Introduction au Génie Logiciel

F. Langrognet

PLAN

1. Génie logiciel

2. Cycle de vie du logiciel
   - Etapes
   - Modèles, méthodes
   - Méthodes historiques
   - Méthodes itératives
   - Les tests au coeur du processus de développement

PLAN

1. Génie logiciel

2. Cycle de vie du logiciel
   - Etapes
   - Modèles, méthodes
   - Méthodes historiques
   - Méthodes itératives
   - Les tests au coeur du processus de développement

Le Génie Logiciel

Définition

Le génie logiciel est une science de génie industriel qui étudie les méthodes de travail et les bonnes pratiques.

Le génie logiciel s'intéresse en particulier aux procédures qui permettent de produire des logiciels qui

- correspondent aux attentes du client, aux besoins
- soient fiables
- aient un coût d'entretien réduit et de bonnes performances
- respectent les délais et les coûts de construction

1re étape : Bien se comprendre

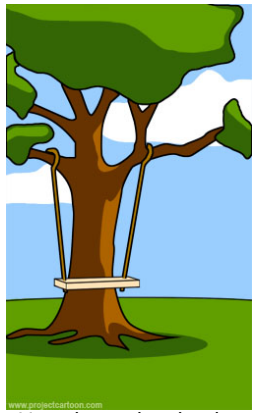
Difficultés

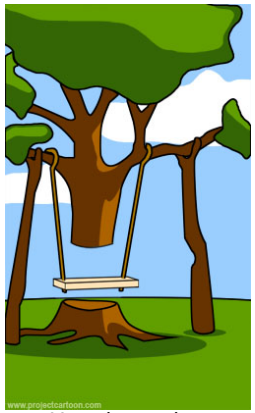
- Multiplicité des acteurs, des rôles
  - Client, utilisateur (ce n'est pas forcément le même)
  - Informaticiens
    - Chef de projet
    - Architecte
    - Développeur
    - Testeur
    - Maintenance, documentation, ...
  - Marketing, diffusion, formation
- Langages, vocabulaires différents

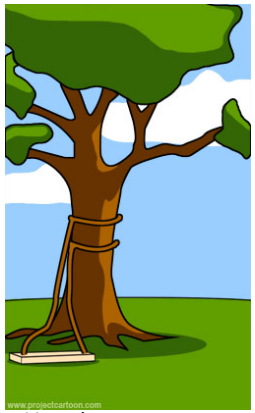
How Projects Really Work (version 1.5)

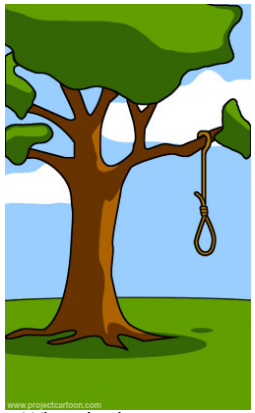
Create your own cartoon at www.projectcartoon.com

