



**HAL**  
open science

## Journée Thématique QeR Assurance Qualité Logiciel : Zenaida TucsnaK

► **To cite this version:**

Zenaida TucsnaK. Journée Thématique QeR Assurance Qualité Logiciel :. École thématique. Journée thématique Assurance Qualité Logiciel, France. 2019. cel-02111908

**HAL Id: cel-02111908**

**<https://hal.science/cel-02111908>**

Submitted on 26 Apr 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - ShareAlike 4.0 International License



# Journée thématique QeR Assurance Qualité Logiciel

18 avril 2019 à Paris

# Maitrise du processus de développement

50 ans d'informatique		
	<i>Logiciel</i>	<i>Langages</i>
1945	Logiciels sur mesure, calcul scientifique	Assembleur ENIAC
1958	Logiciels scientifiques et de gestion	FORTRAN COBOL
1970	<b>CRISE</b>	<b>CRISE</b>
1990	Logiciels scientifiques temps réel ERP	UNIX Bases de données
2000	E business	Client-serveur

Absence de maitrise des projets  
Mauvaise qualité des produits (logiciels défectueux)

## Réponse à la crise

Appliquer un modèle de cycle de vie

Développement  
Définition des besoins  
Spécifications  
Conception  
Codage  
Tests

Prototypage  
Documentation

Changement de phase

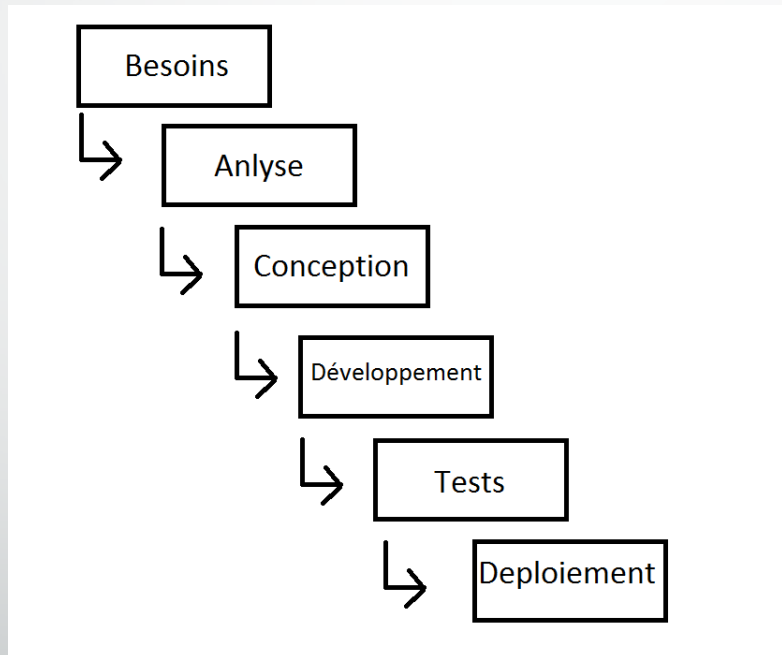
Gestion de projets

PROJET  
Besoin spécifique  
Objectif unique  
Durée limitée ( début et fin)

