



## Gestion de projets

---

**Chouki Tibermacine**

**Chouki.Tibermacine@umontpellier.fr**



# Contenu de ce cours

## Ce que nous verrons

- Notions fondamentales de la gestion de projet
- Méthodes de planification, d'estimation et de suivi
- Méthodes de gestion de la qualité et des risques
- Méthodes agiles
- Bonnes pratiques et retours d'expérience

## Ce que nous ne verrons pas

- Outils de réalisation de projets (conception, codage, tests, ...)
- Aspects juridiques et économiques dans la gestion de projets

# Connaissances et compétences

## Connaissances

- Méthodes et outils pour la gestion de projet
- Principales techniques modernes et pratiques industrielles

## Compétences

- Savoir gérer un projet informatique de A à Z :
  - Savoir préparer l'avant-projet
  - Savoir assurer le suivi
  - Savoir livrer et clore un projet de qualité

# Organisation de l'ECUE

## Séances

- 12h CM (8 séances)
- 12h de TD/TP (8 séances)
- Emploi du temps sur Prose (toujours le vérifier)
- Transparents disponibles en ligne avant les cours :
  - <https://moodle.umontpellier.fr/course/view.php?id=10283>

# Évaluation de l'ECUE

## Modalités

- Examen final (Quiz) lors d'une séance (qui sera) programmée à la fin de la première partie du cours
- Notes de TD/TP : atelier SCRUM et TP sous ProjectLibre
- C'est une ECUE (compensation possible)
- Nombre d'ECTS : 2/7,5 (+FAS & ASAIoT)

# Plan de l'ECUE

1. Introduction générale à la gestion de projets
2. Techniques d'estimation des charges et de planification
3. Gestion des risques, aspects humains dans la gestion de projet et pilotage de projets (Arnaud Castelltort, Polytech)
4. Retours d'expérience (REX) industrielles (Damien Delbart, Sopra Steria)
5. Méthode SCRUM (Alexandre Preget, Sopra Steria)
6. Framework SAFe – *Scaled Agile Framework* (Matthieu Mouammar, Sopra Steria)

# Références bibliographiques



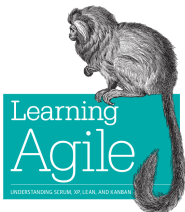
+ de 20 000 exemplaires vendus

Chantal Morley

8<sup>e</sup> édition

DUNOD

O'REILLY



Andrew Stellman & Jennifer Greene

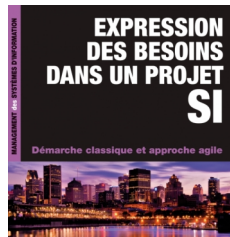


Olivier Englander  
Sophie Fernandes

## Manager un projet informatique

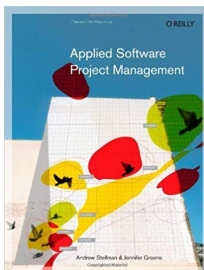
Comprendre pour faire les bons choix tout au long du projet

EVROLLES



Chantal Morley  
Jean Hugues  
Bernard Leblanc

DUNOD



## Applied Software Project Management

Andrew Stellman & Jennifer Greene  
Copyrighted Material



## Software Engineering

TENTH EDITION  
Ian Sommerville

# Plan du cours

1. Introduction générale
2. Grandes étapes d'un projet informatique
3. Découpage et modèles de cycle de vie d'un projet



# Plan du cours

1. Introduction générale
2. Grandes étapes d'un projet informatique
3. Découpage et modèles de cycle de vie d'un projet

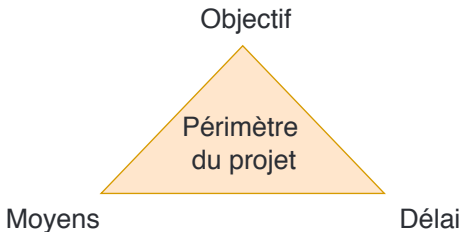
# Naissance des méthodes de gestion de projet

- Dans les années 50, émergence de réflexions sur la conduite de projets dans l'industrie : aéronautique, BTP, militaire, ...
- Objectifs :
  - augmenter la maîtrise des travaux
  - améliorer la coordination d'équipes pluridisciplinaires
- Formalisation mathématique des problèmes de gestion (modèles d'optimisation) pour prendre les décisions optimales
- Évolution vers des associations professionnelles (IPMA en Europe et PMI en Amérique du nord), des corpus (PMBok) et des certifications (PMP)