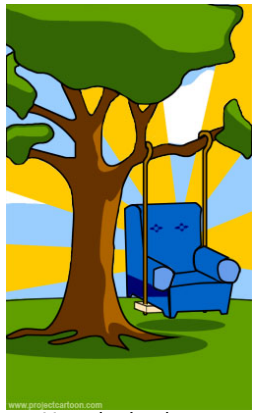
How the customer explained it

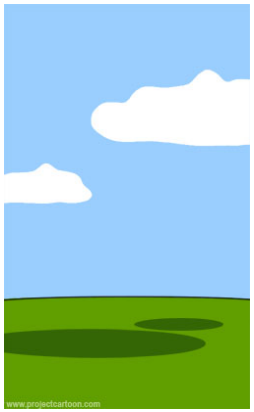

How the project leader understood it

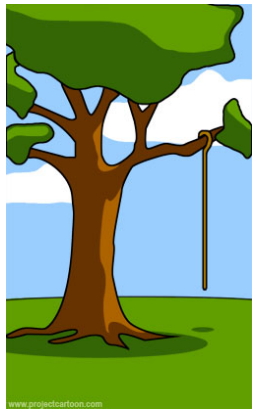

How the analyst designed it

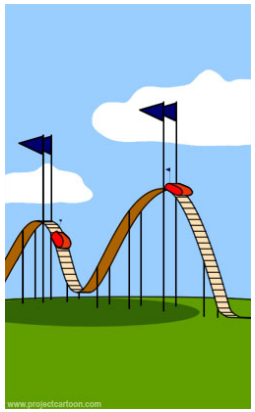

How the programmer wrote it

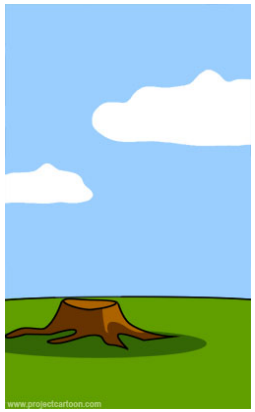

What the beta testers received


How the business consultant described it


How the project was documented


What operations installed


How the customer was billed


How it was supported

What marketing advertised

What the customer really needed

PLAN

1. Génie logiciel

2. Cycle de vie du logiciel
   - Etapes
   - Modèles, méthodes
   - Méthodes historiques
   - Méthodes itératives
   - Les tests au coeur du processus de développement

PLAN

1. Génie logiciel

2. Cycle de vie du logiciel
   - Etapes
   - Modèles, méthodes
   - Méthodes historiques
   - Méthodes itératives
   - Les tests au coeur du processus de développement

Les étapes (1)

Liste (non exhaustive...)

- Définition des besoins
  -> Dans le langage du client
- Spécifications
  Traduction des besoins dans un langage *plus informatique*
  -> Ce que doit faire le logiciel (et non comment il le fait)
- *Conception*
  Traduction des spécifications en termes de concepts logiciels
- *Codage*
  -> Traduction de la conception en code
- *Tests*
  - *unitaires (Test de chaque module individuellement)*
  - *d'intégration (Test de la composition de plusieurs modules)*

Les étapes (2)

Liste (non exhaustive...)

- Validation - Vérification
  Avons-nous construit le bon logiciel ? ->Validé
  Avons-nous bien construit le logiciel ? ->Vérifié
- Livraison / Diffusion
- Support, formation
- Maintenance
- Evolution (nouvelles versions, ...)

PLAN

1. Génie logiciel

2. Cycle de vie du logiciel
   - Etapes
   - Modèles, méthodes
   - Méthodes historiques
   - Méthodes itératives
   - Les tests au coeur du processus de développement

Modèles - Méthodes

Cycle de vie

Le cycle de vie désigne toutes les **étapes** du développement d'un logiciel, de sa conception à sa disparition.

Modèles

Les modèles décrivent les liens, les relations entre les différentes **étapes** du cycle de vie du logiciel.