## Quatre phases de développement logiciel – cas d'un gros projet

1. Définition des besoins
2. Architecture et design
3. Développement
4. Test, livraison et feedback

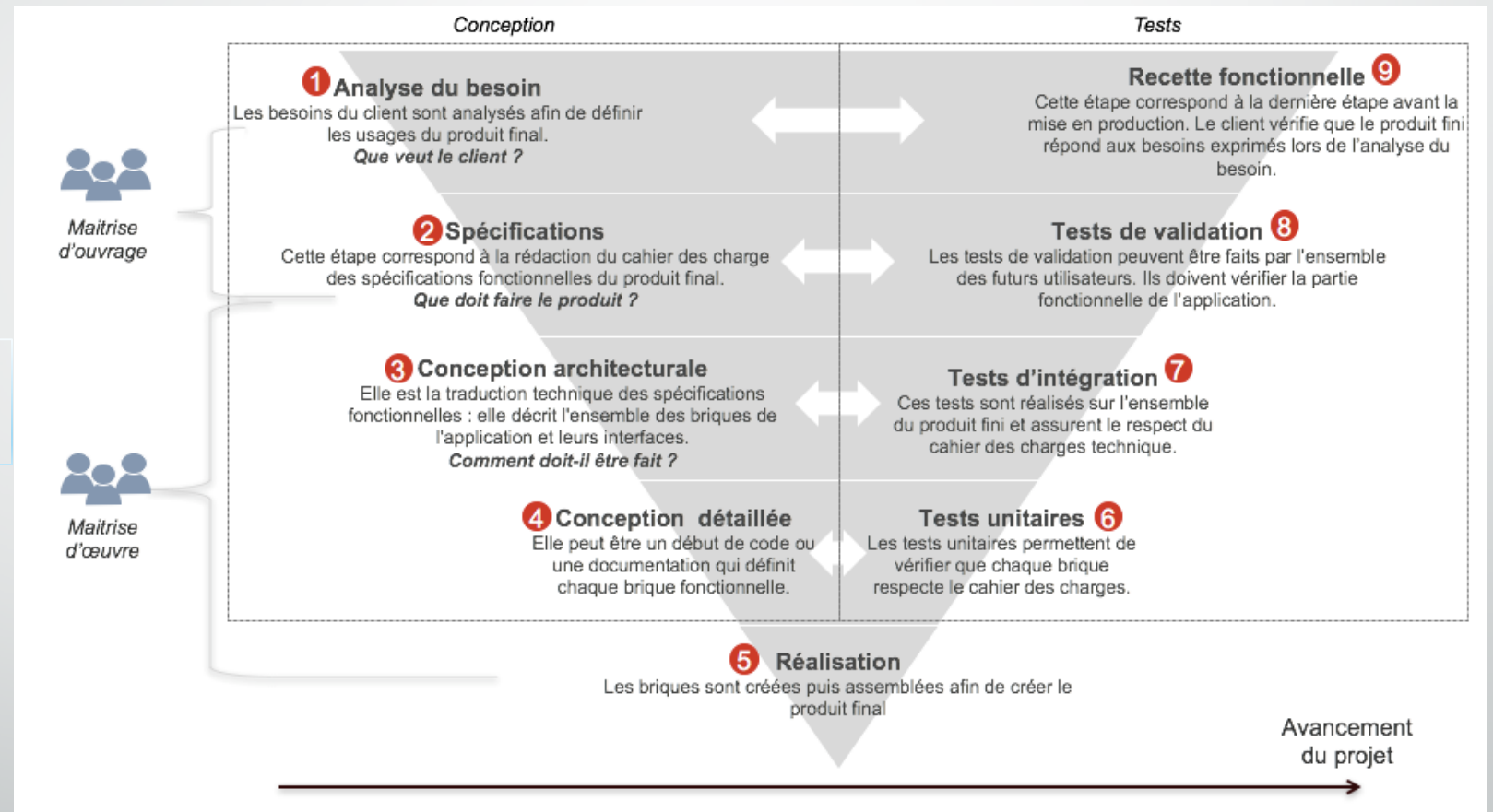
# Méthodes classiques: Modèle en cascade



# Méthode « Cycle en V »

## Etapes

Méthode introduite des 1966,  
formalisé en 1970



# Méthode « Cycle en V »

## Rôles / Maîtrise d'Ouvrage (MOA)

- Le maître d'ouvrage stratégique (MOAS)
- Le maître d'ouvrage délégué (MOAD)
- Le maître d'ouvrage opérationnel (MOAO)
- L'assistant à la maîtrise d'ouvrage (AMOA ou AMO)
- L'expert métier
- L'utilisateur , au service duquel se trouvent toutes les autres fonctions

# Méthode « Cycle en V »

## Rôles / Maîtrise d'Oeuvre (MOE)

### MOE et MOED ( Maîtrise d'œuvre délégué)

- L'équipe architecturale
- L'équipe de développement
- Titulaire du marché

Répartition des rôles en fonction des étapes

ROLES	Analyse de besoins et faisabilité	Spécifications	Conception architecturale	Conception détaillé	Codage	Test unitaires	Test d'intégration	Test de validation	Recette fonctionnelle
FONCTIONNEL MOA-AMOA									
SYSTÈME MOE-MOED									
TECHNIQUE Equipe Architecturale									
Equipe de développement									



