant sur l'étape précédente.
  - Une itération apporte des améliorations et nous procure l'évolution.
- Il est piloté par les risques:
  - Les risques d'inadéquation aux besoins ou d'inadaptation technique.
  - Les risques d'incapacité à intégrer les technologies.
- Il est piloté par les exigences des utilisateurs
  - Le système doit pouvoir répondre techniquement et fonctionnellement aux besoins des utilisateurs (ceci fait partie des risques!)
- Un processus de modélisation avec UML
  - UML est le langage de modélisation standard de 2TUP. Un ensemble de diagrammes UML est utilisé dans des étapes du processus en Y. (Par exemple: Le Diagramme des cas d'utilisation est utilisé dans la capture des besoins fonctionnels et aussi la capture des besoins techniques) .
- Un processus centré sur l'architecture
  - le système est décomposé en modules pour des besoins de maintenabilité et d'évolutivité.

- Un processus orienté vers les composants:
    - Les regroupements de concepts définissent des packages et des composants dans le modèle, leur réutilisation peut se situer à tous les niveaux (lors de la capture des besoins fonctionnels, les cas d'utilisation sont regroupés en packages pour organiser le modèle de spécification fonctionnel et lors de l'analyse, les classes sont regroupées en catégories pour organiser successivement le modèle d'analyse métier et le modèle d'analyse de l'application)



source: http://slideplayer.fr/slide/1289506/.

13.8.2. Avantages et inconvénients

Avantages

- Séparation entre le fonctionnel et le technique au démarrage du processus (2 Track)
- L'utilisateur principal participe au cycle de vie du projet et tout glissement par rapport à ses exigences initiales ou tout changement sera possible en cours de conception
- Réalisation des fonctionnalités prioritaires pour démarrer

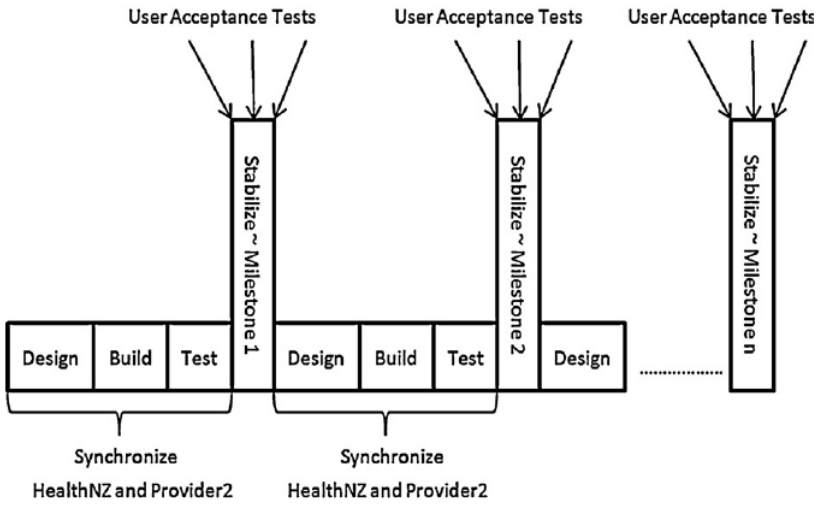
Inconvénients

- Les utilisateurs ou un utilisateur clé au moins doit être très disponible durant le cycle de vie
- Philosophie qui doit faire partie de l'organisation

13.9. Synch-and stabilize

13.9.1. Présentation

Fig. 1. Synchronize and stabilize lifecycle model.



Source: https://www.researchgate.net/figure/257001598_fig1_Fig-1-Synchronize-and-stabilize-lifecycle-model

Nous retrouvons à nouveau un processus incrémental.
Le projet global est divisé en incréments gérables dont la durée, pour chacun d'eux, est de 6 à 8 semaines (quoique rien de vraiment formalisé à ce niveau!). Ils sont traités par ordre de priorités décroissantes puis développés. A chaque incrément nous aurons des données complémentaires qui nous permettrons de mener à bien notre travail. La connaissance qui est accumulée durant le développement du logiciel servira à améliorer la gestion future du logiciel et à obtenir un produit final de haute qualité. Il est conseillé à ce niveau de créer une base de données de connaissances (mentionnant les problèmes rencontrés, les risques...)

13.9.2. Les phases

- Spécification des exigences
- Design du système
- Design détaillé
- Implémentation et Tests
- Intégration et Tests

Nous observons ici en fait nous avons un modèle en cascade (waterfall) qui est appliqué dans chacune des phases du modèle synch-ans-stabilize qui sont de petites itérations aisément gérables. La méthode synch-ans-stabilize hérite donc ses phases de développement du modèle waterfall.