Méthodes

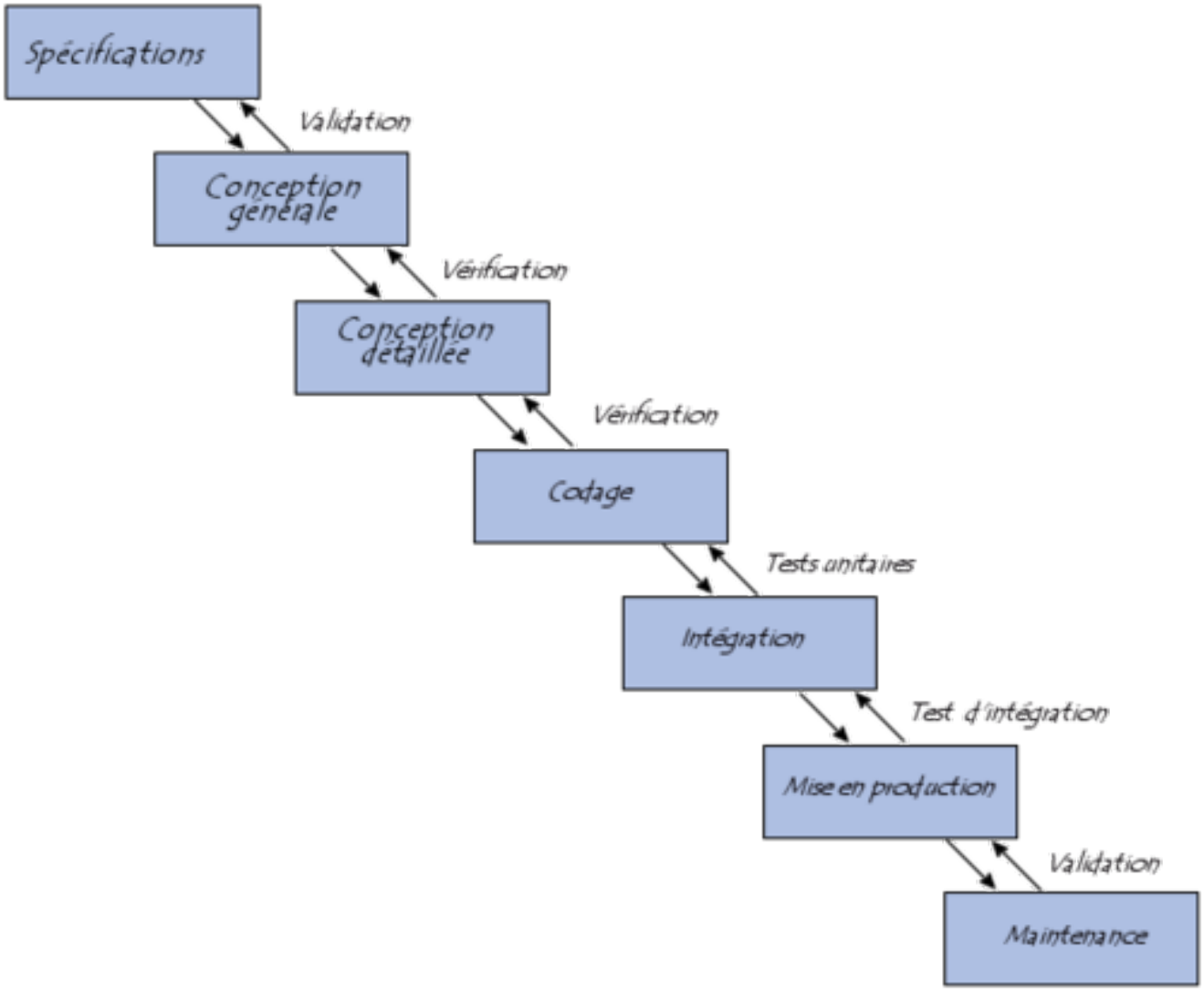
Les méthodes permettent de mettre en oeuvre un développement logiciel selon un modèle en organisant les différentes **étapes** du cycle de vie du logiciel.