# Définition d'un projet

## C'est quoi un projet ?

- Ensemble d'activités à effectuer pour atteindre un *objectif* défini de façon spécifique, avec des *moyens* ad hoc et dans un *délai* donné<sup>1</sup>



1. Chantal Morley. Management d'un projet système d'information, 8ème édition. Dunod, 2016 10/58

# Définition d'un projet -suite-

## Travailler en mode projet

- Travailler sur un projet unique (avec ses caractéristiques propres) qui ne peut être traité par une procédure complètement standardisée
  - Différent d'une *activité répétitive* (traitée par une procédure standardisée). Ex : commande d'achat à chaque fois qu'il y a un besoin d'approvisionnement
- Travailler avec des contraintes de délai (temporalité)
  - Différent d'une *mission permanente*. Ex : un chargé de mission Qualité en entreprise surveille en permanence la qualité des produits/services (pas de délai)
- Implique un mouvement d'acteurs dans l'organisation (effet dynamisant)

# Définitions standardisées d'un projet

## ISO 10006

Processus **unique** qui consiste en un ensemble d'activités coordonnées et maîtrisées comportant des dates de début et de fin, entrepris dans le but d'atteindre un objectif conforme à des exigences spécifiques telles que les contraintes de délais, de coûts et de ressources

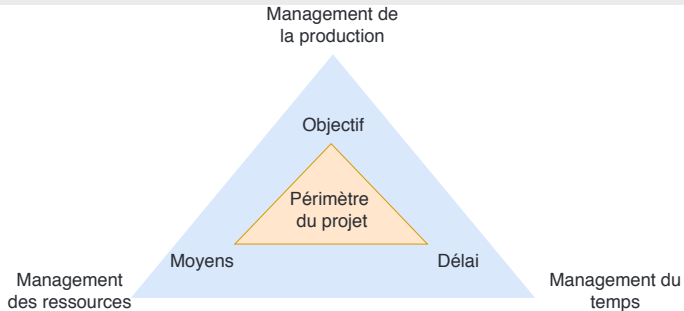
## PMBok du PMI

Effort temporaire exercé dans le but de créer un produit, un service ou un résultat **unique**

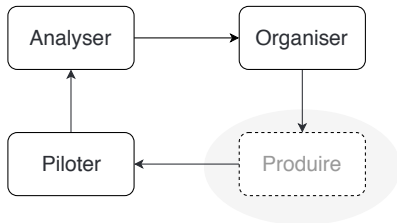
# Gestion ou Management d'un projet

## Définition

Conduite d'un projet tout au long de sa vie, en structurant et en supervisant son déroulement



## Gestion ou Management d'un projet -suite-



### Activités de gestion de projet

1. **Analyser** : étudier le contexte, évaluer les risques, estimer les charges, ...
2. **Organiser** : ordonner les tâches, constituer l'équipe et l'animer, répartir la charge de travail, ...
3. **Piloter** : suivi de l'avancement, gérer l'écart, prendre des décisions, gérer les conflits, gérer la qualité, ...

## Direction vs gestion d'un projet (Déf. ISO 10006)

- Deux niveaux liés à la prise de décision
- **Direction de projet** : prendre des décisions liées aux objectifs, à la politique, aux aspects financiers et à l'organisation du projet
- **Gestion de projet** : apporter à l'équipe de direction des éléments analysés qui aident à prendre des décisions pertinentes et opportunes
- Fonctions pouvant être assurées par la même personne



# Plan du cours

1. Introduction générale
2. Grandes étapes d'un projet informatique
3. Découpage et modèles de cycle de vie d'un projet

# Phases essentielles

1. Avant-Projet
2. Conception et réalisation
3. Validation
4. Recette et réception
5. Livraison
6. Bilan de fin de projet

## Phase 1 : Avant-projet

- On rédige, en tant que *prestataire de service*, et en accord avec le *commanditaire*, une **lettre de mission**
- Négociée et signée par les deux parties (pour éviter les malentendus à la fin du projet)
- Bien rédigée, elle permet de se prémunir des réorientations intempestives au cours de la mission

# Phase 1 : Avant-projet

## Contenu de la lettre de mission

- Objet de la mission
- Contour de la mission (existant, acteurs, contraintes et besoins précis)
- Modalités et délais de réalisation (moyens/fréquence de communication, livrables et dates butoirs estimées)
- Moyens offerts par le prestataire/commanditaire pour réaliser le projet
- Montant des honoraires

## Phase 1 : Avant-projet

- Ensemble d'études pour définir les caractéristiques principales de base du projet et produire le **cahier des charges**

### Étude d'opportunité

Permet de répondre à la question “Faut-il oui ou non aller de l'avant et démarrer le projet ?”

