Génie logiciel
et
Gestion de projet

Introduction

- Le génie logiciel (software engineering) existe depuis plus de 30 ans
- Né des constatations que les logiciels :
  - Ne sont pas fiables
  - Sont incroyablement difficiles à réaliser dans les délais
  - Ne satisfaisaient pas le cahier des charges

Exemples

- Perte de la 1ère sonde Mariner 1 (1962) vers Vénus suite à une erreur de programmation dans un programme Fortran
- Perte, en 1971, de 72 ballons d'expérimentation météorologique à cause d'un bug logiciel
- Abandon d'un projet d'informatisation de la City après 4 ans de travail et 100 M£ de perte
- Echec d'ARIANE 501 (1996) suite à un bug logiciel
- Invalidation de version de Windows XP suite au changement de version du Windows Genuine Advantage (2011)

Génie Logiciel

- Si l'on veut maîtriser le développement de systèmes complexes, il faut :
  - Rédiger de façon claire les spécifications du système (ce que l'on attend)
    - Comment être sûrs que ces spécifications sont complètes ?
    - Comment être sûrs que ces spécifications sont cohérentes ?
  - Valider/vérifier toutes les étapes du développement
    - A-t-on des moyens de validation/vérification (mathématiques) ?

Génie Logiciel

- Si l'on veut maîtriser le développement de systèmes complexes, il faut :
  - Réutiliser des sous-systèmes déjà réalisés (mais pas n'importe comment)
    - A-t-on des règles, des outils pour aider à la réutilisation ?

Nécessité d’une base théorique et d’une approche ingénierie (science de l’ingénieur) du logiciel

Génie Logiciel

- Le génie logiciel comporte des aspects de gestion de projet et des notions de qualité (satisfaire le client)
- Ceci en utilisant des méthodes, des modèles, et des outils.