# Méthode « Cycle en V »

## Avantages/Risques

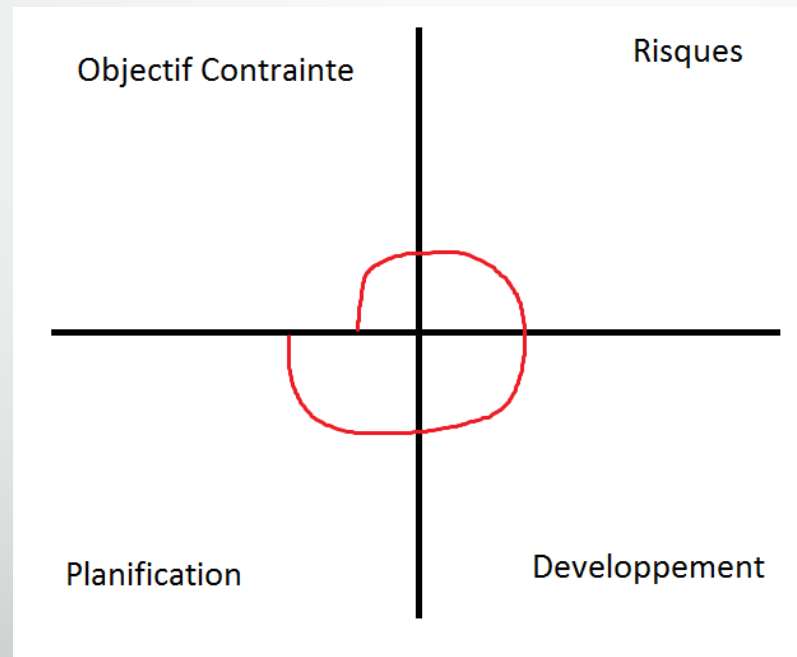
Acteurs principaux: le chef de projet, le MOA (maitrise d'ouvrage) et le MOE (maitrise d'œuvre).

**Avantage :** gains d'échelle

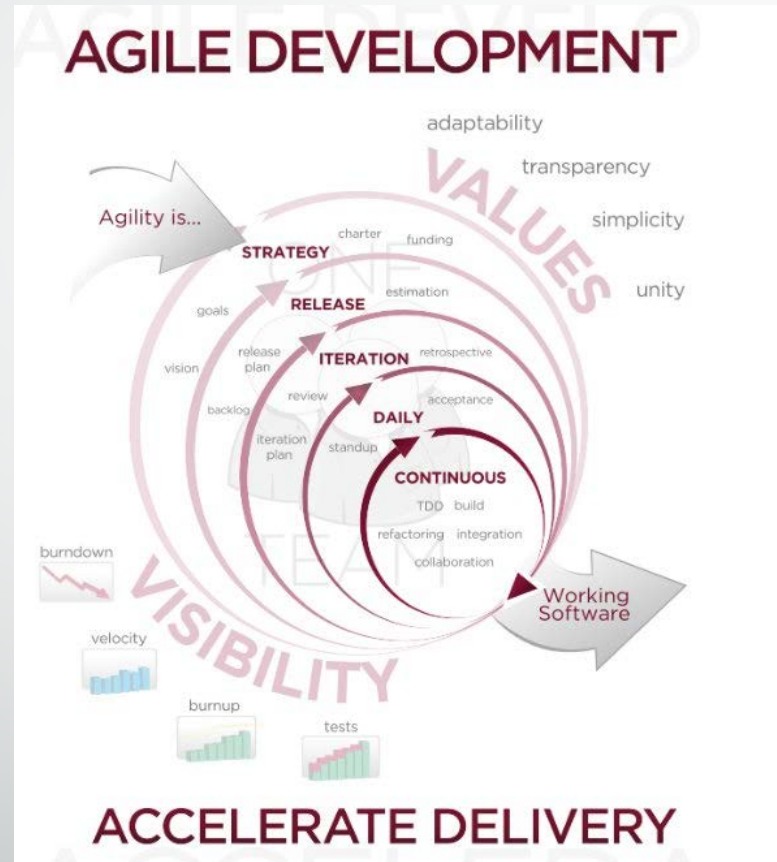
**Risques :**

- Manque de communication
  - Manque de souplesse
- Durée potentiellement longue
  - Manque de feedback