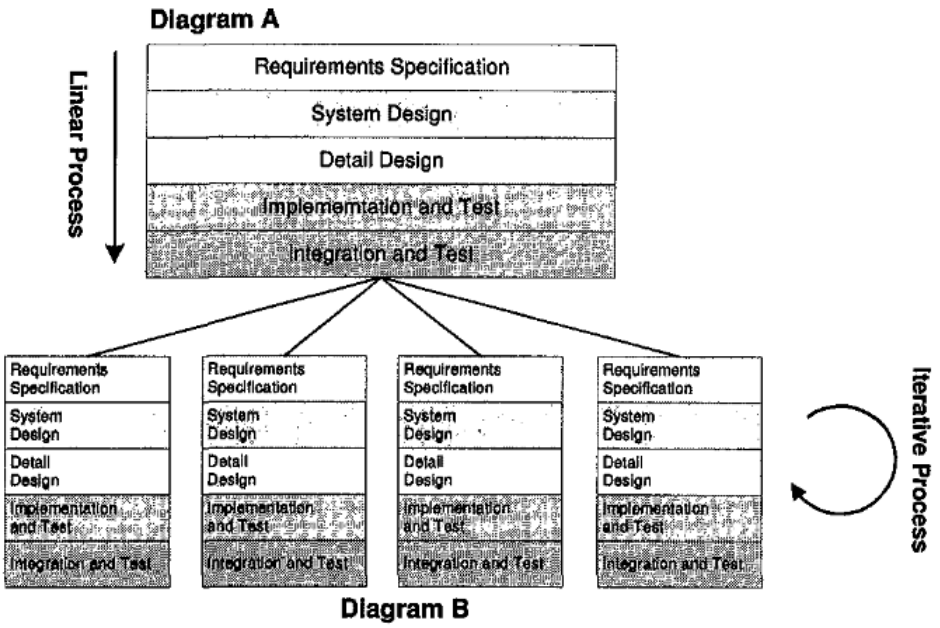


Figure 2.0: The waterfall and incremental models.

Source:
http://uir.unisa.ac.za/bitstream/handle/10500/592/dissertation.pdf;jsessionid=63D888E45E0E51C48282F7688292F438?sequence=1

Dans la figure ci-dessus, le bloc A illustre la séquence monolithique du modèle traditionnel de développement waterfall. Avec cette approche les spécifications sont gelées. L'implémentation du système final est suivie par un long cycle de tests et d'intégration.
Le bloc B illustre la division du modèle waterfall dans des incréments plus petits qui doivent être implémentés. Autrement dit, le système final sera un agrégat d'un certain nombre d'incréments finis et stables.

13.9.3. Question: qu'est-ce qui doit être synchronisé et stabilisé?

Un contrôle indispensable et un mécanisme de coordination du système est le 'daily build'. Le travail des développeurs est vérifiés tous les jours à une (ou des) heure(s) déterminée(s). Un 'build' est créé et des séries de 'smoke test' sont réalisés. Un 'smoke test' est un script de tests, ou des procédures de tests, qui testent les opérations basiques du système (tests manuels ou automatisés). Un objectif important des 'smoke tests' est de vérifier que chaque incrément de code du système est en harmonie avec le code existant. Si nous rencontrons un problème alors la présomption sera d'abord portée sur le code qui a été ajouté en dernier lieu. De cette manière le travail de l'équipe de développement est continuellement synchronisé et de façon régulière le système est stabilisé (par incrément). La stabilisation dépendra évidemment de la qualité du code qui a été ajouté pendant la synchronisation.
La phase de stabilisation s'occupe des problèmes rencontrés comme la performance, les ressources consommées par celui-ci et le fine-tuning. Si cette phase de stabilisation n'est pas bien gérée alors le danger est que le système se dégrade.

En résumé, le travail est exécuté en parallèle et en petites phases. Par exemple, les spécifications, le codage, les tests et l'intégration sont exécutées en parallèle. Ce modèle vise la rapidité du développement, sans compromettre la facilité d'utilisation et la qualité du produit final. Comme toute technique avancée de développement, elle requière une excellente coordination et accroît les responsabilités sur l'entièreté de l'équipe technique.

13.10. RUP - Rational Unified Process

Le RUP est une méthode mise au point par la société IBM Rational.

Le RUP est une approche du développement logiciel itérative, centrée sur l'architecture et pilotée par les cas d'utilisation.

- Méthode générique, itérative et incrémentale
- Centrée sur l'architecture dès les premières étapes du projet. Un système fonctionnel est bâti dès les premières itérations de manière à lever les risques techniques au plus tôt.
- Pilotée par les cas d'utilisation car toutes les fonctionnalités du système découlent des cas d'utilisation. Ceci garantit que seules les fonctionnalités qui offrent de la valeur ajoutée aux utilisateurs seront développées.
- RUP utilise UML (et sera certainement également abordée lors de ce cours!)

Cette méthode vous sera expliquée dans le cours UML

13.11. Extreme Programming (XP)

13.11.1. Introduction

Il s'agit d'une méthode AGILE, considérée par certains comme un phénomène de mode de par la profusion de littérature qu'elle a déjà engendrée, férocement défendue par ses adeptes aux quatre coins du globe, est devenue particulièrement populaire dans le milieu des développeurs qu'elle séduit en tout premier lieu. Elle se caractérise par quatre valeurs et douze pratiques que nous allons détailler ci-après sous l'angle architectural.