PLAN

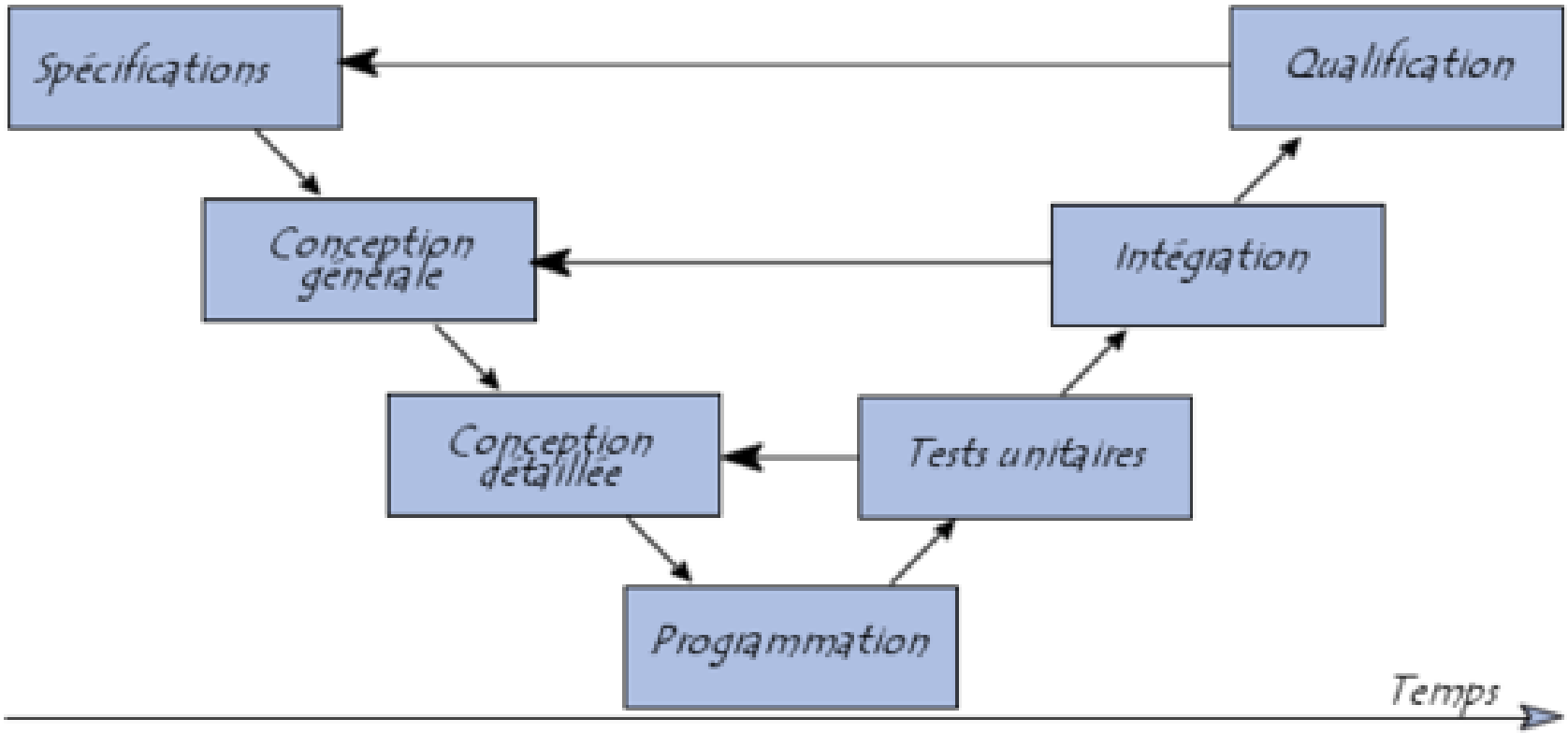
1. Génie logiciel

2. Cycle de vie du logiciel
   - Etapes
   - Modèles, méthodes
   - Méthodes historiques
   - Méthodes itératives
   - Les tests au coeur du processus de développement

Cycle en cascade



Cycle en V


Spécifications
Qualification
Conception générale
Intégration
Conception détaillée
Tests unitaires
Programmation
Temps

Méthodes tradtionnelles

Avantages

- Aisé à comprendre et à mettre en œuvre
- Forte structuration : définition puis réalisation

Inconvénents

- Modèle idéalisé, ne tient pas compte de la nature itérative d'un projet
- Logiciel délivré seulement à la fin du projet
  - Attente du client
  - Conformité, validation tardive
  - Coût élevé en cas de non-conformité

