

# Chapitre 1 - Gestion d'un projet informatique et cycle de vie.

## BUT 2 - R3.03 Analyse

Halim Djerroud, d'après le cours de  
**Thomas Dufaud** <thomas.dufaud@uvsq.fr>



révision : 0.1

# Plan

## Gestion d'un projet informatique et cycle de vie

- 1 Gestion de projet informatique
  - Développer des systèmes complexes
  - Le Génie logiciel
  - Les activités du Génie Logiciel
- 2 Cycle de vie du logiciel
- 3 Vers d'autres méthodes de gestion
  - Constat
  - Les méthodes agiles
  - Cycles de vie multiniveaux, l'exemple du processus unifié

# Objectif du cours

Objectif du cours :

- **Gestion de projet** pour réaliser des **solutions informatique** avec une **fortes contraintes de qualités** dans le cadre des **systèmes d'information des organisations**.

C'est quoi un systèmes d'information :

## Définition 1.1 : Système d'information (SI) source (Reix et al. 2016)

Un système d'information est un ensemble organisé de ressources (matériels, logiciels, personnel, données et procédures) qui permet de :

- acquérir l'information (sur différents supports)
- traiter l'information (effectuer des opérations)
- stocker l'information (conserver l'information)
- communiquer l'information (transmettre : éditer, imprimer, etc.)

# Composition d'un SI

## Nature hétérogène :

- Le système informatique est composé de ressources matérielles, logicielles, humaines et organisationnelles.
- Les ressources matérielles sont les composants physiques du système, tels que les ordinateurs, les serveurs, les logiciels, les réseaux, etc.
- Les ressources logicielles sont les programmes informatiques qui permettent de faire fonctionner le système.
- Les ressources humaines sont les personnes qui utilisent le système et qui en assurent le développement, la maintenance et le support.
- Les ressources organisationnelles sont les règles, les procédures et les structures de l'organisation qui encadrent le développement et l'utilisation du système.

## Tâches accomplies par un SI :

- La collecte, le stockage et la diffusion de l'information
- Le traitement de l'information
- La prise de décision
- La communication
- La collaboration

### Attention :

Un **système informatique** fait partie d'un système d'information mais **ne constitue pas** nécessairement un **système d'information**.

# A quoi sert l'outil informatique

- Les outils de l'informatique permettent de réaliser les spécifications d'un SI.
- Plusieurs systèmes informatiques peuvent interagir pour réaliser ces spécifications, ou remplacer d'autres outils.

Nous nous intéressons donc ici aux **méthodes** nécessaires au **développement des applications** utiles aux systèmes d'information, et s'intégrant de fait dans un environnement complexe avec différents acteurs.

# Méthodologie pour la gestion de projet informatique

Afin de proposer une méthodologie pour la gestion de projet informatique dans le contexte des systèmes d'information il nous faut définir :

- Le cadre théorique spécifique à l'informatique et la manière dont on peut **découper les activités** d'un développeur
- Cet enchaînement définit ce que l'on appelle le **cycle de vie logiciel**
- Il n'y a pas de cycle unique.
- Il est possible d'adapter son cycle de vie logiciel et définir un processus métier particulier pour le développeur.

# Le Génie logiciel

- Le génie logiciel nous permet de définir une méthodologie pour la réalisation d'un logiciel.

Le Génie Logiciel est la discipline liée à tous les aspects de la production du logiciel complexe et avec d'importantes contraintes de qualité. Elle est liée à l'application de théories, de méthodes et à l'utilisation d'outils pour le développement logiciel d'une façon professionnelle. Elle favorise et permet le travail en équipe.

# Les objectifs du génie logiciel

- Adopter une approche systématique et organisée
- Utiliser les techniques et outils appropriés selon la nature du problème, les contraintes de développement et les ressources

# Principes du génie logiciel (GL)

- ➊ **Rigueur et formalisme** : permettent de communiquer et discuter en équipe et avec le client
- ➋ **Abstraction** : permet d'identifier et définir les concepts du projet ainsi que les éléments qui le compose
- ➌ **Modularité** (décomposition en sous-système) : proposer une décomposition en sous-système
- ➍ **Anticipation** : implique de prendre en compte les multiples contraintes d'un projets tant techniques qu'organisationnelles
- ➎ **Généralisation** : permet d'envisager une utilisation plus générale des composants d'un système ou des procédures de réalisation.
- ➏ **Croissance incrémentale** : un principe à viser si l'on veut pouvoir faire évoluer le système par étape

# Les activités du Génie Logiciel

Pour mener à bien un projet informatique, le développeur ou l'équipe de développement doit réaliser un ensemble d'activité allant de l'analyse des besoins du projet à la maintenance du logiciel produit en passant par son implémentation et les tests.

