

### Qu'est-ce qu'un projet ?

- Satisfaction d'un besoin spécifique et particulier
- Se déroule dans un contexte précis
- Unique dans le temps et l'espace (caractère novateur, au moins partiel)
- Daté, marqué par un début et une fin
- Autonome : il se suffit à lui même
- Mobilise des ressources d'une façon ponctuelle
- Aboutit à un résultat : produit ou service

### Principales étapes dans un projet

- Recueil et analyse des besoins
  - interview (questions ouvertes et fermées), recherche documentaire
  - Analyse des besoins (synthèse, cohérence, complexité, faisabilité)
  - Evaluation des problèmes techniques, recherche de solutions et d'informations complémentaires
  - Diviser pour réduire la complexité
  - Définir une stratégie de réalisation (priorités, organisation)

- Etude de faisabilité
- Finalisation : Cahier des charges fonctionnel, dossier des besoins et backlog produit
- Construction de la solution
  - Conception de logiciel
  - Implémentation et tests
    - \* Spécifications des briques de base
    - \* Codage des différents éléments et tests unitaires
    - \* Liens entre les briques
    - \* Tests de non régression
  - Documentation technique (dont les commentaires du code)
  - Dossier de conception
  - Dossier de spécification
  - Dossier de tests
- Livraison, évolution et maintenance
  - Produit finalisé
  - Documentation complète
  - Elements utiles pour évolution et maintenance
  - Bilans et retour d'expérience
  - Présentation orale finale

**Tâches :** Les étapes s'enchainent et peuvent se répéter en fonction des choix méthodologiques retenus. Elles se découpent en tâches de plusieurs types

- Uniques
- Répétitives
- Séquentielles
- Parallèles
- Collaboratives
- ...

Elles doivent être

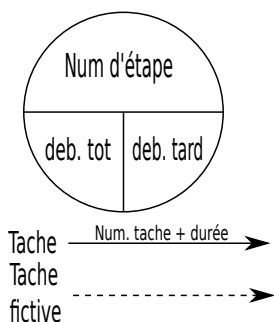
- Nommées
- Evaluées (temps, ressources, coût)
- Organisées (dépendances, priorités, ...)
- Attribuées et réalisées

**Principes généraux :**

- Découper le projet en phases, étapes et tâches → plus on découpe, plus on avance
- Pour chacune :
  - Date de début au plus tôt
  - Date de fin au plus tôt
  - Date de début au plus tard
  - Date de fin au plus tard
  - → La différence entre date de début au plus tôt et de début au plus tard s'appelle la **marge** de la tâche.

**PERT** : Program Evaluation and Review Technique

- Permet de visualiser la **dépendance des tâches** et de procéder à leur **ordonnancement** → graphe de dépendance des tâches.
- Permet de mener un projet dans les contraintes de temps et d'identifier les **tâches critiques** (doit commencer au plus tard à la date ou elle peut commencer au plus tôt) qui conditionnent la **durée minimale du projet**



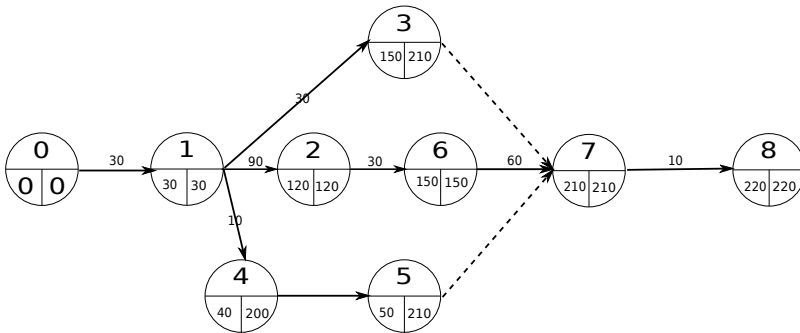
**Date au plus tôt :** Parcours du diagramme de gauche à droite et calcul du temps du plus long chemin menant du début du projet à cette tâche

**Date de début au plus tard :** Parcours du diagramme de droite à gauche, en commençant à la durée totale du projet puis on soustrait à chaque étape la durée de la tâche.

**Exemple :**

1. Préparer le menu (30min)
2. Acheter les ingrédients (90min)
3. Préparer l'apéritif (30min)
4. Nettoyer la table (10min)
5. Mettre la table (10min)
6. Préparer les ingrédients (30min)
7. Cuisiner les plats (60min)

## 8. Servir le repas (10min)



### Gantt :

1. Abscisse : temps
2. Ordonnée : différentes tâches ou lots
3. Durée d'exécution d'une tâche = barre horizontale
4. Liens de dépendance entre les tâches : flèche

### Construction :

1. Définir la liste des tâches avec leurs dépendances et leur durée, puis la liste des ressources affectées à chacune des tâches
2. Placer les tâches à effectuer dans le diagramme de Gantt dans l'ordre défini par leur priorité et en fonction des ressources encore disponibles
3. Placer des jalons (tâches importantes à temps presque nul et indispensables (ex : présentation du cahier des charges)).