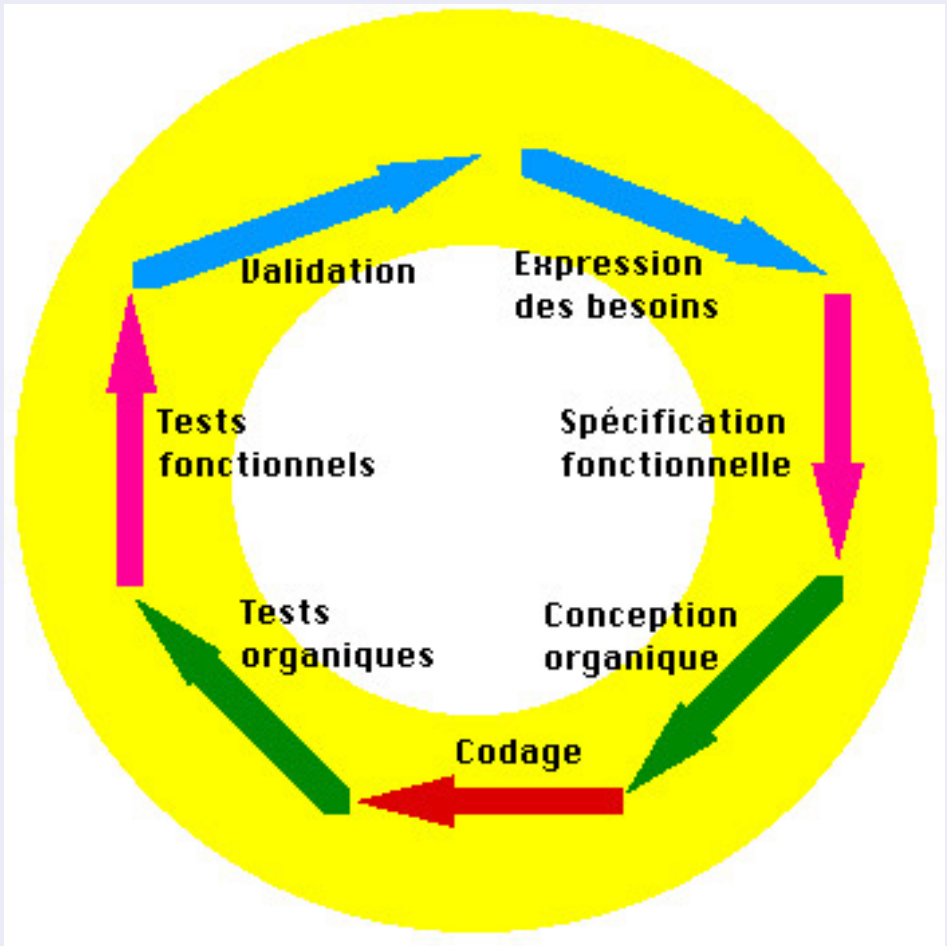
PLAN

1. Génie logiciel

2. Cycle de vie du logiciel
   - Etapes
   - Modèles, méthodes
   - Méthodes historiques
   - Méthodes itératives
   - Les tests au coeur du processus de développement

Cycle itératif (ou en spirale)

Evolution vers des méthodes plus souples, plus itératives



Cycle itératif

Avantages

- Combine les avantages des modèles en cascade/V
- Tient compte de la nature itérative d'un projet
- Bonne visibilité au cours du cycle de vie