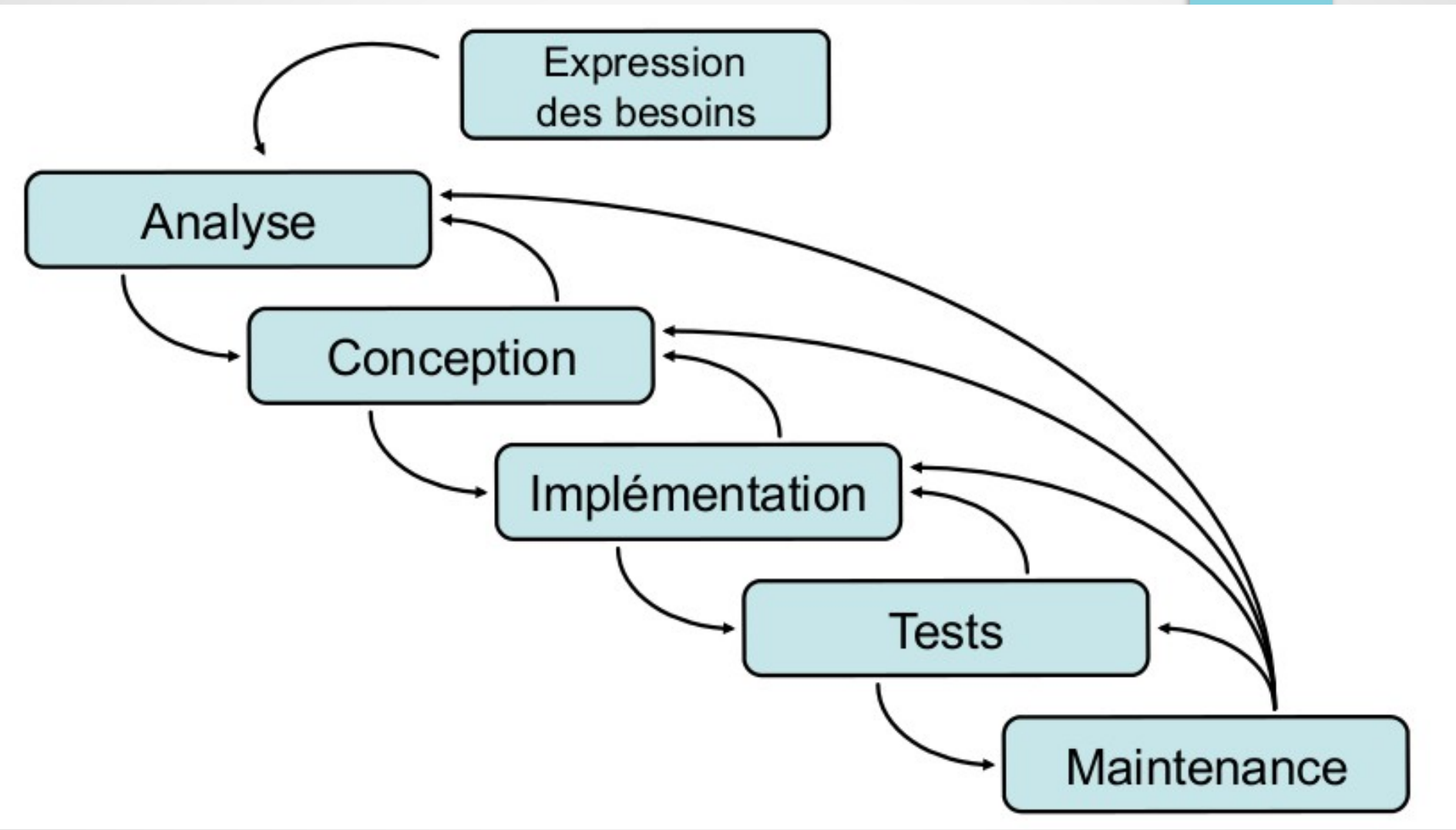
Modèles de développement

- Modèle en cascade
- Modèle en V

Modèle en cascade

- Atteinte de l'objectif par atteinte ordonnée de sous-objectifs. Les activités sont représentées dans des processus séparés.
- Processus séquentiel: Chaque étape doit être terminée avant que la suivante commence.
- Livrables:
  - A la fin de chaque étape, le livrable est vérifié et validé.
  - Vérification: le livrable est-il correct ?
  - Validation: est-ce le bon produit ? (Comparé à l'énoncé de l'étape).