# Cycle en spirale (itératif et incrémental) (Boehm, 1988)



# Méthodes agiles



# Manifeste agile – 2001

## un changement culturel

### Valeurs principales

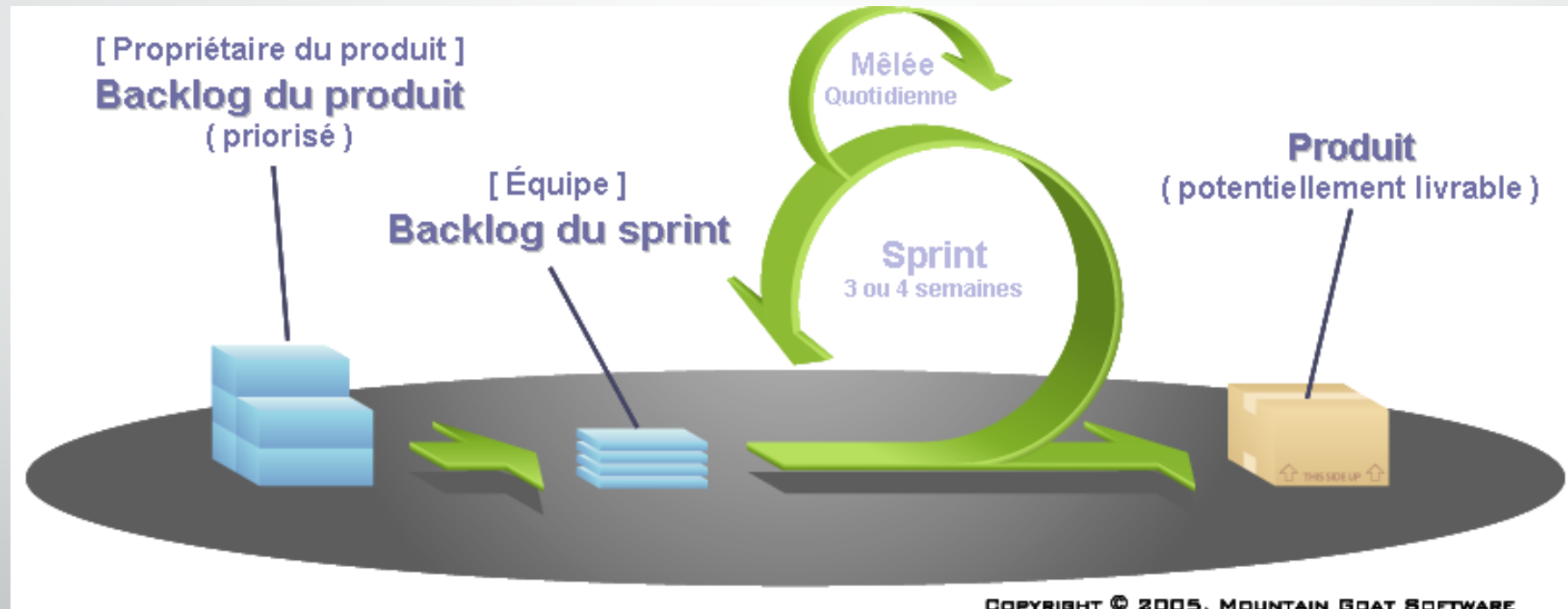
1. La priorité est donnée aux personnes ainsi qu'à leurs interactions plutôt qu'au processus et aux outils.
2. La priorité est donnée à la collaboration avec le client plutôt qu'à la négociation avec le client.
3. La priorité est donnée à la production de fonctionnalité plutôt qu'à la rédaction de document.
4. La priorité est donnée à s'adapter ainsi qu'à l'éventualité d'un changement plutôt qu'au suivi du plan d'origine.

### Principes fondamentaux

1. Priorité à la satisfaction client
2. Les changements dans le projet ne sont pas un point négatif, ils doivent être exploités pour satisfaire au mieux les clients.
3. Des livraisons régulières d'un logiciel opérationnel sous formes de cycles le plus court possible.
4. Le développement du projet devra être fait de façon collaborative entre les développeurs, les utilisateurs et les représentants et ce durant toute la durée du projet.
5. Le projet devra être réalisé par des personnes motivés.
6. La transmission d'information devra se faire par le biais d'un dialogue en face à face.
7. L'avancement du projet est connu par le fait que le logiciel livré à l'issue de chaque cycle fonctionne.
8. Le rythme de travail devra être constant et soutenable par tous les membres participants au projet
9. L'excellence technique ainsi que la bonne conception permettent de renforcer l'agilité.
10. Minimiser la quantité de travail sera une partie indispensable de l'agilité.
11. Une équipe auto organisée fournira les meilleures architectures, spécifications et conceptions.
12. Adapter et faire évoluer son comportement