### Analyse de la valeur

Permet de mettre en balance les services rendus par le produit du projet avec ses coûts initiaux

### Étude de faisabilité

Permet d'analyser la faisabilité économique, organisationnelle et technique du projet

# Étude d'opportunité

Étudier 4 aspects :

1. le contexte :

- définition de l'origine du besoin (pourquoi a-t-on besoin de ce projet ?)
- le périmètre (quelles sont les personnes concernées ?)
- le volume (combien d'utilisateurs sont touchés ?)

2. l'existant : faire un état des lieux : organisation, fonctionnalités, contraintes, points forts et points faibles

## Étude d'opportunité -suite-

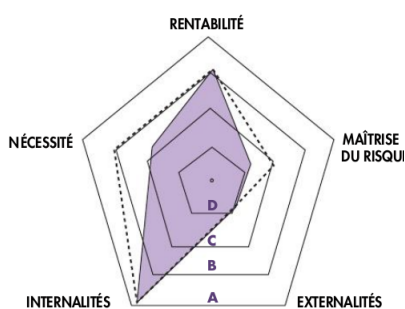
3. les fonctionnalités (de base et optionnelles ainsi que leurs contraintes), les grandes lignes sur les choix techniques, et les contraintes (humaines, financières, techniques, matérielles, etc.) qu'engendre la réalisation du projet
4. les enjeux : quels sont les gains attendus (qualitatifs, quantitatifs, organisationnels, financiers) ?

## Analyse de la valeur

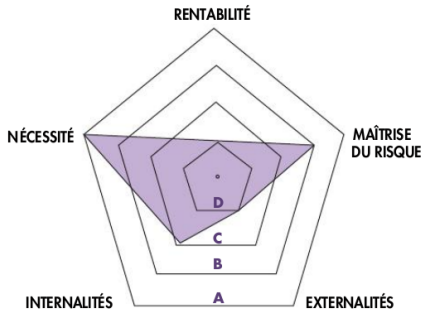
- Concept apparu dans les années 1940 aux États-Unis chez General Electric
- Appliquée aux projets informatiques à partir des années 90
- Objectif : recenser les besoins du commanditaire, mais aussi les analyser pour évaluer la valeur du projet
- Exemple de méthode pour les projets de SI : MAREVA 1 & 2 (Méthode d'Analyse et de REmontée de la VAleur) utilisée dans le secteur public en France
- Objectif : produire un graphique de type radar à cinq dimensions, qui restitue la valeur d'un projet (selon 5 critères)



# Méthode MAREVA



Projet interne



Projet d'infrastructure

- Tableurs à remplir avec des chiffres ou réponses à des questions fermées permettent d'obtenir ce type de graphiques

## Livrable de ces études

- Étude d'opportunité + analyse de valeur  
⇒ Livrable : Note de cadrage du projet
- Document qui officialise l'intention du projet
- Si ces études sont concluantes et validées (par les instances décisionnelles), l'étude de faisabilité peut démarrer

## Étude de faisabilité

- Elle vise à analyser la faisabilité économique, organisationnelle et technique du projet
- Pour permettre aux décideurs de situer le projet dans l'environnement de l'entreprise et d'en mesurer l'impact lors de la mise en oeuvre
- Elle est constituée des éléments suivants :
  - analyse de la demande (besoin utilisateur exprimé et périmètre)
  - objectifs recherchés (améliorer le système existant, ...)
  - bilan de l'existant (audit)
  - présentation des solutions (décrire brièvement différentes solutions et les comparer) et les conditions de leur mise en oeuvre (moyens humains et matériels)
  - conclusions et propositions
- Si accord, rédaction du **cahier des charges**

## Cahier des charges

- Document contractuel décrivant ce qui est attendu du maître d'oeuvre (MOE) par le maître d'ouvrage (MOA ou commanditaire)
- Élaboré par le MOA et destiné au MOE
- Il doit faire apparaître le besoin de manière fonctionnelle, indépendamment de toute solution technique, sauf à préciser l'environnement technique dans lequel la solution demandée doit s'insérer