## **Les règles les plus courantes d'affectation des ressources sont :**

1. Priorité à la réalisation des fabrications dont la date de livraison est la plus rapprochée
2. Priorité à la première commande arrivée
3. Priorité aux fabrications dont la durée totale est la plus courte (ou l'inverse selon la criticité)
4. Priorité aux fabrications qui utilisent le moins de ressources critiques
5. Priorité aux fabrications qui disposent du minimum de marge globale

## **Avancement :**

1. Barre de durée remplie proportionnellement à son degré d'accomplissement
2. Ligne verticale sur la date de jour : tâches accomplies à gauche, tâches non commencées à droite, tâches en cours traversent la ligne
3. Les jalons permettent de scinder le projet en phases clairement identifiées.

## PROJET N° 3      ANALYSE ET CAHIER DES CHARGES FONCTIONNEL

---

**Contenu du cahier des charges :** Dire **à quoi va servir** le produit et non ce qu'il sera. On identifie les fonctions qu'il doit remplir pour rendre service à l'utilisateur auquel il est destiné.

**Objectif :** Trouver le bon produit (qualité + réponse au besoin) (80 % des logiciels sont à la poubelle au bout d'un an, 60 % sont à la poubelle immédiatement)

### **Analyse fonctionnelle :**

- Traduction des besoins en fonctions à assurer et en tests d'acceptation.
- Critères d'appréciation des fonctions (faire valoriser devant le client les tâches chères mais nécessaires)
- Prioriser les fonctionnalités proposées en fonction de :
  - Positionnement dans l'architecture
  - Demande du client
  - Ressources disponibles
  - Valeur ajoutée
    - \* Pour l'utilisateur (ses attentes, sa motivation, ses capacités) → valeur d'usage et valeur d'estime.
    - \* Pour le concepteur (complexité, temps et ressources nécessaires)



**Contenu :** On propose une (ou des) réponse(s) aux besoins

- Architecture logicielle
- Fonctionnalités fournies
- Tests proposés
- Planning de réalisation et de livraison

**Contrat :** Accord sur le produit et le service attendu/réalisé ayant pour objet l'établissement d'obligations à la charge ou au bénéfice de chacune de ses parties.

**Propriétés d'un bon cahier des charges :**

- Cohérent
- Non Ambigu
- Précis
- Complet
- Réalisable, testable et maintenable
- Titre, Date, Version, Auteur(s), Liste de diffusion, Mots clés associés, Table des matières et des figures

**Plan classique**

1. Introduction : présentation du projet et docs de référence
2. Objectif : description de la future solution (fonctionnalités visibles et invisibles, besoins, technos)

3. Performance et configuration : prestations attendues
4. Organisation : intervenants et rôles, répartition des activités, suivi
5. Capacité et qualité de service (noms de variable, commentaires, tests)
6. Calendrier
7. Glossaire, lexique



**Besoins réels :**

1. Processus d'expression des besoins (reformulation)
2. Processus de développement du logiciel
3. Processus de tests (vérifier la conformité avec les besoins : tests unitaires, scénarios)

**Réalisation d'un logiciel de qualite :**

1. Pas seulement en fin de processus
2. Production échelonnée de documentation (normes, traces, mémoire)
3. Prendre en compte les couts de maintenance

**Assurance qualité :** Mise en oeuvre d'une approche préventive de la qualité, cad ensemble d'actions de prévention des défauts qui accompagnent le processus de développement des artefacts logiciels. (exemple : développement d'un plan de test)

**Contrôle qualité :** Mise en oeuvre d'une approche curative de la qualité. Bien que le processus de production soit satisfaisant, il présente des dysfonctionnements dont les effets doivent être éliminés. (exemple : exécution du plan de test)

**Activité d'assurance qualité :** Ensemble des actions préétablies et systématiques nécessaires pour donner la confiance appropriée

en ce qu'un produit ou un service satisfera aux exigences données relatives à la qualité. Cela passe par l'élaboration d'un Plan d'Assurance Qualité Logiciel.

**Manuel qualité (PAQL) :** Document décrivant les dispositions générales prises par l'entreprise pour obtenir la qualité de ses produits ou services; décrit l'ensemble des méthodes, règles et procédures mises en oeuvre pour développer du logiciel de qualité.

1. Usage interne et externe
2. Maîtrisé par tous (en interne)
3. Démonstration
4. Formation : techniques, méthodes et outils

### **Facteurs et critères de qualité :**

1. Facteurs = subjectif. Des choses qui vont favoriser la qualité ou non
2. Critères = objectif. Des choses qui vont "prouver" la qualité du projet. (Exemple : taux de couverture)

**Fiabilité des logiciels :** Probabilité pour qu'une panne du logiciel provoquant un écart par rapport au résultat attendu au delà des tolérances spécifiées, ait lieu dans un environnement opérationnel de référence et à l'intérieur d'une durée d'utilisation donnée.