L'une des raisons de son succès est qu'elle produit de la valeur quasi immédiatement pour le client en implémentant rapidement ses scénarios utilisateur (*user story*). Ce chapitre décrit les grandes lignes de la méthode et s'attache surtout aux aspects architecturaux qu'elle impose.

13.11.2. Quatre valeurs

La méthode XP repose sur les quatre valeurs fondamentales suivantes :

- la **communication** (une place de choix est faite à l'oralité, considérée comme moins sujette aux incompréhensions que l'écrit), -
- la **simplicité** (toujours faire la chose la plus simple qui puisse marcher),
- le **feedback** (impliquer le client très tôt, prendre en compte ses impressions ainsi que le recul que peuvent nous donner nos collègues) et
- le **courage** (passer par les initiatives).

13.11.3. Douze pratiques

a) Développement piloté par les tests (TDD):

On parlera aussi de « test-first » design. Dès lors qu'on commence par développer les tests, on s'assure qu'ils échouent et seulement enfin on commence à coder pour faire passer l'ensemble des tests. Il y a deux types de tests : les tests unitaires implémentés par les développeurs qui testent le code et le design et les tests d'acceptance qui couvrent les scénarios et qui sont, dans la mesure du possible, écrits par le client. Le regain d'intérêt porté au test unitaire est en partie dû à cette pratique, particulière à XP. C'est un bienfait dont toutes les autres méthodologies peuvent profiter.

C'est la pratique la plus utile que cette méthodologie ait mise en place, elle libère la créativité du développeur en lui permettant d'explorer sans crainte les zones obscures du code, sans avoir besoin de réfléchir des jours durant à la future conception qui lui permettrait d'éviter de faire des erreurs qui, de toute manière surviendront tôt ou tard. Car même si l'on met tout en œuvre pour éviter les failles, il est quand même possible que quelque chose se passe mal. Alors la pratique du TDD permet de faire des erreurs et, à un certain degré, elle les encourage en tant que source d'innovation, car elle donne du feedback en permanence à l'équipe de développement.

b) Le jeu du planning:

Le projet est décomposé en courtes itérations d'une à trois semaines dans lesquelles le client décide des scénarios à implémenter. Le client définit le contenu, le planning et les priorités tandis que les membres de l'équipe technique estiment et définissent l'ordonnancement des tâches.

c) Client dans l'équipe:

XP considère comme un client toute personne qui utilisera réellement le système une fois produit. Il est recommandé que ce client soit physiquement présent lors des meetings pour résoudre toutes les ambiguïtés de scénario et définir les priorités.

d) Petites livraisons:

L'objectif est de mettre en production très rapidement un système simple et de livrer ensuite les nouvelles versions par cycles très courts. L'idée est de fournir au client les fonctionnalités qu'il souhaite voir dans le produit final pour qu'il puisse jouer avec elles très rapidement et faire des retours du type : « cela devrait marcher plutôt comme ceci, ou alors ne pourrait-on pas plutôt présenter cela de telle manière ».

En pratique, cela donne un prototypage itératif dans lequel le client (ou utilisateur final) agit en qualité de marionnette de crash test. L'inconvénient d'une telle pratique est qu'en définitive, cette phase d'expérimentation opérationnelle peut durer longtemps, jusqu'à excéder parfois le temps d'une livraison programmée par une méthode moins agile et parfois même mener à l'abandon du projet…

e) Usage des métaphores:

La métaphore du système est un scénario global imageant le fonctionnement du système. Cette histoire originelle induira le nommage des objets du domaine et des composantes du design ; elle est censée permettre à l'équipe d'harmoniser ses nomenclatures. Cette métaphore est vue comme un aspect fondamental de la communication dans un projet XP. En effet, elle doit servir de guide à l'équipe pour trouver la bonne architecture et bien communiquer la structure du logiciel avec le client.

L'usage des métaphores en langage naturel pour un projet XP est un art délicat et il s'avère en pratique de peu d'utilité dans la communication de l'équipe. À cela plusieurs raisons, dont le fait que les développeurs sont principalement des gens techniques et qu'ils n'ont pas forcément le champ sémantique approprié à la métaphore du métier pour lequel ils produisent un système d'information. L'une des autres causes est le côté imagé qui ne produit pas les mêmes représentations mentales chez l'individu en fonction de sa culture et de son expérience.

f) Design simple:

Toujours faire au plus simple. Un design simple est plus facile à coder et à maintenir qu'un design complexe.

g) Refactorisation à outrance:

La refactorisation est la refonte permanente de la base de code existante. Les batteries de tests unitaires servent de garde-fou en garantissant qu'aucune régression n'est introduite dans le système malgré les modifications du design. Les tests doivent continuer de passer après chaque refonte du code et le design doit être simplifié à l'extrême. Cette pratique est rendue particulièrement nécessaire par le fait que XP interdit tout design préliminaire et qu'il y aura de cette façon de grosses portions de code à reprendre pour obtenir un design propre.

h) Propriété collective:

Chacun est responsable du code tout entier. Si un programmeur identifie une portion de code qui ne lui convient pas, il peut la changer à sa guise du moment que les tests continuent de passer. Ceci implique une rotation constante des développeurs sur toutes les zones techniques en dépit des préférences ou domaines d'expertise.

i) Programmation par binômes:

Voilà une autre pratique de la culture extrême qui peut sembler exotique à ceux qui ne l'ont jamais pratiquée. C'est pourtant un moyen efficace d'intercepter les erreurs et d'identifier les moyens de simplifier le design. Plus qu'une recommandation, c'est une obligation. Car selon Robert C. Martin : le code mis en production doit avoir été écrit par une paire de développeurs, les préférences et le confort de ceux-ci ne doivent pas prendre le pas sur la qualité du code livré !

j) Intégration continue:

Le code doit être compilé et testé à chaque modification de la base et la base doit être saine chaque matin.

k) Rythme de travail raisonnable:

Un travailleur fatigué fera plus d'erreurs qu'un travailleur frais et dispos, c'est pourquoi XP encourage un rythme continu mais pas effréné afin de ne pas éreinter son équipe par des courses-poursuites à travers les plannings sur des livraisons surréalistes.

l) Conventions de nommage:

Elles doivent être les mêmes pour chacun des membres de l'équipe afin de garantir la compréhension universelle du code. En mode XP, la rotation des équipes encourage encore plus cette pratique qu'ailleurs.

13.11.4. Particularités

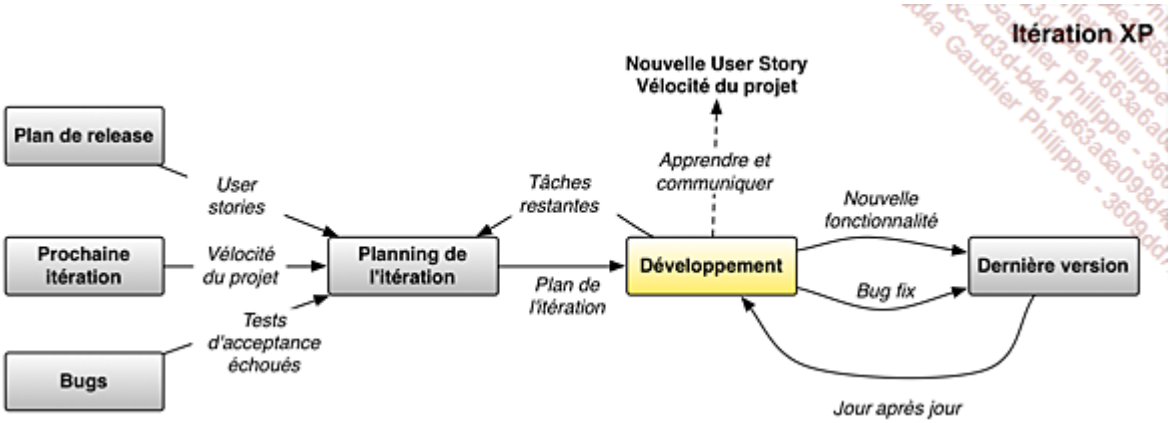
a) Humanisme:

L'une des raisons de la popularité de cette méthode est son aspect humain, qui procure souvent une ambiance amicale dans l'équipe de développement. La pratique du développement en binôme (*pair programming*) et la propriété collective du code participent activement à cette vision humaniste du travail.
Cet aspect est à double tranchant : s'il peut d'un côté plaire aux développeurs de moins bon niveau car leurs bugs seront tôt ou tard découverts et corrigés par d'autres, sachant que les équipes explorent la plupart des zones du code, il peut aussi déplaire à certains, et sans doute les meilleurs éléments, qui peuvent avoir l'impression de travailler pour les autres et d'être bridés dans leur expertise.
Ce point de la méthodologie l'écartera de facto des organisations qui se focalisent sur les talents et qui opteront plutôt pour Scrum ou RUP.

b) Vélocité:

C'est une mesure qui représente la vitesse à laquelle un projet est accompli par une équipe. Une sorte de vitesse à laquelle de la valeur métier est ajoutée au projet. Ce n'est pas seulement la mesure du temps passé par l'équipe mais plutôt une quantification de valeur du point de vue du client. L'unité de mesure est le scénario (*user story*). Lors de l'étape de planification, on attribue à chaque scénario utilisateur un nombre de points suivant sa complexité et l'ampleur de la tâche. À la fin d'une itération, on totalise les points des scénarios réalisés, ce qui donne la vélocité du projet pour cette itération.