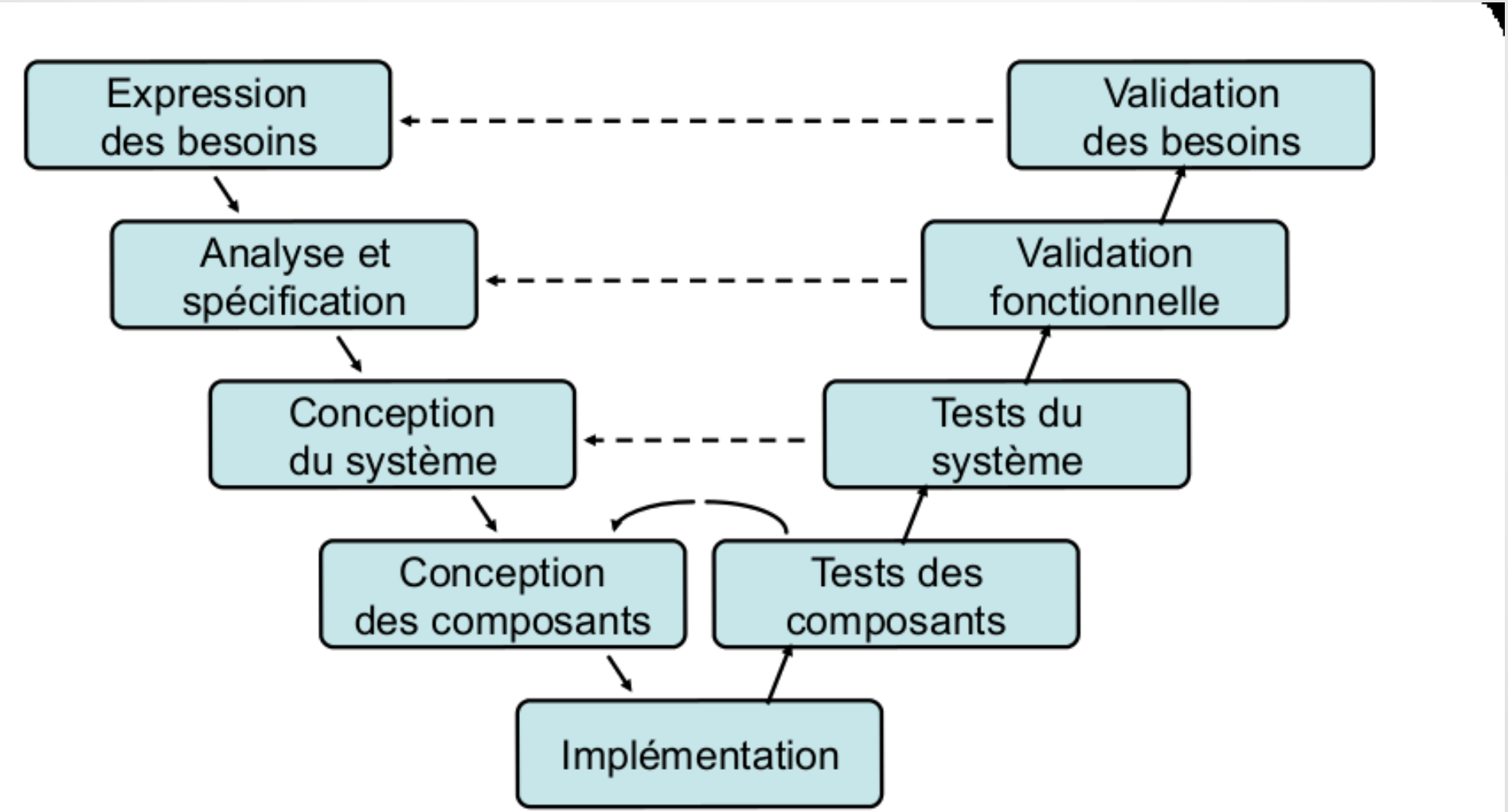
Modèle en cascade



Modèle en V

- Amélioration du modèle en cascade
- Met en évidence la symétrie et la relation qu'il y a entre les phases du début du cycle de vie et celles de fin.
- Les phases du début doivent être accompagnées d'une planification des phases de fin.
- Lors de la planification, on développe et documente les plans de test.

Modèle en V


Expression des besoins
Validation des besoins
Analyse et spécification
Validation fonctionnelle
Conception du système
Tests du système
Conception des composants
Tests des composants
Implémentation

Activités de développement

- Elles sont décrites de façon indépendante en indiquant leur rôle.
- Selon le modèle, une activité peut jouer un rôle plus ou moins important et parfois ne pas exister du tout.

Activités de développement

- Elles concernent :
  - Planification (Étude de la faisabilité)
  - Spécification des besoins
  - Analyse (Spécification formelle)
  - Conception (Spécification technique)
  - Implémentation (Codage) et tests unitaires
  - Intégration et tests d’ensemble
  - Livraison
  - Maintenance

Planification

- Objectifs :
  - identification de plusieurs solutions et évaluation des coûts et bénéfices de chacune d'elles
- Activités :
  - simulation de différents scénarios de développement
- Résultats :
  - Rapport d'analyse préliminaire et un schéma directeur contenant :
    - la définition du problème et les différentes solutions étudiées, avec, pour chacune d'elles, les bénéfices attendus, les ressources requises (délais, livraison, etc.)

