

Cycles de développement

Cycles de développement

- **Plan**
 - **Objectifs des diagrammes UML**
 - **Cycle de développement**
 - **Cycle en V**
 - **Cycle en Y**
 - **« Cycle » agile**

Objectif des diagrammes UML

Objectif des diagrammes UML

- **Objectifs 1/6**

- **Diagramme des acteurs**

- Définir qui utilise le système
 - Définir les frontières du système

- **Diagramme des cas d'utilisation**

- Définir les besoins de chaque acteur
 - Définir les données échangées entre le système et les acteurs
 - Définir les données manipulées par le système
 - Définir les traitements effectuées sur les données

Objectif des diagrammes UML

- **Objectifs 2/6**

- **Diagramme d'activités**

- **Modéliser les actions du système par rapport à celles des acteurs**
 - **Modéliser les actions du système par rapport à celles des systèmes externes**

- **Diagramme de déploiement**

- **Modéliser les plate formes techniques**
 - **Production**
 - **Intégration**
 - **Développement**

Objectif des diagrammes UML

- **Objectifs 3/6**

- **Diagramme de déploiement**

- **Prendre en compte l'ensemble des contraintes opérationnelles**

- **Flux**
 - **Volume**
 - **Sécurité**
 - **Rapport**
 - **Administration**
 - **Sauvegarde / Restauration**

Objectif des diagrammes UML

- **Objectifs 4/6**
 - **Diagramme de composants**
 - **Découper le système en sous parties**
 - Autonomes
 - Remplaçables
 - Communiquant par interfaces
 - **Diagramme de packages**
 - Définir l'arborescence des sources
 - Définir le découpage du code

Objectif des diagrammes UML

- **Objectifs 5/6**
 - **Diagramme de classes candidates**
 - Modéliser les objets métiers
 - Modéliser les traitements métiers
 - **Diagrammes séquences et états**
 - Modéliser les appels entre les classes métiers
 - Valider le modèle métier
 - Modéliser les états des classes métiers
 - Valider les processus métiers
 - Enrichir le modèle métier avec les classes techniques

Objectif des diagrammes UML

- **Objectifs 6/6**
 - **Diagramme finalisé de classes**
 - Finaliser les diagrammes de classes avant le codage

Cycle de développement

Cycles de développement

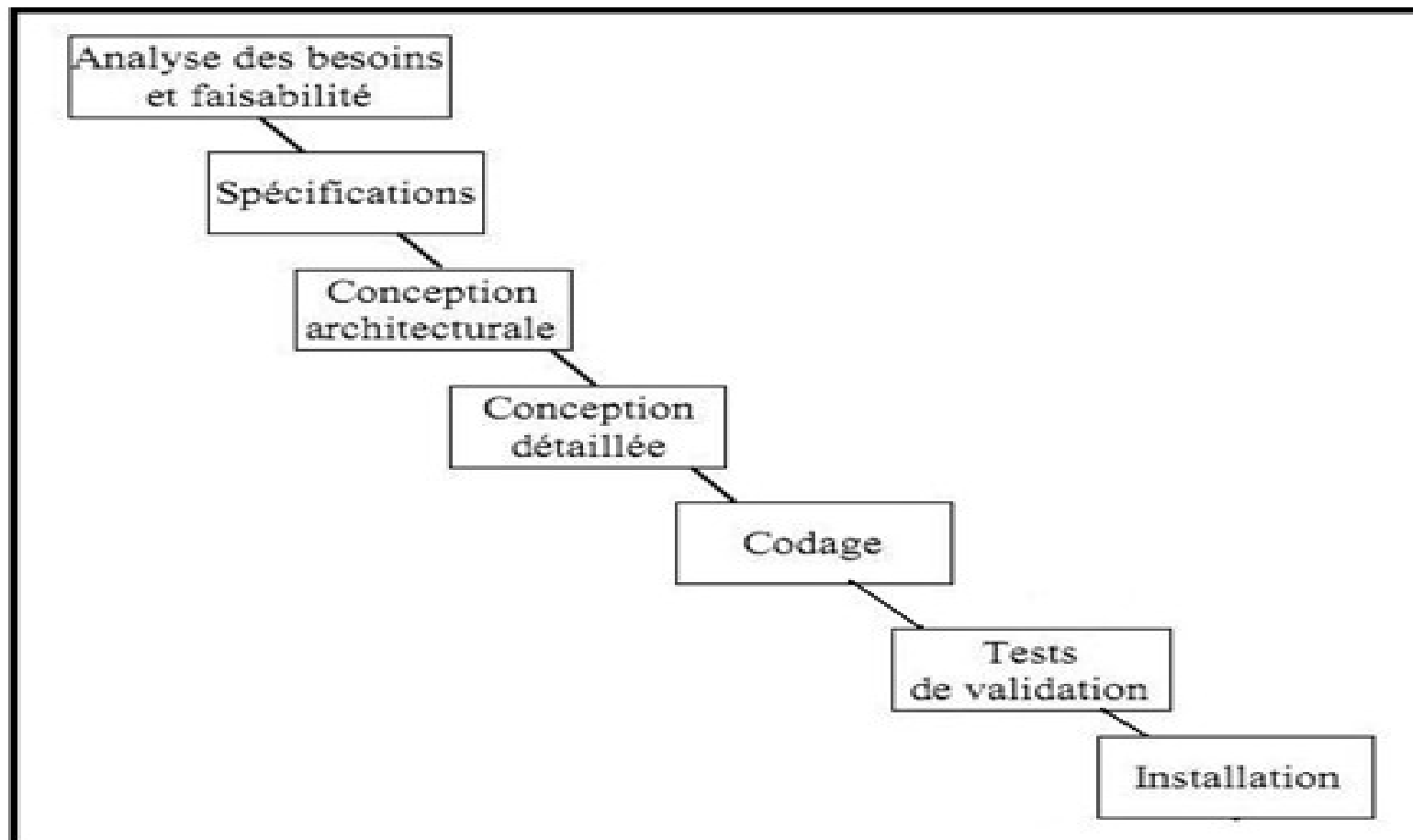
- **Cycle de développement 1/2**
 - **Contenu**
 - Définit les activités de développement
 - Définit l'ordre de ces activités
 - Définit les entrées/sorties des activités
 - Définit les parties prenantes des activités
 - **Construit par le service méthode**
 - Ou des salariés expérimentés
 - Ou des consultants ...