- 1 Analyse des besoins
- 2 Spécification
- 3 Conception
- 4 Programmation
- 5 Validation et vérification (Tests)
- 6 Intégration et gestion des configurations

## (a) Analyse des besoins (ADB)

- **Rôle :**

- Identifier le problème
- Documenter les exigences
- Impliquer les utilisateurs et les experts dans le domaine de l'application

- **Entrées :**

- Cahier des charges (rédigé par le client)
- Données fournies par les experts du domaine et par les utilisateurs potentiels

- **Résultats :**

- Document d'ADB destiné aux utilisateurs et utile pour les développeurs.
- Recueil des besoins
- Cahier des charges (rédigé de manière technique avec le client)

## Remarque

Attention, de nombreux produits ne correspondent pas aux besoins du client. On peut identifier plusieurs causes, comme par exemple

- Le client ne sait pas ce qu'il veut
- Il existe des problèmes de communication entre le client et l'équipe de développement.
- L'équipe de développement ne comprend pas la politique d'organisation du client
- On observe un changement des exigences au cours du projet
- Les délais ne sont pas raisonnables
- etc.

## (b) Spécification

- **Rôle :**

- Décrire de façon précise le système à construire, le **QUOI** et non le comment.

- **Entrées :**

- Document d'ADB destiné aux utilisateurs et utile pour les développeurs.
- Recueil des besoins
- Cahier des charges (rédigé de manière technique avec le client)

- **Résultats :**

- Documentation textuelle avec description de scénarii (cas d'utilisation) possiblement appuyée par des diagrammes (ex : std UML)

## (c) Conception

- **Rôle :**

- Enrichir la spécification du logiciel en l'orientant vers la réalisation, le **COMMENT** et non le quoi.

- **Entrées :**

- Spécifications
- Détail de la plateforme de développements (dans le Recueil des besoins)

- **Résultats :**

- **Modèle** défini dans une documentation textuelle appuyée par des diagrammes (ex : std UML)

## (d) Programmation

- **Rôle :**
  - Spécifier le système en utilisant un langage de programmation.
- **Entrées :**
  - Documentation de la conception détaillée
  - Spécification (si pas de conception détaillée)
- **Résultats :**
  - Code qui implémente les fonctionnalités
  - Documentation technique du code
  - Dépôt, historique de gestion de version

## (e) Validation (e1) et Vérification (e2)

- **Rôle :**

- (e1) Répond à la question "Le logiciel satisfait-il les attentes de l'utilisateur ?"
- (e2) Répond à la question "Le logiciel satisfait-il les spécifications ?"

- **Entrées :**

- Documentation de la conception détaillée pour les tests unitaires
- Documentation de la conception architecturale pour les tests d'intégration
- Documentation d'ADB pour les tests d'acceptation

- **Résultats :**

- Cas de tests
- Analyse des résultats d'exécution des cas de tests

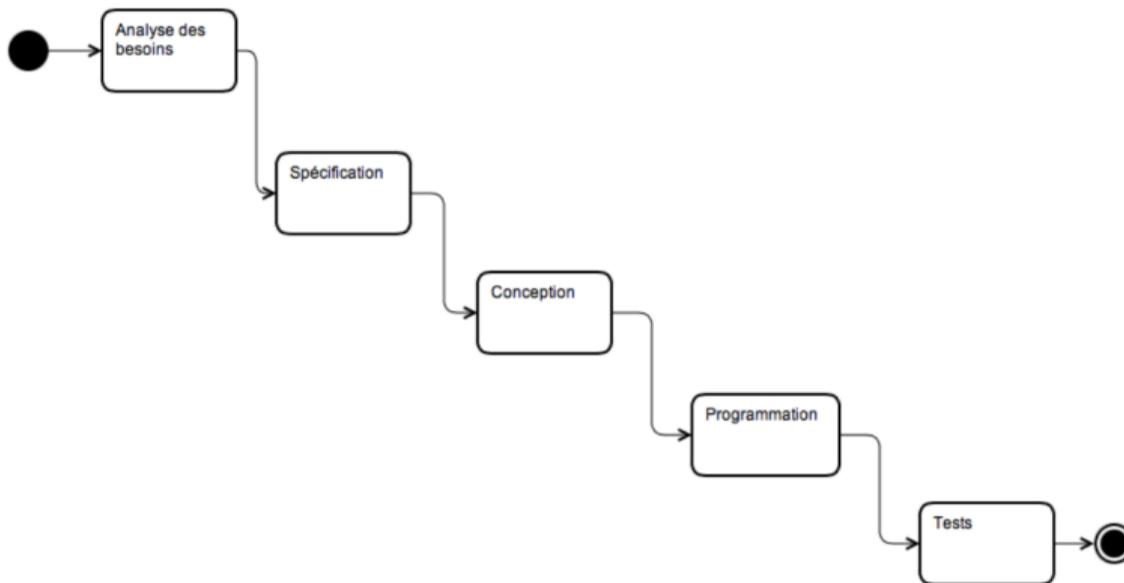
## (f) Intégration et gestion des configurations