## Organisation du cahier des charges

1. Présentation du projet (contexte, environnement, existant, ...)
2. Objectif : description de la future solution (description détaillée des besoins –*user stories*, diagrammes UML, ...–)
3. Prestations attendues (critères de choix de prestataires)
4. Organisation (intervenants et leur rôle)
5. Calendrier (planning souhaité par le MOA, avec des jalons pour éviter l'effet tunnel)

Faire attention au vocabulaire utilisé (dialogue MOA ↔ MOE ↗)

## Phase 2 : Conception et réalisation

- Conception : phase primordiale (réduit les risques durant le développement)
- Elle s'appuie sur l'analyse fonctionnelle (conceptuelle) : étude des données et des traitements, grâce aux techniques de modélisation (UML ou Merise, par ex.)
- Déterminer et ordonnancer les tâches de développement (découpage et planification –prochain cours)
- Estimer leurs charges et déterminer les profils nécessaires à leur réalisation
- Réalisation : développement des programmes et des sources de données, écriture des tests, ...

## Phase 3 : Validation

Trois techniques de validation :

- La preuve formelle (utilisée surtout dans certains domaines métiers de systèmes critiques, comme l'aérospatial, médical, ...)
- La révision de code par les pairs (*code review*)
- Les tests :
  - techniques : unitaires, d'intégration et de performance (montée en charge)
  - fonctionnels (métier) : simuler des utilisateurs qui testent les fonctionnalités (interfaces, ergonomie, sécurité, ...)

Livrable : plan de test (document qui décrit les objectifs et les moyens pour réaliser les tests ainsi que leur organisation technique et les bilans)

# Malentendus dans la réalisation d'un projet



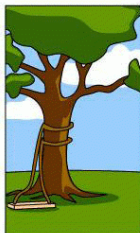
Comment le client l'a souhaité



Comment le chef de projet l'a compris



Comment l'analyste l'a schématisé



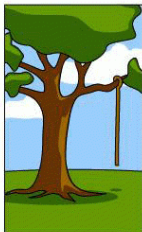
Comment le programmeur l'a écrit



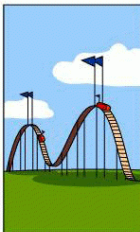
Comment le Business Consultant l'a décrit



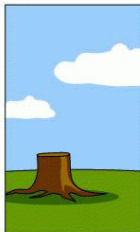
Comment le projet a été documenté



Ce qui a été installé chez le client



Comment le client a été facturé



Comment le support technique est effectué



Ce dont le client avait réellement besoin



## Phase 4 : Recette et réception

- Recette ou “essai de réception” : vérifier la conformité de l’ouvrage vis à vis du cahier des charges
- Préparation de la recette et son exécution : tests automatiques et manuels
- Livrable : document de recette
  - présentation générale : objectif de la réception (recette logicielle, màj, ...), présentation des livrables, limites (tests non couverts), ...
  - responsabilités (une équipe de réception (différente de l’équipe de réalisation) est identifiée au sein de l’équipe projet, pour la préparation et l’exécution de la réception)
  - procédure de réception (modalités de mise à disposition des livrables)

## Phase 4 : Recette et réception -suite-

- Document de recette -suite-
  - ajournement de la réception (si l'équipe de réception estime que la qualité de la livraison est insuffisante)
  - organisation de la recette (informations nécessaires à la préparation et l'exécution des tests de recette : un sous-ensemble représentatif des tests sur les points les plus critiques)
  - déroulement de la recette (détaille les dispositions suivies pendant l'exécution des tests, en termes de recueil des anomalies)
- Procès-verbal de réception indique les conclusions de la réception, et permettra de déclencher la facturation
- Procès verbal définitif si pas de réserves

## Phase 5 : Livraison

- Mise à disposition (fournir l'application aux utilisateurs finaux) : sur sites pilotes d'abord ensuite son déploiement en masse (en production)
- "Conduite du changement" : accompagnement aux utilisateurs finaux pour acquérir de nouvelles habitudes et apprendre à utiliser le nouveau produit (formation, hotline, ...)
- Formation : sessions planifiées (pour une dizaine de personnes max), avec enquête de satisfaction
- Suivi : maintenance (tâche contractuelle) corrective, évolutive, adaptative, ... (maintenance externalisée : TMA – tierce maintenance applicative)