Cette mesure permet de se projeter assez fidèlement dans le temps car on peut connaître la vélocité moyenne de l'équipe sur un projet et estimer fidèlement le contenu d'une itération à venir.



c) YAGNI

« *You aren't gonna need it !* » ou « vous n'aurez pas besoin de cela ! » est le crédo du gourou XP. C'est ce qui simplifie la vie du développeur qui n'a plus autant besoin d'anticiper, mais c'est aussi ce qui rend si difficile l'intégration d'un architecte au sein d'une équipe XP. Comment faire comprendre à quelqu'un d'ordinaire en charge du design pour les futurs développements que ce document n'a pas lieu d'être car il fait désormais partie du code ? L'approche « modéliser pour anticiper » est fondamentalement chamboulée par les principes XP et son design émergent au fil des itérations.

d) Espace

Lorsqu'on fait appel à un coach agile pour introduire XP dans son organisation, il va se heurter à bon nombre de réticences. Mais la plus tenace, celle qui lui demandera le plus d'inventivité pour parvenir à l'acceptation du changement, c'est de loin la reconfiguration de l'espace.

Le travail en binôme, la rotation sur les différentes zones du code, la propriété collective sont autant de concepts qui sont antithétiques avec un travail dans un milieu cloisonné et intime. Il faudra donc reconfigurer tout ou partie de l'espace physique de travail et (malheureusement) le dépersonnaliser. Finis les bureaux vitrés tapissés des photos de famille où s'entassent divers éléments de la sphère privée ! Place à l'open space, aux vastes bureaux dégagés et aux casques de musique pour ceux qui veulent de l'isolement. On voudra désormais que les équipes communiquent et que chacun sorte de son isolement. C'est dire si la mutation qu'on demande aux habitudes est profonde.

13.11.5. Conclusion

Tout est possible, évidemment, chacun prendra ses responsabilités quant à ses choix méthodologiques. Toutefois, il est dans les attributions de cet ouvrage de mettre en garde les futurs adeptes d'XP et des méthodes agiles. Ces méthodes sont applicables à du développement web ou à des applications à faible criticité, ou qui nécessitent un Time To Market minimal ou du prototypage opérationnel. Cependant, une organisation devra sérieusement hésiter à appliquer ce choix pour la production d'applications plus critiques comme des systèmes temps réel, de la sûreté aérienne, des systèmes d'échange bancaires, des pilotes automatiques…

On évitera donc les déviances extrémistes dans l'un ou l'autre des 12 principes.

13.11.6. Avantages et inconvénients

Avantages:

- Gain de temps et d'argent: ceci est dû à:
  - la réduction de lourdes documentations au profit des échanges oraux (la communication est un atout dans XP)
  - aux dates fixées pour l'atteinte des objectifs
  - simplicité du design et du code (que l'on pourra étoffer plus tard si désiré)
- responsabilisation de tous les membres de l'équipe
- contact continu avec le client final
- changements possibles au niveau des exigences (retour en arrière)
- répond aux attentes du client et rien de plus (ni de moins!)

Inconvénients:

- nécessite un grand investissement en temps du client
- Design trop simplifié dans certains cas (surtout quand il s'agit de produits destinés à la vente)
- Philosophie difficile à implémenter dans une organisation (résistance au changement!). On adopte tout ou rien dans XP!
- Manque de documentation, qui pourrait poser problème dans le futur, surtout dans les projets de grandes envergures (en cas de mouvements de personnel par exemple)

13.12. SCRUM

13.12.1. Introduction

Cette méthode est également une méthode AGILE qui est proche de la méthode XP.

SCRUM dérive de la théorie des systèmes adaptatifs et fut influencé par les pratiques de l'industrie japonaise, en particulier par les principes de développement LEAN ainsi que par les stratégies de gestion des connaissances de Takeuchi et Nonaka.

Scrum n'est pas à proprement parler une méthodologie ou un processus, il s'agit plutôt d'une compression des meilleures pratiques issues de 50 ans de développement logiciel. On parlera d'un cadre de travail (Framework) simple qui implémente les bonnes pratiques du Manifeste Agile. De plus, l'une des singularités de Scrum est qu'il n'est pas applicable qu'au seul domaine informatique, mais à bien d'autres comme l'organisation d'un mariage, la gestion d'un projet de paysage…

C'est une méthode visant à produire de meilleurs logiciels plus rapidement et plus agréablement, aujourd'hui adoptée sur des projets notables comme par exemple Google AdWords qui est entièrement issu de cette méthodologie.

13.12.2. Concept

Scrum est pensé pour ajouter de l'énergie, de la clarté et de la transparence au projet en termes d'implémentation et de planification. L'usage de cette méthodologie aura pour effets:

- De démultiplier la vitesse de développement.
- D'aligner les objectifs individuels et ceux de l'organisation.
- De créer une culture pilotée par la performance.
- De fournir un haut niveau de communication de la performance à tous les niveaux.
- D'améliorer la qualité de vie et le développement personnel.