- **Rôle :**
  - Décrire le système et ses évolutions.
- **Entrées :**
  - Version du logiciel
- **Résultats :**
  - Dossier de maintenance

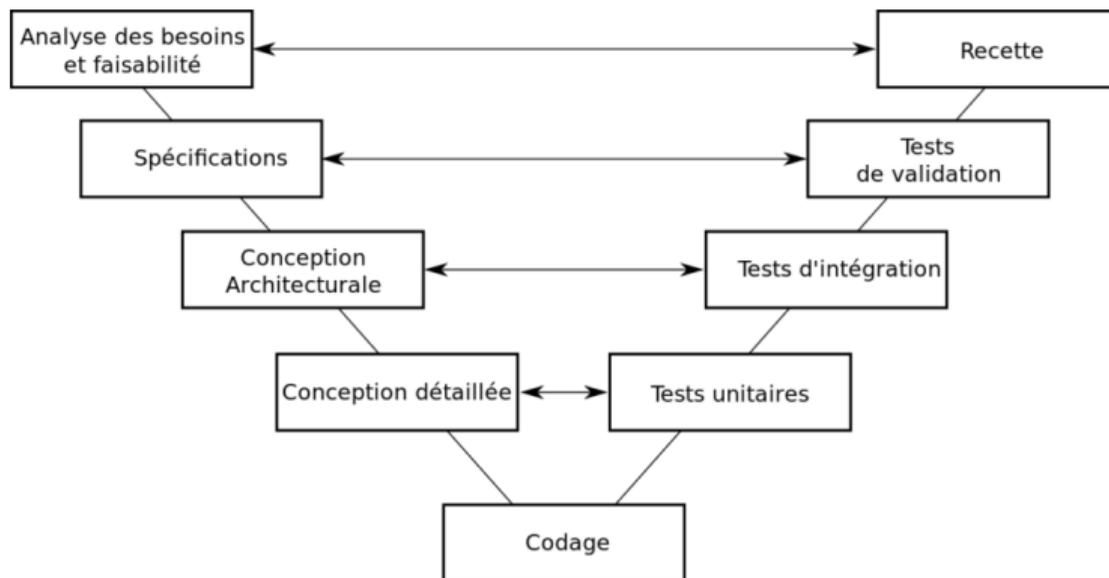
# Processus de développement

- L'équipe projet suit un processus de développement : ensemble d'activités et de résultat qui conduisent à la création d'un produit logiciel
- Le processus de développement indique la forme dans laquelle les activités sont connectées entre-elles. L'ordre dans lequel s'enchaînent les activités s'appelle le cycle de vie du produit logiciel