# Méthodes agiles

## Méthode SCRUM (1996)



# Les piliers de SCRUM

1. Transparence : langage commun entre l' équipe et le management
2. Inspection : faire le point sur les différents artéfacts produits
3. Adaptation : si une dérive est constatée pendant l'inspection, le processus doit être adapté

Évènement: Réunion de planification de sprint

Évènement: Mêlée quotidienne

Évènement: Revue de sprint

Évènement: Rétrospective de sprint

# Les rôles dans le Framework Scrum

1. Propriétaire du produit (Product Owner) : vision du produit à réaliser (client)
2. Maître de mêlée (Scrum Master) : garant de la méthodologie Scrum
3. L'équipe de développement qui réalise le produit

# La vie d'un projet Scrum

- **Planification du Sprint** (Sprint = itération)
- **Revue de Sprint**
- **Rétrospective de Sprint** (principe d'**amélioration continue**)
- **Mêlée quotidienne**

1. Qu'est ce que j'ai terminé depuis la dernière mêlée ?
2. Qu'est ce que j'aurai terminé d'ici la prochaine mêlée ?
3. Quels obstacles me retardent ?



# Quelques évènements et artéfacts

- *Definition of done* ou *DoD* :  
**définition de fini**
- **Sprint** (itération)
- **Product Backlog** (Carnet du produit) : il n'est jamais fini
- **Sprint Backlog** (Carnet de sprint)

# Méthode SCRUM

## Avantages/Risques

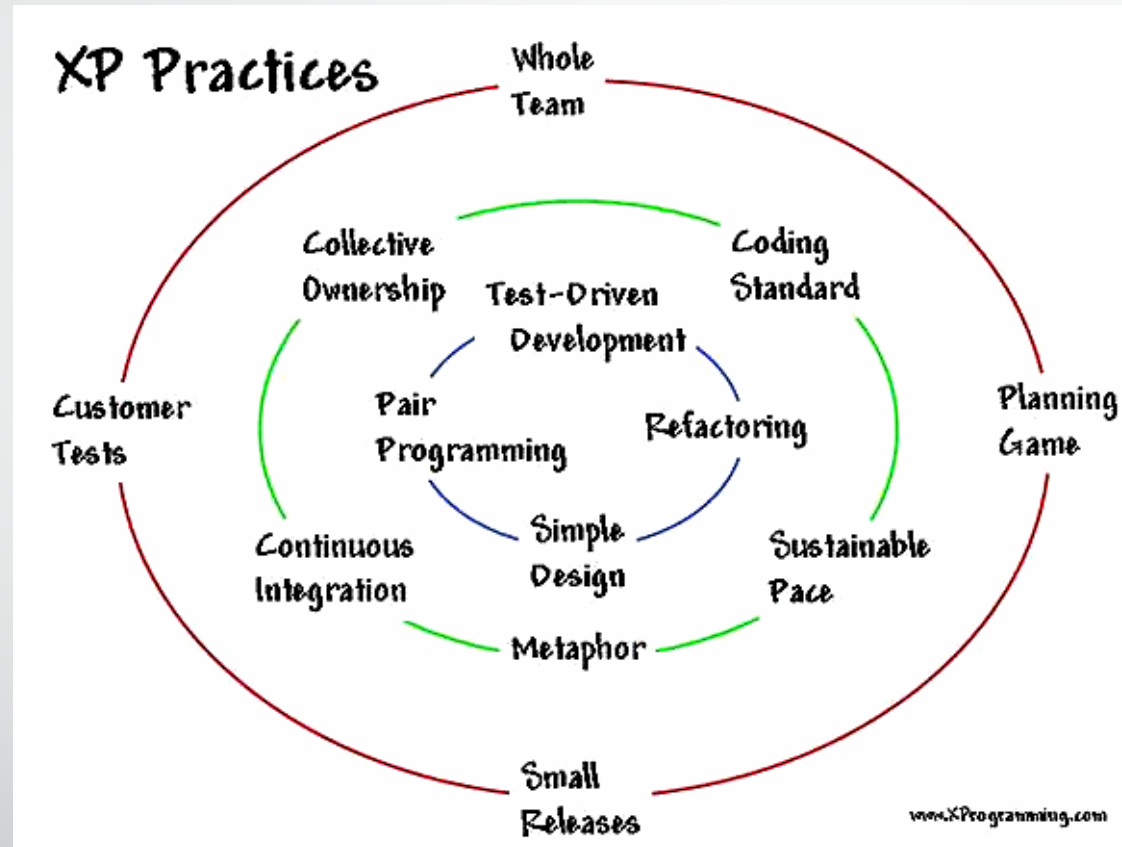
### Avantages :