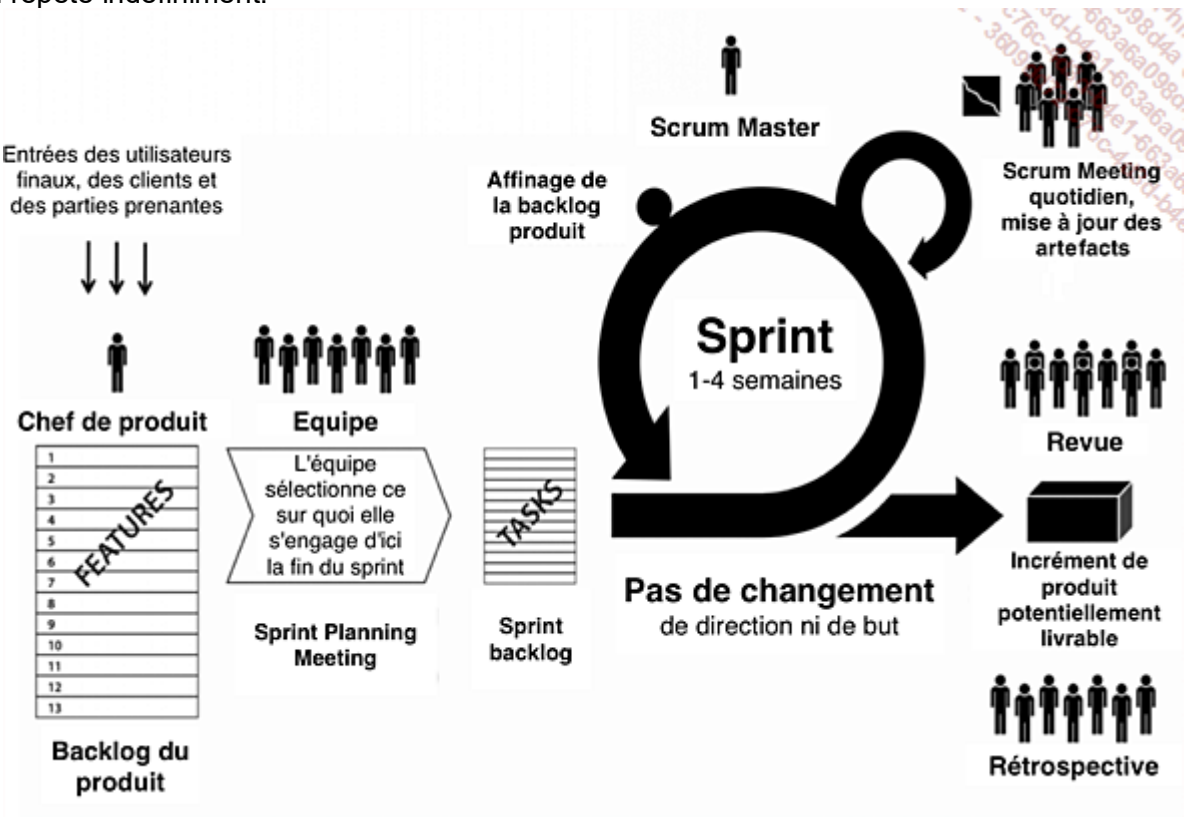
Dans le même esprit que XP, avec Scrum on pense qu'il est inutile voire nuisible de spécifier le produit en amont, d'en faire un bon design préalable, car il est humainement impossible de tout prévoir et qu'une bonne idée survenant dans les dernières étapes du projet serait plus de l'ordre de la menace que du bienfait.
En outre, les processus classiques de type *Waterfall* ou *Spirale* se sont avérés source de souffrance morale pour les équipes qui subissent des plannings établis plusieurs mois à l'avance en dépit des aléas du projet.
De plus, Scrum est censé permettre de répondre aux changements d'humeur du client qui comprend toujours trop tard ses besoins exacts.

13.12.3. Processus

Scrum est un framework itératif, incrémental pour la création d'applications logicielles. Il structure le développement en cycles de travail appelés **sprints**. Ces itérations ont une durée d'une à quatre semaines. Ces sprints, qui ont une durée fixe non étirable, s'arrêtent quel que soit l'état du développement en cours (tâches terminées ou non). Au début de chaque sprint, les équipes sélectionnent des **items** parmi une liste priorisée (exigences client) qu'ils doivent implémenter d'ici la fin du sprint. Durant ce sprint, les items choisis seront immuables (pas sujet au changement).

Tous les jours, les membres de l'équipe s'entretiennent rapidement sur leur progression respective. À la fin du sprint, l'équipe expose son sprint aux parties prenantes du projet et montre ce qui a été réalisé. Ils obtiennent ainsi un feedback qu'ils pourront introduire dans le sprint suivant.

Scrum met en valeur la partie opérationnelle de la fin du sprint, c'est-à-dire le code fonctionnel, testé et livrable contenu dans le produit.
Un thème majeur de Scrum est « *Inspect and Adapt* ». Comme le développement implique inévitablement de l'apprentissage, de l'innovation et des surprises, Scrum encourage à prendre une courte portion d'itération pour inspecter le produit résultant ainsi que l'efficience des activités courantes (bon fonctionnement) pour ensuite adapter les objectifs et le processus.
On répète indéfiniment.