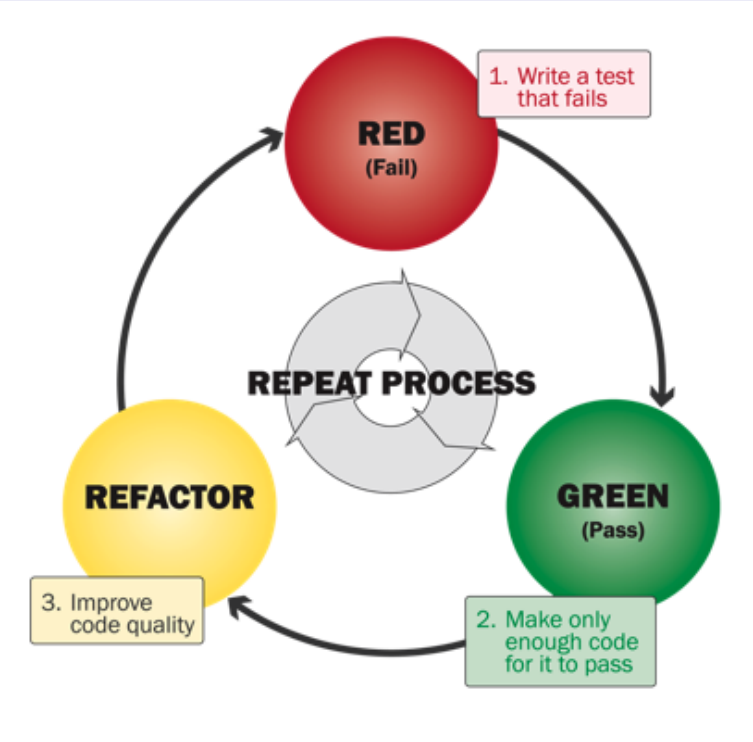
Inconvénents

- Difficile à comprendre sans être expert technique
- Nécessite une capacité à bien analyser les risques

Agilité et TDD

Le test au coeur du processus

Dans le TDD (Test Driven Development) : on écrit les tests avant le code



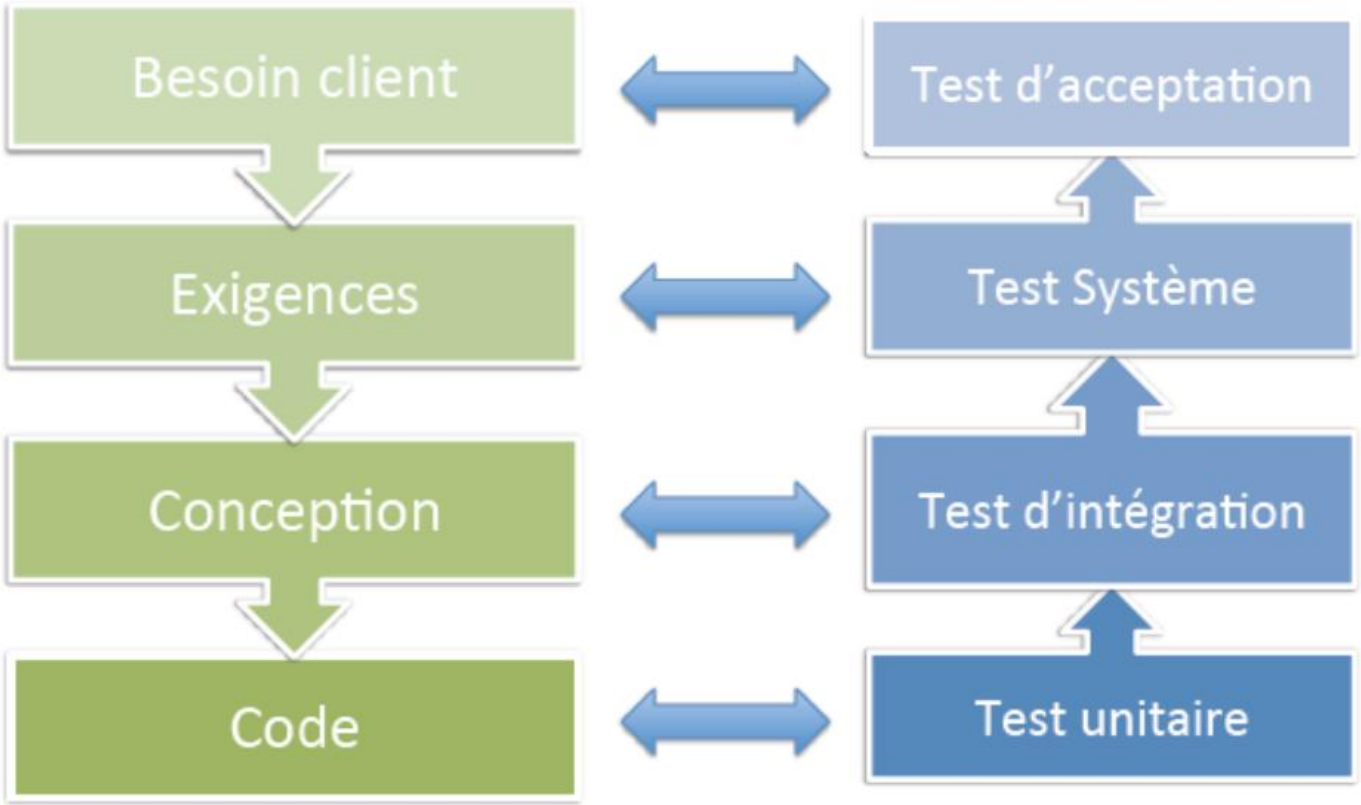
Les méthodes agiles reposent sur ce concept et intègre des règles pour diminuer les délais de mise à disposition d'une nouvelle version ('petits pas')