## Phase 6 : Bilan de fin de projet

- Vise à présenter les succès et/ou les améliorations à apporter
- Permet de comparer les dispositions initialement prévues avec le déroulement réel du projet
- Vise à évaluer le projet (atteintes des objectifs, raisons des écarts, bilan des difficultés, etc.) et capitaliser le savoir-faire (tirer tous les enseignements techniques, de management ou d'organisation, utiles pour les projets futurs)
- Livrable : bilan de fin de projet
- Enquête de satisfaction (une dizaine de questions simples) : faire des stats et communiquer les résultats à l'équipe de projet

# Bilan de fin de projet

Organisation du livrable :

1. Identification du projet
2. Bilan de la charge et du coût des achats (causes et conséquences des écarts/dépassements)
3. Bilan de l'organisation (fréquence et efficacité des réunions, organisation de la com., ...)
4. Bilan technique (points forts et faibles des solutions techniques)
5. Anomalies et évolutions
6. Bilan qualité (critique du système de qualité et coûts d'obtention de la qualité)
7. Appréciation globale

# Plan du cours

1. Introduction générale
2. Grandes étapes d'un projet informatique
3. Découpage et modèles de cycle de vie d'un projet

# Découpage d'un projet

## Principes de découpage

- Sommets du triangle "Management de projet" :  
production, ressources et temps
- Découper le projet pour répartir sur le temps la production et les ressources
- Découpage en sous-ensembles quasi-autonomes, pour réduire la complexité tout en prenant en compte les dépendances entre les sous-ensembles

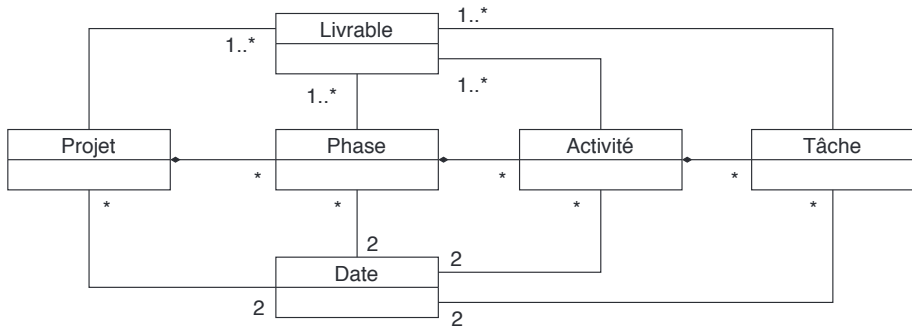
## Découpage d'un projet -suite-

### Caractéristiques des sous-ensembles

- Chaque sous-ensemble donne lieu à un résultat bien identifié
- La charge propre à chacun peut être évaluée
- Les contraintes d'enchaînement entre les sous-ensembles sont repérables (parallélisme et séquençement)
- Découpage récursif (sous-sous-ensembles, ...)

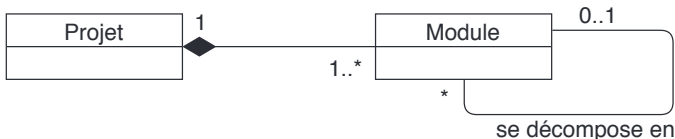


## Découpage temporel d'un projet



- Livrable = résultat à atteindre (peut faire l'objet d'un engagement contractuel)
- Ensemble ordonné des phases = **cycle de vie du projet**

## Découpage structurel d'un projet



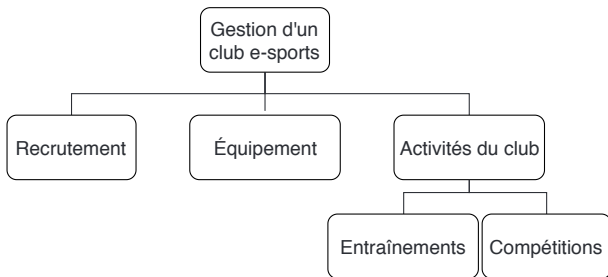
- Maîtriser un projet de taille importante
- Répartition dans des sous-projets séparés (parfois sous-traités)
- Développement en parallèle (accélérer la réalisation)
- Développement incrémental (livraison de versions successives avec de + en + de modules)
- Mais ça nécessite une visibilité suffisante sur le résultat à produire