- produit proche des besoin du client
- La communication entre les acteurs du projet est facilité

### Risques :

- Manque de documentation
- Mauvaise interprétation des « daily scrum meetings »

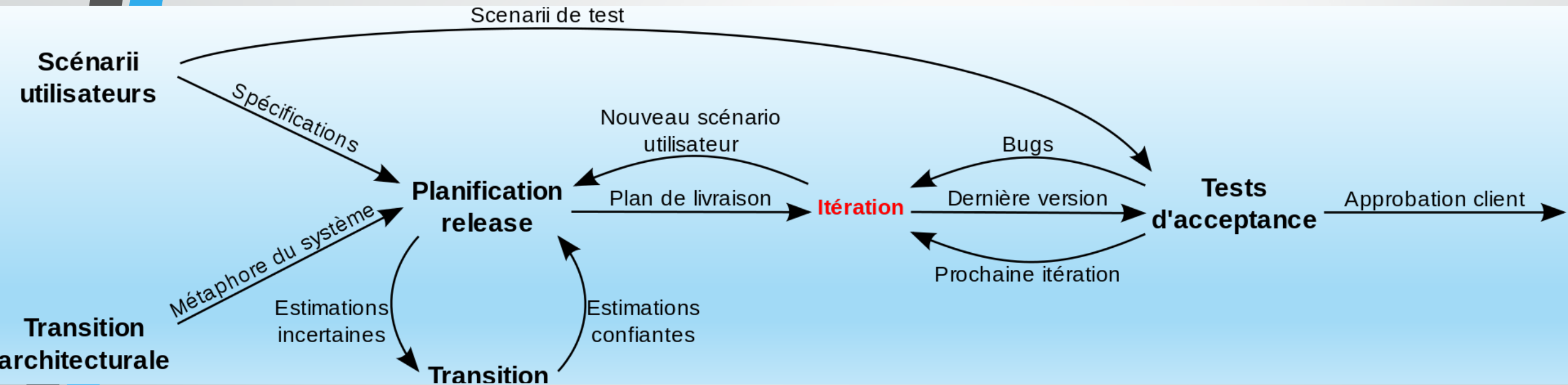
# Méthode agile eXtreme Programming



# Les valeurs dans la méthodologie eXtreme

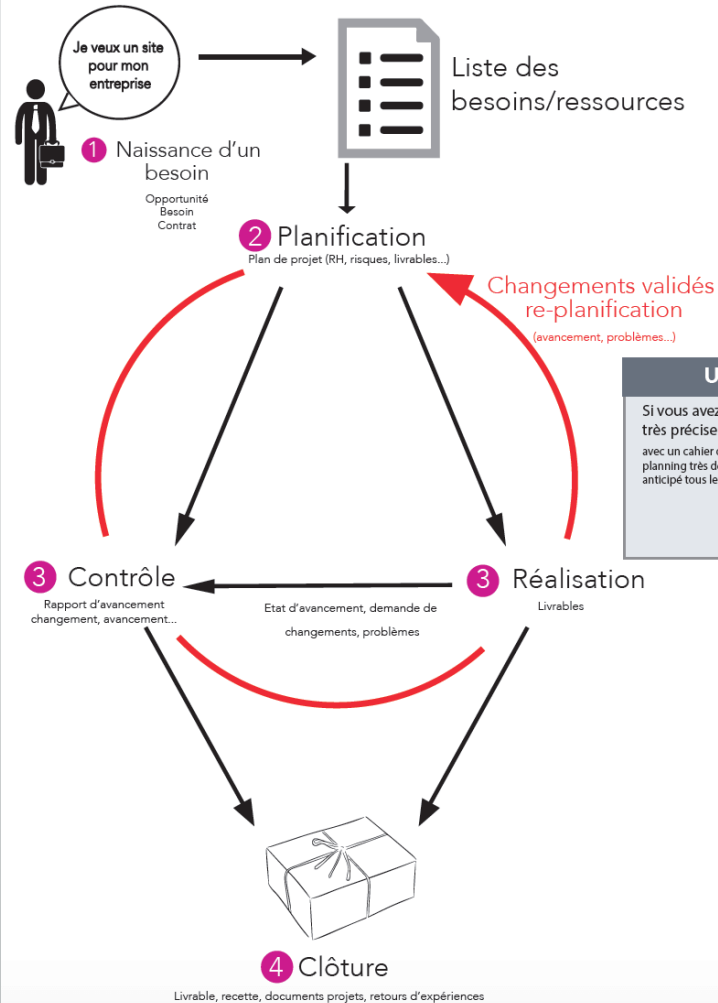
1. Communication ( programmation en binôme)
2. Simplicité
3. Feedback
4. Courage (estimations, changements, adapter les processus)
5. Respect ( qualité, meilleure solution, refactoring)