PLAN

1. Génie logiciel

2. Cycle de vie du logiciel
   - Etapes
   - Modèles, méthodes
   - Méthodes historiques
   - Méthodes itératives
   - Les tests au coeur du processus de développement

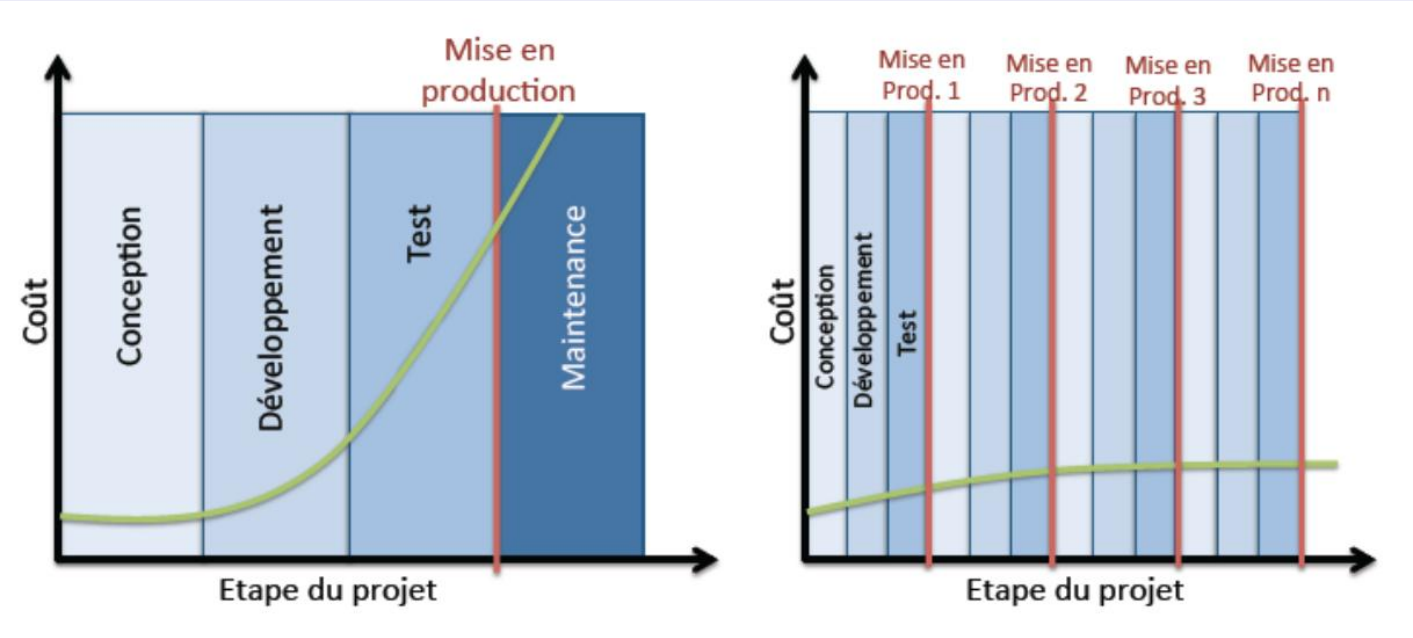
Développement et tests



Coût de résolution de bugs

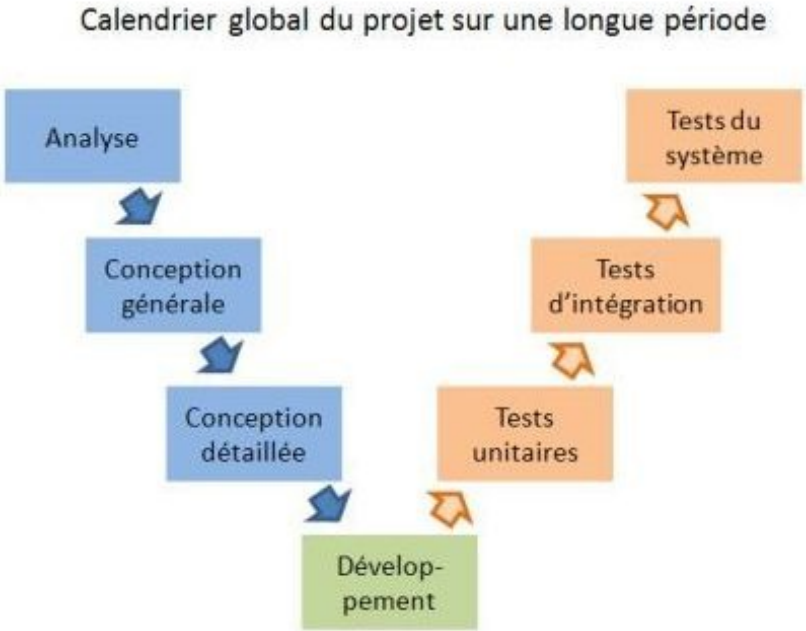
Tests et développement

- Tests unitaires
  -> 50% du coût total du projet