Cycles de développement

- **Cycle de développement 2/2**
 - **Sous contrôle du service qualité**
 - Description du processus
 - Définition des modèles de documents
 - Production d'exemples de documents
 - Définition des checks list
 - **Élément fondamental d'une structure**
 - Toujours en avoir un
 - Même embryonnaire
 - **Cycle le plus connu : cycle en V**

Cycles de développement

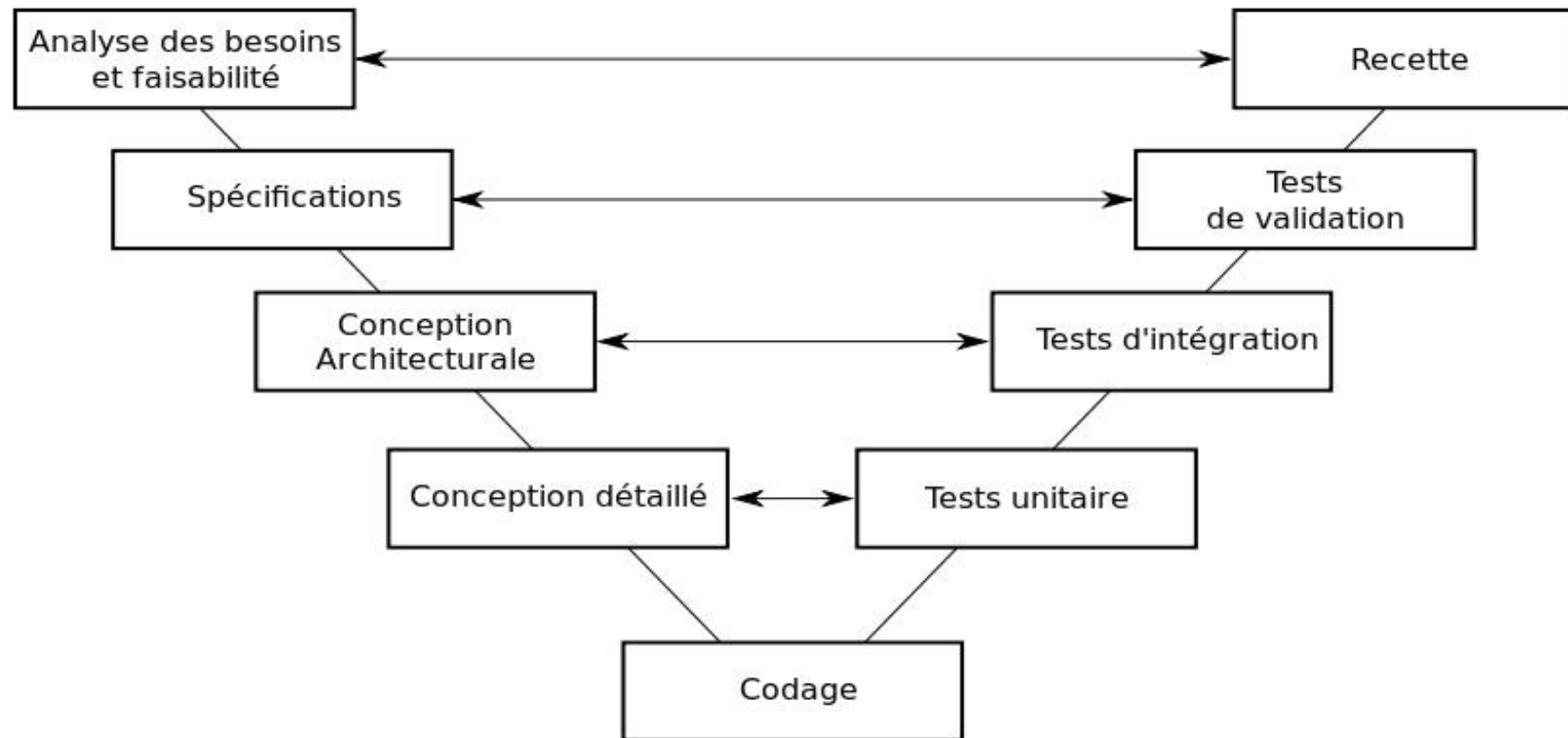
- **Cascade : un des premiers**