**Fiabilité du matériel :** Doubler les composants (ex : le serveur).

**La fiabilité du logiciel dépend de :**

1. Du recueil des besoins
2. De la qualité de la conception
3. De la qualité de la réalisation
4. De la fiabilité des composants logiciels

**Le code :** Il faut ajouter du code pour faire les vérifications nécessaires (ne pas faire d'économie de code, mais éviter les redondances inutiles susceptibles de créer des problèmes de mise à jour).

1. L'efficacité est testée sur une petite partie du code
2. La fiabilité doit être constante sur la totalité du code
3. L'inefficacité peut être prédite, pas la non fiabilité

**Sécurité des logiciels :** Protection des ressources d'un système informatique contre les événements à caractère accidentel ou intentionnel : modification ou destruction des ressources, observation non autorisée des ressources, indisponibilité des ressources ou services etc.

**Répartition des causes de pertes de données :** Sur la base des déclarations des préjudiciés

1. erreurs (des programmeurs): 30%

2. incidents (coupures électriques ou autres): 34%
3. malveillance : 36%

### **Répartition des fraudeurs :**

1. Extérieurs : 10%
2. Utilisateurs : 35%
3. Informaticiens : 55%

### **Traitements induits :**

1. Contrôle d'accès
2. Contrôles de validité, cohérence de l'information saisie
3. Redondances
4. Audits d'intégrité, pistes d'audits
5. Alertes sur seuils
6. Cryptage

### **Plan type d'un PAQL (Plan Assurance Qualité Logiciel)** **:**

1. But, Domaine d'application et responsabilité : Portée du plan qualité et dispositions pour en assurer son application.
2. Documents applicables et documents de références (documents appelés dans le plan qualité)

3. Terminologie (glossaire, vocabulaire) : définir par rapport au client et en interne (définition des termes robustesse, fiabilité etc)
4. Organisation : Personnes intervenant dans le projet (rôle, place dans l'entreprise, responsabilité dans le projet, liens hiérarchiques et fonctionnels)
5. Démarche de développement : Liste des phases de développement en spécifiant :
  - Contenu des activités de la phase
  - Documents ou produits en entrée de la phase
  - Documents ou produits réalisés dans la phase
  - Conditions de passage à la phase suivante et points clés
6. Documentation
  - Liste des documents produits dans chaque phase
  - Références aux plans types de chaque document
  - Statut du document : livrable, consultable, privée
  - Documents classés en :
    - Documents de gestion de projet
    - Documents techniques de réalisation
    - Manuels d'utilisation et d'exploitation
7. Gestion des configurations : Eléments de configuration, moyens de développement et de tests, conventions d'identification



## 8. Gestion des modifications :

- Le responsable de leur mise en oeuvre
- Règles d'évolution et d'identification des éléments modifiés et de la nomenclature

## 9. Méthodes, outils et règles

## 10. Contrôle des fournisseurs (les sous traitants)

## 11. Reproduction, protection, livraison

## 12. Suivi de l'application du plan qualité : Dispositions prises

- Interventions du responsable qualité sur la démarche de développement
- Interventions du responsable qualité dans les procédures de gestion des configurations, de gestion des modifications, la vérification des exigences de qualité envers les fournisseurs.
- Modalités de recette et qualification

## 13. Annexes

- Fiche d'anomalie
- Rapport d'inspection
- Rapport d'audit
- Demande de modification
- Demande de dérogation

- Demande d'évolution du PAQ

**Cycle PDCA ou roue de DEMING :** Plan Do Check Act : idée = mettre en place un processus itératif permettant d'avancer dans le projet en veillant à la qualité. Réaliser progressivement des améliorations et répéter le cycle d'amélioration de nombreuses fois (cercle vertueux). On va s'améliorer au cours du projet, mais aussi d'un projet à l'autre.

**Prévoir Faire Contrôler Agir** en faisant attention à ne pas reculer (mettre en place une "cale"). On peut le traduire par Planifier, Développer(Documenter), Contrôler, Agir(Améliorer).

**Documentation** à chaque phase du cycle, pour montrer qu'on conduit le projet.

**Etape 1 : Préparer, Planifier** Ecrire ce que l'on va faire, se fixer des objectifs, des méthodes de travail, des étapes, montrer comment les objectifs vont être atteints. Il faut utiliser les éléments méthodologiques de l'entreprise (PAQ, standards etc ). Aboutit sur :

1. Cahier des charges
2. Un plan d'action/de projet/de travail

**Etape 2 : Développer et Documenter** Réaliser, produire et tester (écrire ce que l'on a fait).

**Etape 3 : Contrôler** Mettre en oeuvre les points de contrôle pré-établis, utiliser des indicateurs de performance, analyser des écarts sur le tableau de bord (liste quotidienne de tâches) etc. Faire des revues de projet (point plus poussé pour favoriser la visualisation collective), Analyse matricielle (croiser deux listes, exemple : besoins en lignes, produits en colonnes).

**Etape 4 : Agir** Pour maintenir la capacité de l'équipe à faire e travail, réagir pour lancer des actions correctives