13.12.4. Rôles

En Scrum il y a trois principaux rôles:

- **le directeur de produit (Product Owner)**
- **l'équipe (team)**
- **le Scrum Master**

1. Product Owner

Le directeur de produit a pour objectif de maximiser le retour sur investissement (ROI) en identifiant les fonctionnalités du produit, de consigner ces fonctionnalités dans une liste priorisée, décidant laquelle devra se trouver en tête de liste du prochain sprint et de continuellement enrichir et ordonner cette liste. Il est directement responsable des pertes et profits du produit vu comme une entité commerciale. Maximiser le ROI veut dire sélectionner pour le prochain sprint les items à forte valeur ajoutée et à moindre coût de production. Il n'y a qu'une seule personne qui assumera ce rôle et l'autorité finale qu'il confère. On assimilera le directeur de produit au chef de produit des organisations classiques, à la différence qu'il interagit fréquemment avec l'équipe.

2. Team

L'équipe construit le produit dont l'utilisateur finira par se servir, elle propose aussi des idées au directeur de produit afin d'améliorer ce dernier. L'équipe qui est capable d'intervenir sur toutes les zones fonctionnelles, s'auto-organise avec un grand degré de liberté. Il n'y a donc pas de chef de projet. L'équipe seule décide quoi faire et comment implémenter au mieux un item. En Scrum, une équipe se compose de sept (plus ou moins deux) membres, incluant développeurs, analystes, designer d'interface et testeurs. Une équipe est allouée 100 % à un produit pour chaque sprint car il est démontré que des équipes stables gagnent en productivité. Pour de gros projets, on aura de multiples équipes coordonnées et assignées à des groupes fonctionnels distincts. Scrum encourage la pluridisciplinarité et la transmission des compétences et en général les membres vont là où il y a du travail en apprenant des autres s'ils interviennent sur des domaines qu'ils maîtrisent moins bien.

3. Scrum Master

Le Scrum Master accompagne et anime l'équipe afin qu'elle remplisse au mieux ses objectifs. Ce n'est en aucun cas le chef de l'équipe ou du projet, il est au service de l'équipe, la protège des interférences extérieures, éduque et guide le directeur de produit et l'équipe dans ses compétences en Scrum. Il s'assure que toutes les parties prenantes du projet comprennent et maîtrisent les pratiques de Scrum. À la différence d'un chef de projet, il n'assigne pas de tâches ni ne dit ce qu'il faut faire à l'équipe, il va plutôt les guider dans une amélioration de leurs pratiques d'auto-organisation. Il est d'ailleurs rare qu'un ancien chef de projet fasse un bon Scrum Master tant ces deux rôles sont éloignés dans leurs pratiques.

13.12.5. Backlog

La backlog (terme difficile à traduire) du produit constitue sa feuille de route, elle évolue tout au long du projet et s'apparente à un cahier des charges fonctionnel. C'est un recueil d'items qui sont estimés (en termes de valeur métier et d'effort), priorisés et assignés à un sprint. Estimer la valeur métier d'un item n'est pas chose facile, et c'est un des rôles du Scrum Master d'aider le directeur de produit à apprendre à le faire. Ces items peuvent être des exigences utilisateur, des défauts à corriger, des améliorations non fonctionnelles, des modifications d'architecture…

De la même façon qu'avec XP, une exigence est souvent formulée en termes de scénario d'utilisation (par exemple « en tant qu'apiculteur je voudrais faire transhumer mes ruches d'un lieu A vers un lieu B ») en exprimant le point de vue de l'utilisateur.
Ce document est maintenu en permanence par le directeur de produit pour refléter les dernières requêtes utilisateur, les changements de la concurrence, les nouvelles idées, les problèmes techniques qui apparaissent en cours de route. Un sous-ensemble des items qu'il contient constitue la **Release Backlog** ou liste des fonctionnalités destinées à sortir en production.

De même qu'en XP, les estimations sont exprimées en points plutôt que dans une unité absolue comme l'homme/jour. Avec le temps, une équipe saura dire combien de points elle est capable d'implémenter par sprint. Cette mesure permet alors de savoir au bout de combien de temps un ensemble d'items peut être implémenté.
Le niveau de détail de la rédaction des items varie en fonction des priorités et de la visibilité de l'équipe. Il ne sera pas nécessaire de détailler des exigences qui ne seront implémentées que trois sprints plus tard, pas plus qu'une optimisation qui fait l'unanimité dans l'équipe.

13.12.6. Sprint Planning

Au début de chaque sprint a lieu le sprint planning meeting, divisé **en deux réunions distinctes**.

La première partie réunit le directeur de produit et l'équipe pour passer en revue l'ensemble des items de plus haute priorité. On discute alors des objectifs et du contexte de ces items importants en donnant des éclairages sur la vision du directeur de produit. On définit aussi le qualificatif « fait » pour un item : respect des standards, implémenté par du TDD, 100 % des tests automatiques passés, intégrés, documentés. Cette partie se concentre sur la compréhension de ce que veut le directeur de produit.