Cycle en V

Cycle en V

- Cycle en V 1/3



Cycle en V

- **Cycle en V 2/3**
 - **Trop séquentiel**
 - **Peu d'interactions entre le fonctionnel et le technique**
 - **Peu efficace**
 - **Effet tunnel**
 - **Pas de réutilisation**

Cycle en V

- **Cycle en V 3/3**

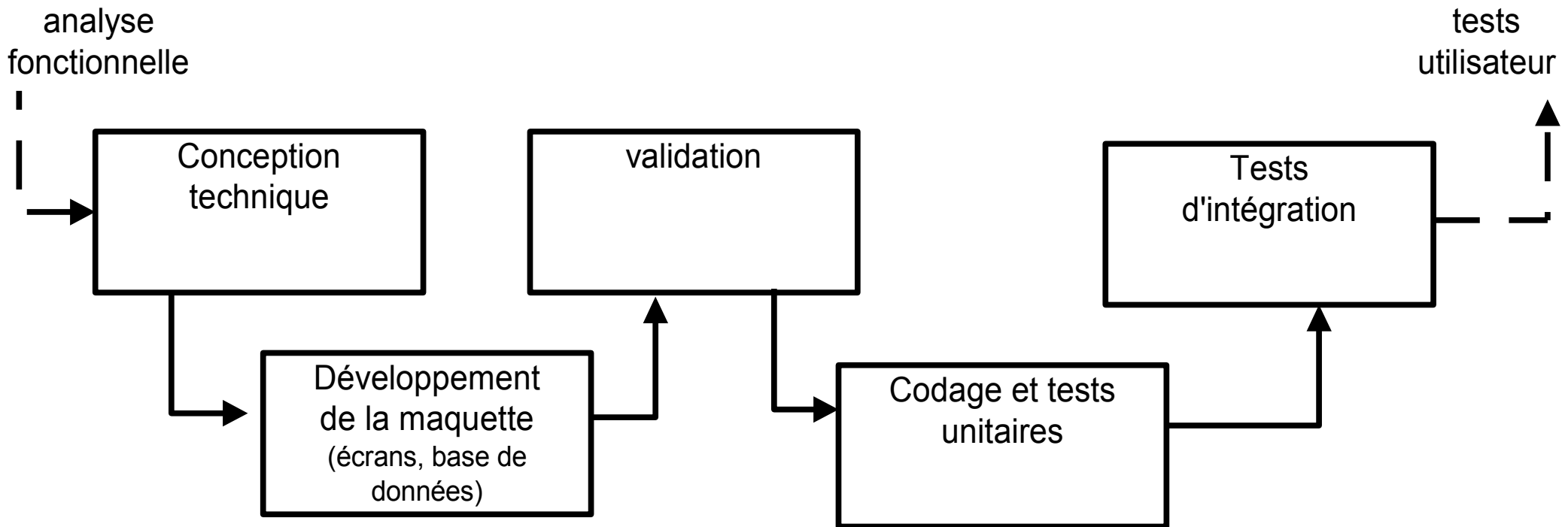
- **Mais**

- **Mets les activités de tests en lien avec les activités de développement**
 - **Facilement compris**
 - **Mise en œuvre facile**