# Conception et développement



# Méthode Agile vs Méthode Classique

## MÉTHODE CLASSIQUE (PRÉDICTIVE, CYCLE EN V, EN W...)



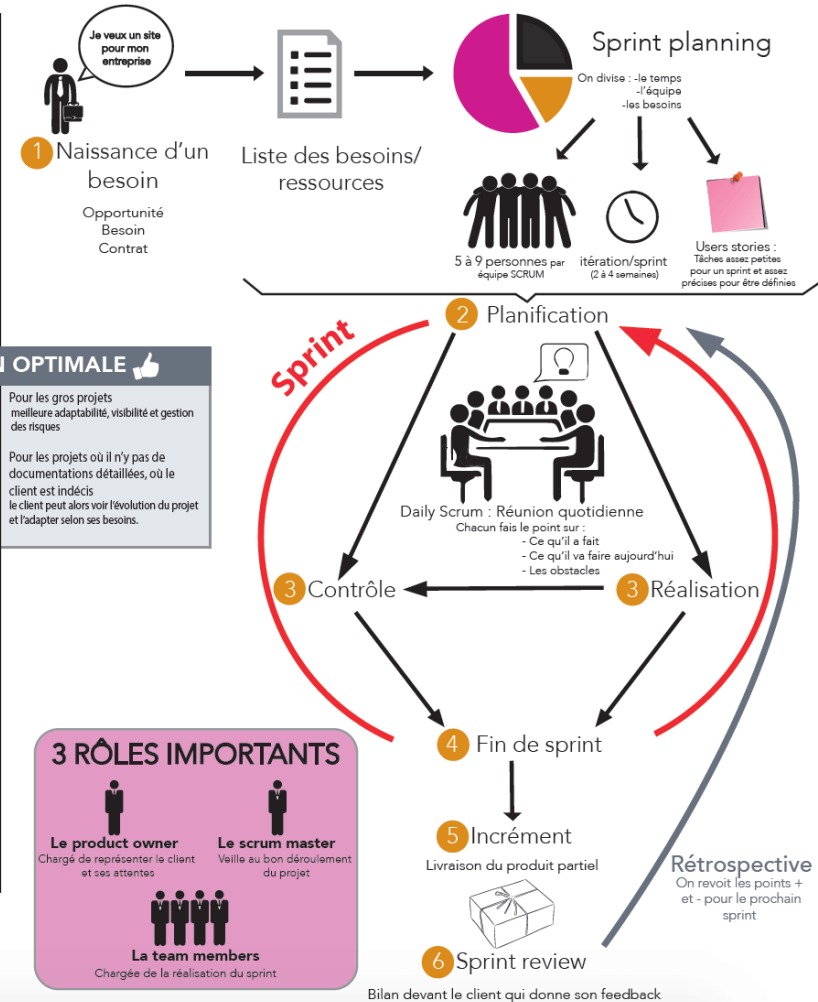
### UTILISATION OPTIMALE

Si vous avez une idée très précise de votre projet avec un cahier des charges et planning très détaillé ou vous avez anticipé tous les risques possibles

Pour les gros projets meilleure adaptabilité, visibilité et gestion des risques

Pour les projets où il n'y pas de documentations détaillées, où le client est indécis le client peut alors voir l'évolution du projet et l'adapter selon ses besoins.

## MÉTHODE AGILE SCRUM



### 3 RÔLES IMPORTANTS

**Le product owner**  
Chargé de représenter le client et ses attentes

**Le scrum master**  
Veille au bon déroulement du projet

**La team members**  
Chargée de la réalisation du sprint