Lors de **la deuxième partie** du meeting, on détaille les tâches qui permettront d'implémenter les items que l'équipe s'engage à implémenter d'ici la fin du sprint. Elle s'assigne elle-même les tâches et quantifie le volume d'effort nécessaire à leur accomplissement, ce qui donne un engagement bien plus fiable que s'il avait été imposé par le directeur de produit. L'équipe dépile les items classés par criticité, mais peut quand même en choisir d'importance moindre si elle les considère liés à certains déjà choisis. Lors de ce planning, les membres de l'équipe estiment chacun le temps réellement disponible pour

le travail du sprint (vacances et temps de réunion, d'e-mail, de formation et autre retranchés), ce qui en général donne dans les 4 à 6 heures par jour de présence.

Ensuite, chaque item est découpé en tâches individuelles qui sont insérées dans le Sprint Backlog et estimées en heures de travail. Lorsqu'il n'y a plus de temps disponible pour le sprint, on arrête de dépiler les items et l'engagement devient ferme et définitif jusqu'au prochain sprint.

13.12.7. Daily Scrum

L'une des autres pratiques clés de Scrum est un meeting quotidien très court, auquel on assiste debout et au cours duquel chaque membre de l'équipe communique sur son travail.

L'un après l'autre, les membres de l'équipe exposent trois et seulement trois points aux autres :

- Ce qu'ils ont accompli depuis le dernier meeting.
- Ce qu'ils ont prévu de faire d'ici le prochain meeting.
- Les points de blocage rencontrés.

Il ne s'agit pas d'un meeting de management, mais d'une phase d'auto-organisation. Le Scrum Master est d'ailleurs utile pour aider à dénouer les points de blocage.

13.12.8. Burndown Chart

Chaque jour, les membres de l'équipe mettent à jour leurs estimations du temps restant pour accomplir la tâche courante. Quelqu'un consolide ces nouvelles données sur un graphe pour toute l'équipe. Si tout se passe bien - ce qui est rarement le cas -, la courbe doit tendre asymptotiquement vers le « zéro effort ». L'important est de montrer à l'équipe sa progression vers l'objectif en termes de « combien d'effort reste à produire » pour parvenir au but. Si cette courbe ne tend pas vers zéro en fin de sprint, c'est signe qu'il faut ajuster le périmètre des items du sprint ou revoir les méthodes de travail ou les technologies. Là encore, le Scrum Master est un allié précieux pour l'équipe pour analyser et comprendre les sources de mauvaise estimation.

13.12.9. Bénéfices

Un des points fondamentaux de Scrum - qui protège l'équipe des changements d'avis intempestifs des directeurs de produit -, c'est que ce dernier ne peut apporter aucune modification aux items sélectionnés dans sa backlog en cours de sprint. Il est libre de modifier ce qui lui chante dans les autres items, mais ne touchera plus à ceux qui sont en cours de production. Toutefois, un sprint peut être interrompu par le DP si des circonstances particulières font que l'équipe perd son temps et qu'il vaut mieux abandonner.
En contrepartie, l'équipe choisit ses items, s'organise et s'engage à les livrer en fin de sprint, ce qui donne une garantie de résultat au directeur de produit.

13.12.10. Avantages et inconvénients

Avantages:

- SCRUM assure une utilisation efficace du temps et du budget
- Les projets de grande envergure sont divisés en sprints gérables facilement
- Les développements sont réalisés et testés durant la revue de sprint
- Cette méthode fonctionne pour des projets à développements rapides
- L'équipe reçoit une information continue et claire durant les meetings
- SCRUM travaille sur base des feedbacks des clients et des stakeholders
- Les sprints de courte durée permettent les changements (basés sur le feedback) beaucoup plus facilement
- L'effort individuel de chaque membre de l'équipe est visible durant les meetings journaliers

Inconvénients:

- SCRUM mène souvent à des glissements de périmètre dû au manque de fixation d'une date de fin de projet
- Les chances d'échec du projet sont grandes s'il y a un manque de coopération de tout ou partie des membres de l'équipe
- Adopter SCRUM dans de grandes équipes est difficile
- Ce cadre de travail ne sera un succès qu'avec des membres d'équipe expérimentés
- Les meetings journaliers peuvent frustrer les membres de l'équipe
- Si un ou plusieurs membres de l'équipe projet quittent le projet lors de son exécution, cela provoquera un impact énorme (négatif) sur le projet
- L'implémentation de la qualité est difficile

En conclusion

Des plannings corrects et des prises de décisions intelligentes peuvent vous aider à vous débarrasser de ces désavantages. Par exemple dans des équipes plus grandes, chaque membre doit avoir un rôle défini et des responsabilités avec des objectifs bien définis, ainsi il n'y aura pas de compromis sur la qualité et pas d'excuse à l'échec. Ceci maintiendra l'équipe concentrée sur les objectifs du projet. De plus, le SCRUM master doit diriger l'équipe de manière efficace pour éviter les pièges et assurer à 100% le succès du projet!

14. Bibliographie

- IT Project Management – Formation certifiée SELOR suivie par l'auteur
- Projets informatiques – S'approprier le guide PMBOK pour réussir sa gestion de projet
- Différentes références Internet mentionnées dans le présent syllabus