Sommaire  
 I) Introduction  
 II) Étapes  
 1) WBS  
 2) PERT   
 3) GANTT  
 4) Priorisation  
 5) Gestion des risques  
 III) Équipe  
 IV) Cahier des charges

**GESTION DE PROJET**

1. Introduction

Un projet est constitué de 4 "parties" :

* le client apporte le projet (un besoin, un objectif) : périmètre
* une notion de délais/temps (=étapes)
* un budget (on parle d’enveloppe budgétaire)
* une partie gestion des risques.

Durant toute sa durée de vie, deux équipes vont travailler sur le projet :

* équipe projet = BUILD, est une équipe (temporelle normalement) qui vient travailler sur le projet, crée la première version
* équipe d’exploitation = RUN = MCO (maintient en conditions opérationnelles) fait tourner le logiciel (maintenances…)

Durée de vie du projet

équipe projet

v2

v1.1

équipe exploitation

projet kill

v1.2

v1

exploitation v2

exploitation v1

Il existe une équipe (occupation) qui commence à voir le jour : le DEVOPS, qui a pour mission de suivre l’équipe projet pour ensuite aider l’équipe d’exploitation le prendre en main. (Ils font échanges build/run)

1. Étapes

Voici, selon William GUISSEFFI, une liste possible des étapes (=checklist) d’un projet :

* Analyser le besoin (et donc livrables = donne au client)
* Lister et décomposer les taches nécessaires au projet (WBS)
* Établir les responsabilités pour chaque tache
* Définir l'ordre des taches et leur durée (PERT)
* Répartir les activités dans le temps (=planning) (GANTT)

On doit parallèlement gérer :

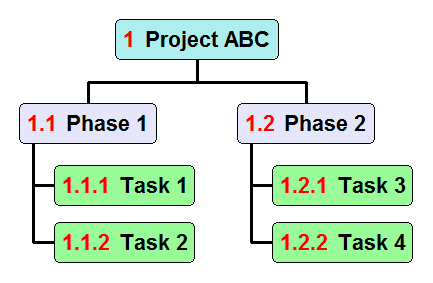
* les couts
* la priorisation
* les risques

La gestion des risques est primordiale pour la réussite d’un projet.

1. WBS

Le WBS (work breakdown structure = organigramme des taches) :

* décomposition hiérarchique des taches à effectuer. Le projet est à la racine.
* Pour chaque activité/tache, on doit pouvoir :
  + contrôler sa durée
  + définir les ressources nécessaires
  + connaître son coût
  + calculer le résultat



(Tiré de mpug.com)

Le projet est ABC.

Phase 1 et Phase 2 sont les tâches mères (cf. GANTT), c’est-à-dire une grande tâche à réaliser.

Task1,… sont des sous-tâches composant la grande tâche.

Généralement, les coûts en en homme jours (HJ) sont inclus pour chaque tache, en commençant par les sous-tâches. (On peut y mettre aussi les coûts en argent).

1. PERT

Après votre WBS réalisé, on peut s'en servir pour faire un diagramme de PERT (=diagramme d'états-transitions = automate).

L'état initial est le projet, les transitions sont les taches et les états sont tache x terminée par exemple...

Remarque : Pour représenter le fait qu'un tâche est plus longue qu'une autre mais que l'on doit attendre deux taches qui se font en parallèle alors elles sont reliés à l'état suivant avec une flèche en pointillé.

Note : A, B, … Sont des tâches.

E

A

H

B

C

D

FF

Pour juger l’avancement, Il existe la courbe à 45 : on choisit des jalons (cf. GANTT) et on leur fixe une date, toute les semaines, on met à jour la date auquel sera fait le jalon et on voit le retard.

1. GANTT

Enfin, GANTT traduit le diagramme de PERT pour créer un planning.

* liste de toutes les taches (suite ordre de PERT)
* Identifie les taches filles (par exemple, Task1… du WBS)
* Créé les dépendances entre les taches (PERT & WBS)
* Donne dates et durées
* On ajoute les JALONS (=événement) ex: signature d'un contrat
  + on parle également de dates buttoir intermédiaires
* Ajout des personnes

Et enfin, au fur et à mesure, on met à jour l'avancement (fait/à faire)

Outils : [GanttProjet](https://www.ganttproject.biz/download) (logiciel), [officetimeline](https://online.officetimeline.com/) (site web, création en ligne).

1. Priorisations

Généralement, toutes les tâches ne seront pas réalisés, de nouvelles pourront être ajoutés, on utilise la notation MoSCoW pour prioriser :  
 Must , Should, Could, Would allant de nécessaire à facultative.

Alternativement, il existe la matrice d’EnseinHower qui fonctionne sur le même principe (Faire, Déléguer, Planifier, Éliminer).