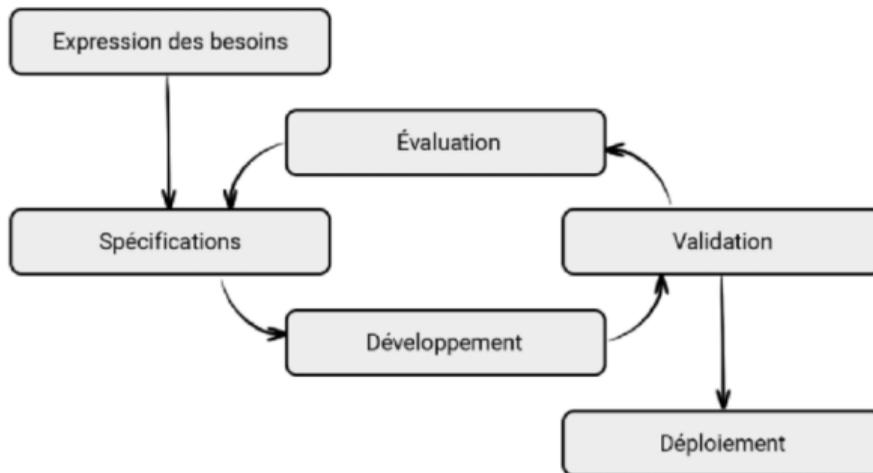
# Cycle de vie linéaire Cascade



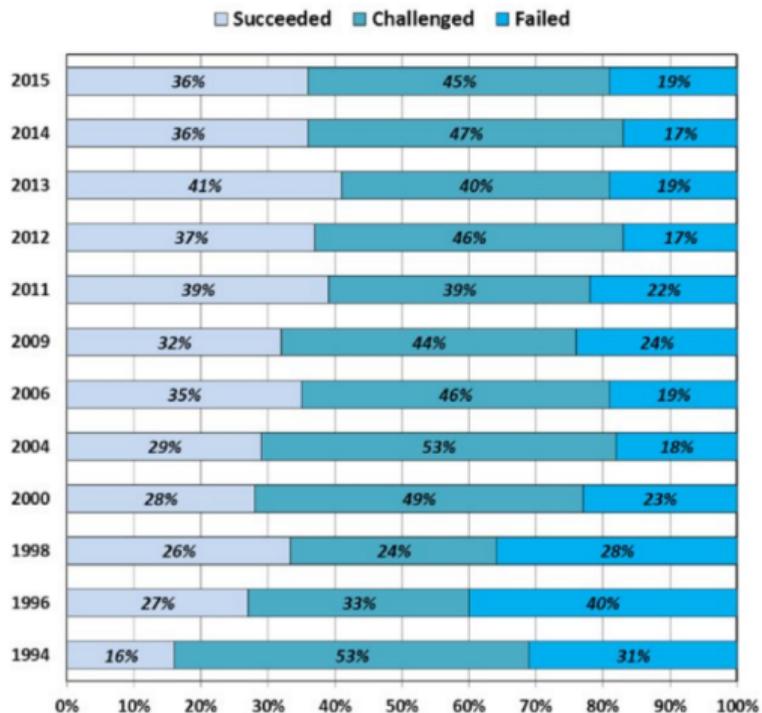
# Cycle de vie linéaire V



# Cycle de vie itérative et incrémentale



# Réussite des projets informatiques,



# Les méthodes agiles

Les méthodes agiles reposent sur des principes, également appelés les valeurs de l'agilité.

- L'interaction avec les personnes plutôt que les processus et les outils.
- Une production opérationnelle plutôt qu'une documentation pléthorique
- La négociation avec un client plutôt que le respect d'un contrat
- La collaboration au changement plutôt que le suivi du plan.

# eXtreme Programming (XP)

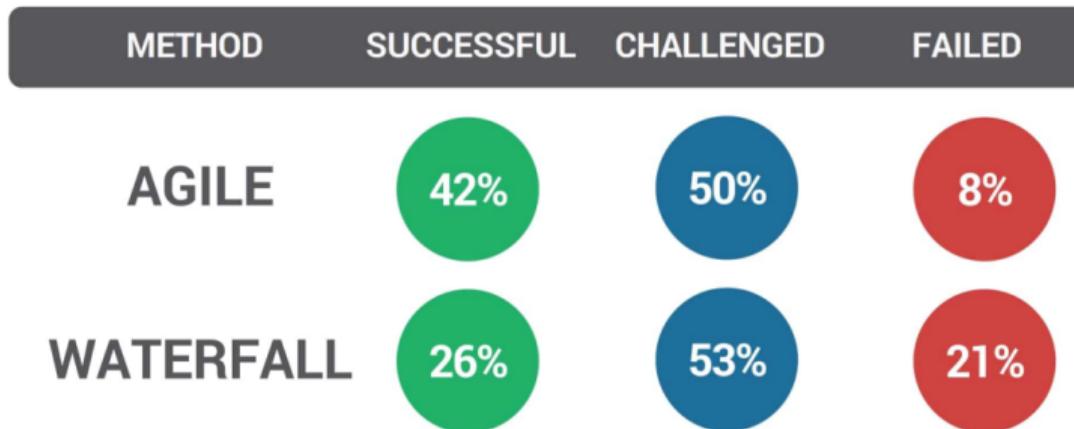
- Cycle de développement court (1 à 6 semaines)
- Cycle piloté par le client qui fournit des scénarii
- Auto-organisation de l'équipe de développement
- Programmation à tour de rôle (en binôme par exemple)
- Le logiciel est livré à l'issue d'un cycle quand les tests fonctionnels sont OK