## Découpages normalisés

Trois découpages dans les normes internationales : PBS, WBS et OBS

### PBS : Product Breakdown Structure

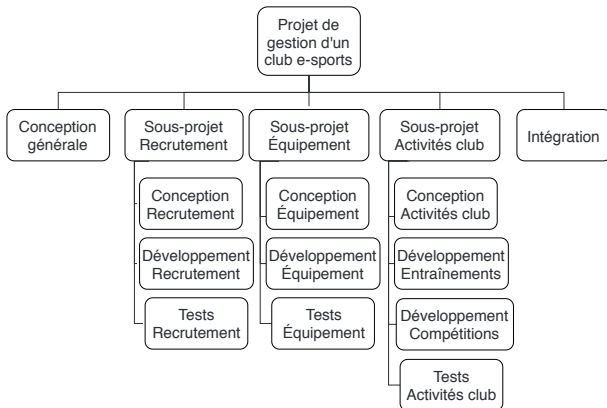
- Structure de décomposition du produit (découpage structurel)
- Arbre regroupant les différents modules/composants du produit final



# Découpages normalisés -suite-

## WBS : Work Breakdown Structure

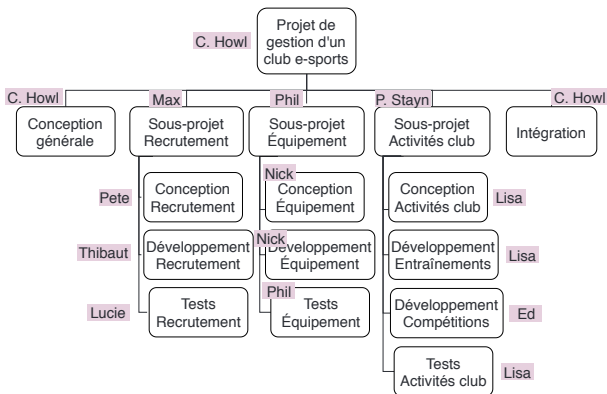
- Struct. de décomp. du travail (organigramme/réseau de tâches)
- Basé sur le découpage structurel et temporel



# Découpages normalisés -suite-

## OBS : Organizational Breakdown Structure

- Structure de décomposition de l'organisation
- WBS + noms des responsables (organigramme fonctionnel)



## Découpage d'un projet SI (norme AFNOR Z67-101)

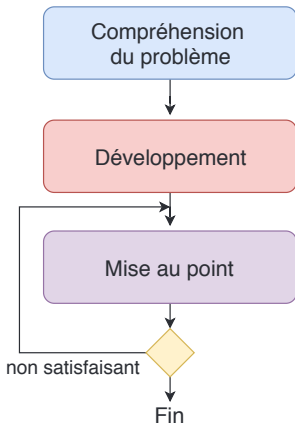
1. Étude préalable (Exploration, conception d'ensemble et appréciation de la solution retenue)
2. Conception détaillée (Conception du système de traitement de l'information, Spécifications fonctionnelles du système informatique, et étude organique générale)
3. Réalisation (Étude organique détaillée, programmation et tests, et validation technique)
4. Mise en oeuvre (réception provisoire et exploitation sous contrôle)
5. Évaluation (évaluation du système informatique, et évaluation du système de traitement de l'information)

Autres normes et réfs : ISO 12207 (processus de cycle de vie logiciel) et ISO 21500 (gestion de projets en général)  $\simeq$  PMBoK et PRINCE2

## Modèles de cycle de vie

- Le modèle du *code-and-fix*
- Le modèle de la transformation automatique
- Le modèle d'intégration
- Le modèle de la cascade (*waterfall*)
- Le modèle en V
- Le modèle itératif-incrémental
- Le modèle en spirale
- Le modèle du processus unifié (UP  $\Rightarrow$  RUP)
- Le modèle par prototypage rapide (RAD)
- Le modèle des méthodes agiles : Scrum, XP, Lean et Kanban

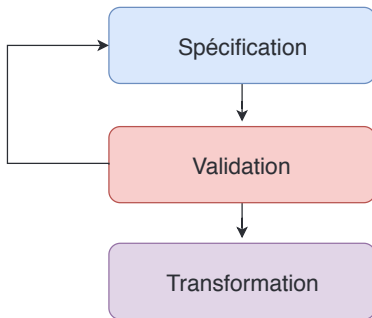
## Modèle du *code-and-fix*



Le modèle le plus intuitif (compatible avec de petits projets avec peu de risques de problèmes de conception tard dans le cycle de vie)