Spécification des besoins

- Objectifs :
  - définition de ce que doit faire le logiciel
- Activités :
  - Description du problème à traiter afin d'identifier les besoins de l'utilisateur, de spécifier ce que doit faire le logiciel : informations manipulées, services rendus, interfaces, contraintes
  - Mise en oeuvre des principes : abstraction, séparation des problèmes, séparation des besoins fonctionnels

Spécification des besoins

- Résultats : cahier des charges et plan qualité
  - un dossier d'analyse comprenant les spécifications fonctionnelles et non fonctionnelles du produit
  - une ébauche du manuel utilisateur pour les non-informaticiens
  - une première version du glossaire contenant les termes propres au projet.
  - un plan de test du futur système (cahier de validation)

Analyse

- Objectifs :
  - Analyse détaillée de toutes les fonctions et autres caractéristiques que le logiciel devra réaliser pour l'usager, tel que vu par l'usager.
- Activités :
  - Répondre au « Que fait le système ? »
  - Analyse de l'existant et des contraintes de réalisation
  - Abstraction et séparation des problèmes, séparation en unités cohérentes

Analyse

- Résultats : Dossier d'analyse et plan de validation
  - Modèle du domaine
  - Définition du modèle conceptuel.
  - Plan de validation, dossier de tests d'intégration

Conception

- Objectifs :
  - Définition de l'architecture générale du logiciel. Spécification de la manière dont chacun des composants du logiciel sera réalisé et comment ils interagiront.
- Activités :
  - Répondre au « Comment réaliser le système »
  - Décomposition modulaire, définition de chaque constituant du logiciel : informations traitées, traitements effectués, résultats fournis, contraintes à respecter

Conception

- Résultats : dossier de conception + plan de test global et par module
  - proposition de solution au problème spécifié dans l'analyse
  - organisation de l'application en modules et interface des modules (architecture du logiciel)
  - description détaillée des modules avec les algorithmes essentiels (modèle logique)
  - structuration des données.

Implémentation

- Objectifs :
  - Réalisation des programmes dans un (des) langage(s) de programmation
  - Tests selon les plans définis lors de la conception
- Activités :
  - traduction dans un langage de programmation,
  - Mise au point (débogage)
- Résultats : dossiers de programmation et codes sources.
  - Collection de modules implémentés, non testés
  - Documentation de programmation qui explique le code

Tests unitaires

- Objectifs :
  - test séparé de chacun des composants du logiciel en vue de leur intégration
- Activités :
  - réalisation des tests prévus pour chaque module
  - les tests sont à faire par un membre de l'équipe n'ayant pas participé à la fabrication du module
- Résultats :
  - résultats des tests avec les jeux d'essais par module selon le plan de test.

Intégration et test du système

- Objectifs :
  - Intégration des modules et test de tout le système
- Activités :
  - Assemblage de composants testés séparément
  - Démarche d'intégration (ascendante, descendante ou les deux)
  - Conception des données de tests
  - Tests Alpha : l'application est mise dans des conditions réelles d'utilisation, au sein de l'équipe de développement (simulation de l'utilisateur final)
  - Documentation des éléments logiciels