- Tests d'intégration (lors composition des modules, ou composants)
  -> 10 fois le coût d'un bug détecté lors des tests unitaires
- Tests système, validation (adéquation aux spécifications)
  ->100 fois le coût d'un bug détecté lors des tests unitaires

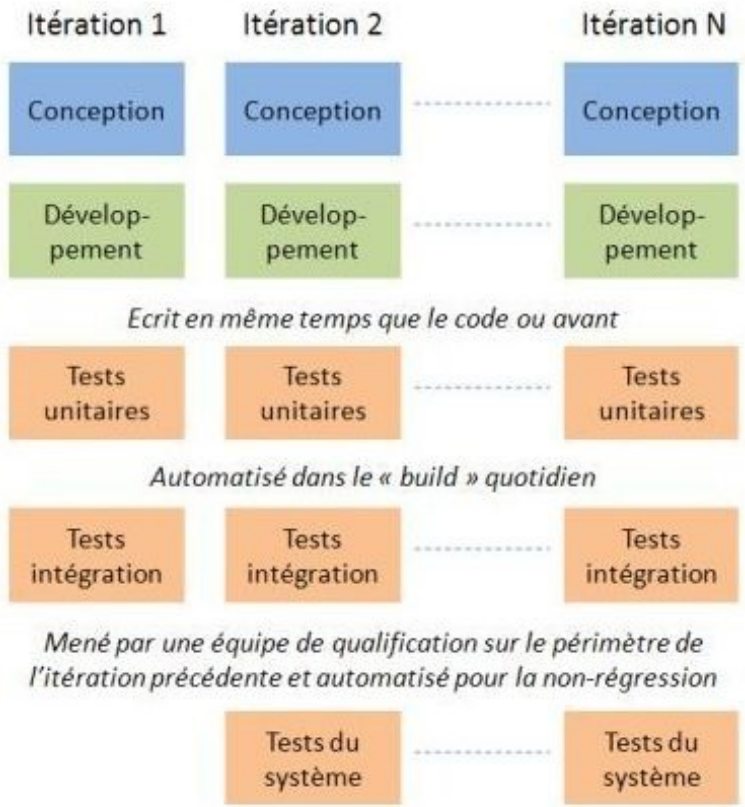


Le positionnement des tests selon l'approche projet

Cycle en V



Méthode Agile



Introduction au Génie Logiciel

F. Langrognet