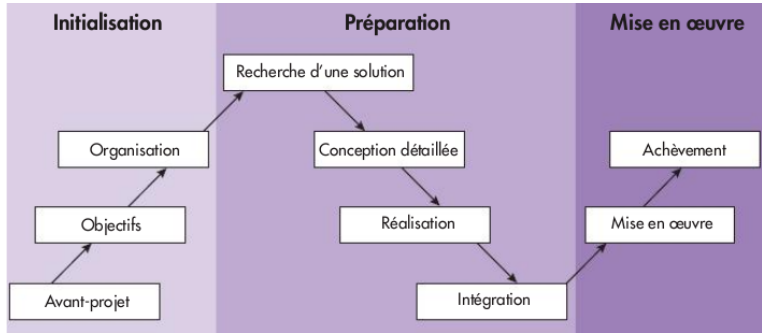


# Modèle de la transformation automatique



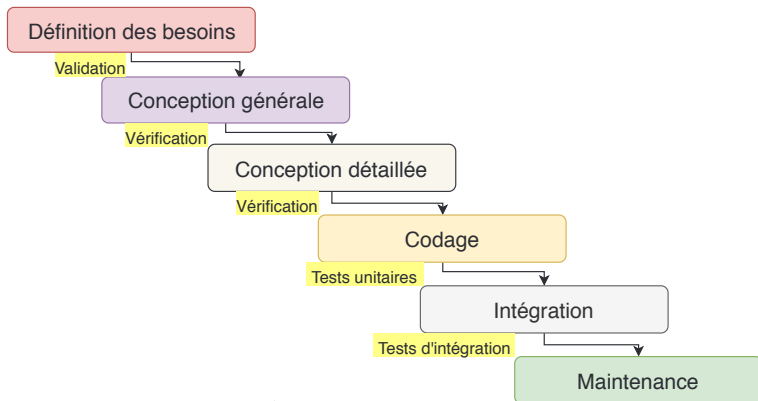
Transformer automatiquement des spécifications (décrites de façon exhaustive) en programmes

# Modèle d'intégration



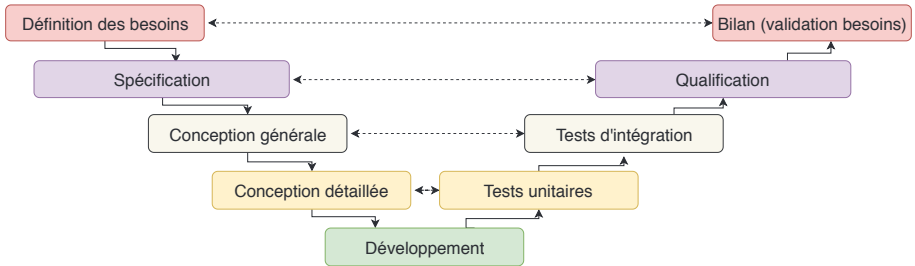
- Introduire une boîte noire “clés en main” dans un système d'information existant
- Conception et Réalisation = Préparation des interfaces d'intégration au SI existant

# Modèle de la cascade (*Waterfall*)



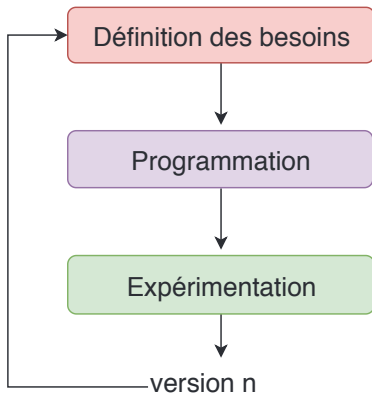
- Modèle séquentiel (validation de chaque phase avant de passer à la suivante)
- Adapté aux gros projets complexes, impliquant des coûts importants lors des phases en aval