Intégration et test du système

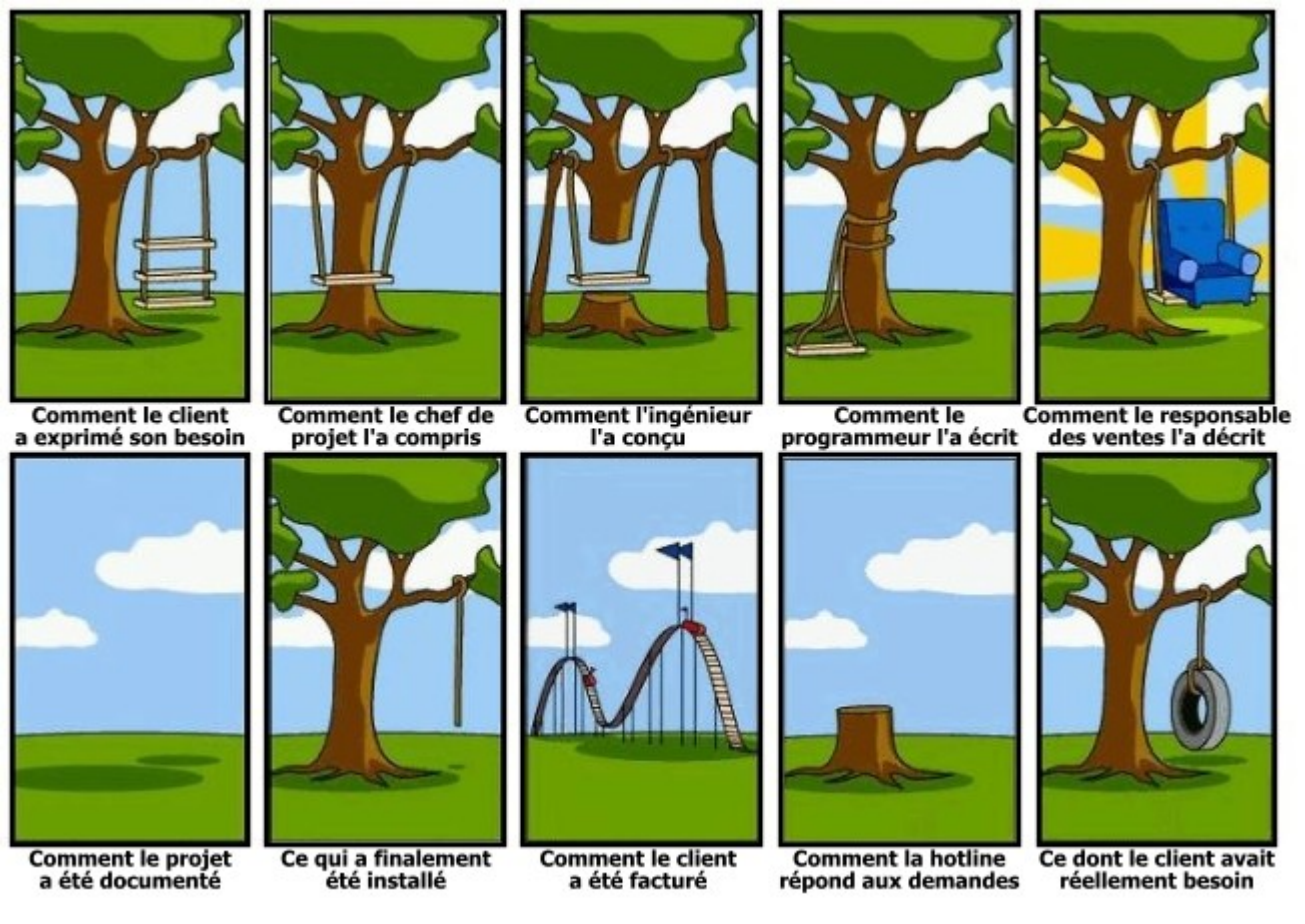
- Résultats :
  - Rapports de test
  - Manuel d'utilisation

Livraison, maintenance, évolution

- Objectifs :
  - Livraison du produit final à l'utilisateur,
  - Suivi, modifications, améliorations après livraison.
- Activités :
  - Tests Bêta : distribution du produit sur un groupe de clients avant la version officielle,
  - Livraison à tous les clients,
  - Maintenance : corrective, adaptative, perfective.

Livraison, maintenance, évolution

- Résultats : la version finale du manuel utilisateur, les traces d'évolution du système, les rapports d'exploitation
  - Produit et sa documentation
  - Trace d'exploitation et d'évolution


Comment le client a exprimé son besoin
Comment le chef de projet l'a compris
Comment l'ingénieur l'a conçu
Comment le programmeur l'a écrit
Comment le responsable des ventes l'a décrit
Comment le projet a été documenté
Ce qui a finalement été installé
Comment le client a été facturé
Comment la hotline répond aux demandes
Ce dont le client avait réellement besoin