# SCRUM

- Cycle itératif (Sprint) (2 à 6 semaines)
- Auto-organisation de l'équipe de développement
- Équipe encadrée par un SCRUM master
- Nouvelle version livrée à l'issue d'un sprint

# Gain observé entre un choix Agile

## PROJECT SUCCESS RATES AGILE VS WATERFALL



WWW.VITALITYCHICAGO.COM

Source: Standish Group Chaos Studies 2013-2017

## Processus Unifié (*Unified Process*) (UP).

### Processus Unifié

*Le processus unifié (PU), ou « unified process (UP) » en anglais, ou « Unified Software Development Process (USDP) » est une famille de méthodes de développement de logiciels orientés objets. Elle se caractérise par une démarche itérative et incrémentale, pilotée par les cas d'utilisation, et centrée sur l'architecture et les modèles UML (wikipedia)*

## Définition 3.2 (Livable)

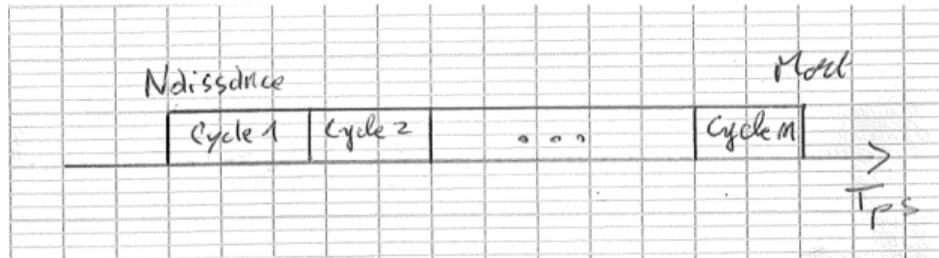
- un **corps de code source** réparti sur plusieurs composants pouvant être compilés et exécutés
- un **manuel** (et/ou une documentation technique)
- **les produits associés** (ou dépendances)

Il doit prendre en compte les besoins des utilisateurs, et de tous les intervenants (tous ceux amenés à l'exploiter) Il comprend donc :

- Un recueil des besoins
- Les cas d'utilisations (besoins fonctionnels)
- Les spécifications non fonctionnelles
- Les cas de test

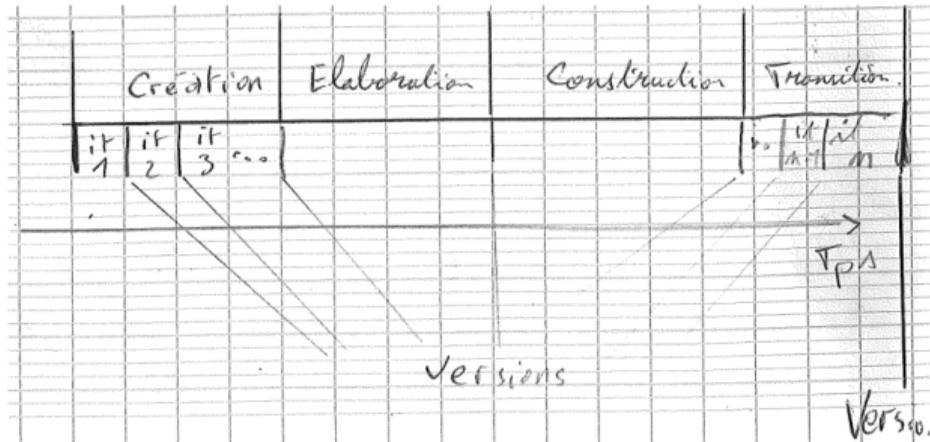
# Processus Unifié

Le Processus Unifié est un enchainement de cycles :



# Cycle

Un cycle := Phases composées d'itérations, chaque itérations est composées d'activités :



# Principes des cycles

- À chaque cycle correspond une nouvelle version du logiciel (voir Figure 5)
- Un cycle est composé de 4 phases (voir Figure 6) :
  - 1 Étude préliminaire / création
  - 2 Élaboration
  - 3 Construction
  - 4 Transition
- Chaque phase se divise en itérations