1. Gestion des risques

On commence par évaluer les risques :   
cause (%chance d’arriver) => risque => effets/conséquence (% gravité)  
  
Les % sont sur le même principe que MoSCoW (25, 50, 75,100) et on évalue le risque en faisant le produit des deux probabilités.

On commence généralement par traiter les risques mineurs.

(faire un tableau de suivit des risques)

1. Équipe

Les points importants à gérer (Chef de projet, ou autres) :

* Collaboration (->outils), partage d’informations, échanges
* Intérêt de chacun
* Recherche de l’erreur (pas du coupable)
* Motivation, adhésion des membres de l’équipe
* Gestion des conflits

1. Les réunions

On prévient avant la réunion, les personnes qui vont intervenir, et combien de temps pour qu’elles puissent se préparer. (Ordre du jour)

Le but est de partager l’avancement, les points bloquants et de trouver des solutions.

On rédige un compte rendu (CR) de réunion qui contient les infos échangées, les décisions prises, les actions à suivre (note personnes responsables de leur suivit) et la date de la prochaine réunion.

Date HeureDébut Lieu Durée  
Participants  
Ordre du jour : sujet X 15’ (Mr …), sujet Y 30’ (Mme …), …  
Infos échangés  
Décisions  
Plan d’action :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Description | Responsable | Échéance livrable | Validations par |
|  |  |  |  |  |

Date de la prochaine réunion

On distingue souvent 2 types de réunions, les réunions du comité de pilotage qui ont lieu par exemple 1 fois par mois, des réunions de suivit qui ont lieu par exemple 1 fois par semaine.

Dans les réunions de pilotage, on aurait par exemple, le client qui vient suivre l’avancée du projet, alors que dans les réunions de suivit, on détermine ce qu’on a fait, ce qu’il reste à faire…

1. Le reporting

On partage régulièrement l’information, structurée de la même manière, pour tenir informé les personnes qui par exemple n’ont pas pu être présentes à une réunion.

On donne une vision synthétique du projet :  
 - l’objectif et les enjeux  
 - indications sur l’avancement  
 - précisions sur les points compliqués

Attention ! Un point clef est de toujours avoir une version qui marche !

1. Psychologie de l’équipe (motivation, adhésion, …)

Durant le projet, des changements plus ou moins importants peuvent avoir lieu, que ce soit sur le programme ou dans les membres de l’équipe.

Cela nécessite un effort d’adaptation, une possible surcharge pour les membres. On représente ces notions par la courbe du changement.

Dénis

Résistance

Engagement

Exploration

Dénis : refus de changer  
Résistance : négociation, marchandage  
Exploration : créativité, recherche de soi, changement déclenché  
Engagement : Motivation de l’entourage, performances en croissance

On distingue 3 types de personnes face au changement :

* les opposants (contre) : 10%
* les proactifs (pour) : 10%
* les passifs (attendent des preuves) : 80%

1. Le cahier des charges

Le cahier des charges a pour but d’aider à comprendre les besoin. Il est composé généralement :

* une présentation de l’entreprise, du projet ainsi que de l’équipe.
* liste des livrables.
* d’une liste de taches à faire
* de spécifications (fonctionnelles et techniques)
* une éventuelle charte graphique

Une représentation possible des tâches à faire :

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N° | tâche | Délais | Charge prévisionnelle (hj) | Priorité |
|  | Phase 1 | 20j/5s | 85 (=60 + 25) | M |
| A | Task 1 | 8j/2s | 25 | M |
| B | Task 2 | 12j/3s | 60 | C |

Remarque : Phase 1 est le tâche mère qui comprends deux sous-tâches task1 et task 2 (repris le WBS précédent). J’ai mis des priorités au hasard.

1. Spécification Fonctionnelle (+périmètre fonctionnel)

Il s’agît d’une liste des fonctionnalités (que fait l’application), on attend pour chaque fonctionnalité, si besoin :

* maquettes (wireframe)
* des diagrammes UML
* des phrases claires qui
  + décrivent la fonctionnalité
  + les potentielles contraintes que devra gérer l’équipe de développement (+ vérifications qu’ils devront gérer)
  + comment est l’apparence (cf. wireframe)
  + comment on accède à la fonctionnalité
  + les détails que le programmeur devra gérer (ex : envoi d’un email 1/semaine avec infos aux clients…)

1. Spécification Technique

Il s’agît des aspects plus complexe de l’application ce qui comprends :

* langages utilisés, framework… (ne pas oublier les versions !)
* logiciels
* les contraintes techniques (plateformes, navigateurs, appareils)
* services tiers utilisés (Google analytics, …)
* l’architecture (organisation des dossiers, des menus)
* conventions d’écriture, de nommage
* Modèle de données (éventuels schéma de base de données), méthode de transfert (Post, Get, …)
* nom de domaine pour un site web et emails