# Modèle en V



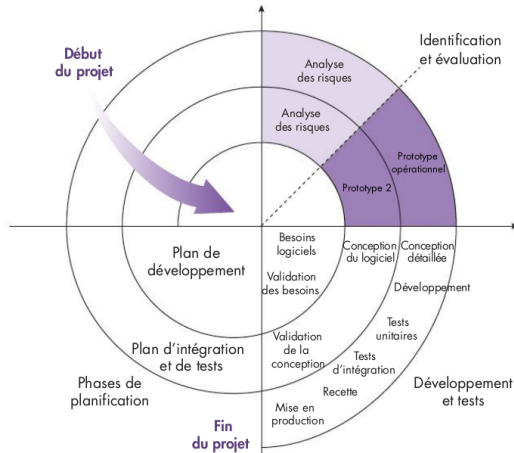
- Éviter l'effet tunnel en explicitant des critères d'acceptation, qui permettent (dans la 2ème branche du V) une validation, avec un possible retour en arrière vers chaque phase dans la première branche du V

## Modèle itératif-incrémental



- Idée de base : les besoins ne peuvent être complètement définis qu'après expérimentation sur des versions successives (même incomplètes), qui sont fournies au commanditaire

# Modèle en spirale

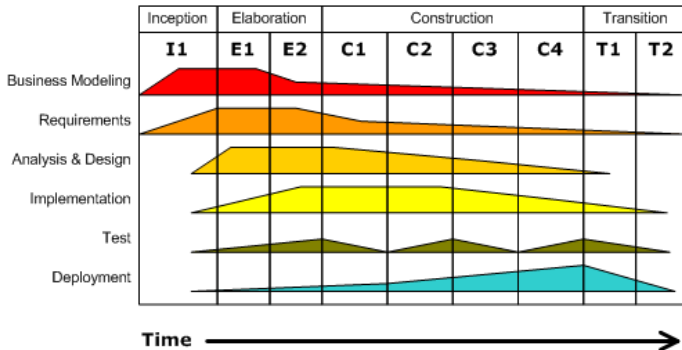


- Modèle basé sur l'analyse des risques pour préparer les objectifs du cycle suivant de la spirale (modèle incrémental)
- Modèle qui privilégie des prises de décisions alternatives

# Modèle du processus unifié (RUP)

## Iterative Development

Business value is delivered incrementally in time-boxed cross-discipline iterations.



- 4 phases et 6 types de tâches répartis à des degrés différents sur les phases
- Certaines phases sont incrémentales voire itératives ( $E_i$ ,  $C_i$  ...)

## Modèle RAD (*Rapid App Dev*)

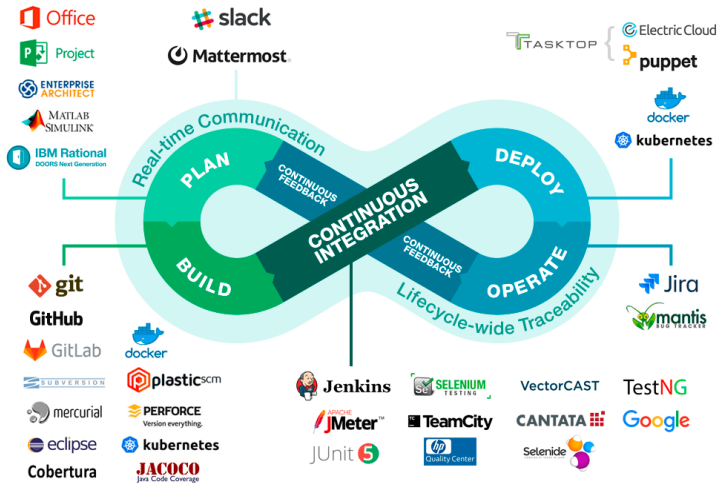
- Rupture avec le Waterfall (itératif, incrémental et adaptatif)
- Livrer rapidement un minimum de fonctions viables (un prototype) pour assurer un retour sur investissement rapide
- Démarche participative de tous les intervenants du projet (ateliers permettant d'aboutir rapidement à un consensus sur les besoins à couvrir)
- Priorité aux délais (une limite de temps est fixée pour chaque résultat souhaité)
- Ancêtre du modèle des méthodes agiles



## Modèle des méthodes agiles

- Modèle adapté aux projets très évolutifs
- Remise en question périodique pour rester au plus près des besoins du commanditaire
- Modèle itératif et incrémental
- Bonne communication au sein de l'équipe du projet, et implication importante du commanditaire
- Tests intensifs lors des différentes itérations de développement
- Plus de détails dans un prochain cours

# Modèle DevOps



<https://intland.com/devops-it-operations/>