Cycle en Y

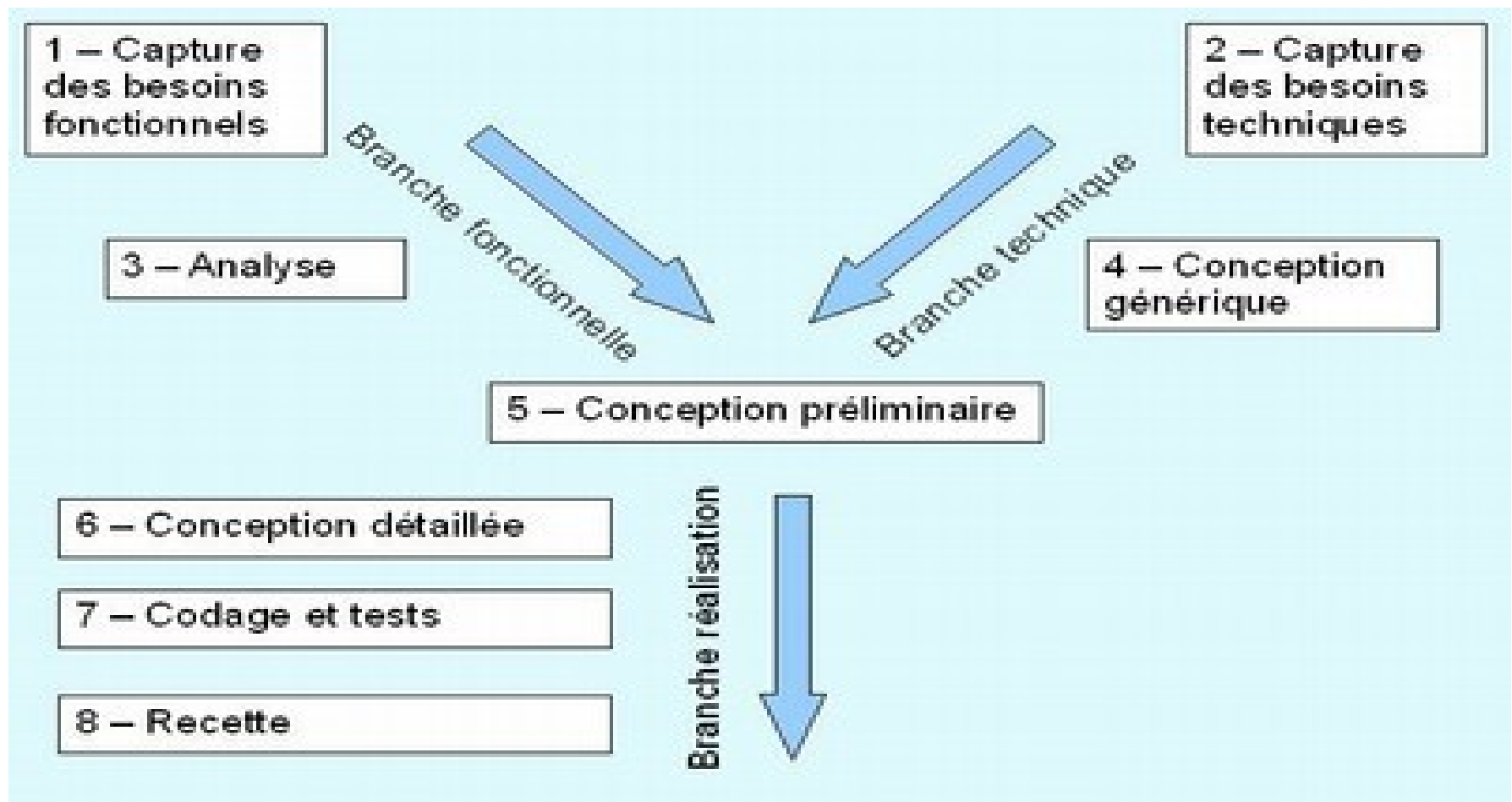
Cycle en Y

- **Intermédiaire : W**



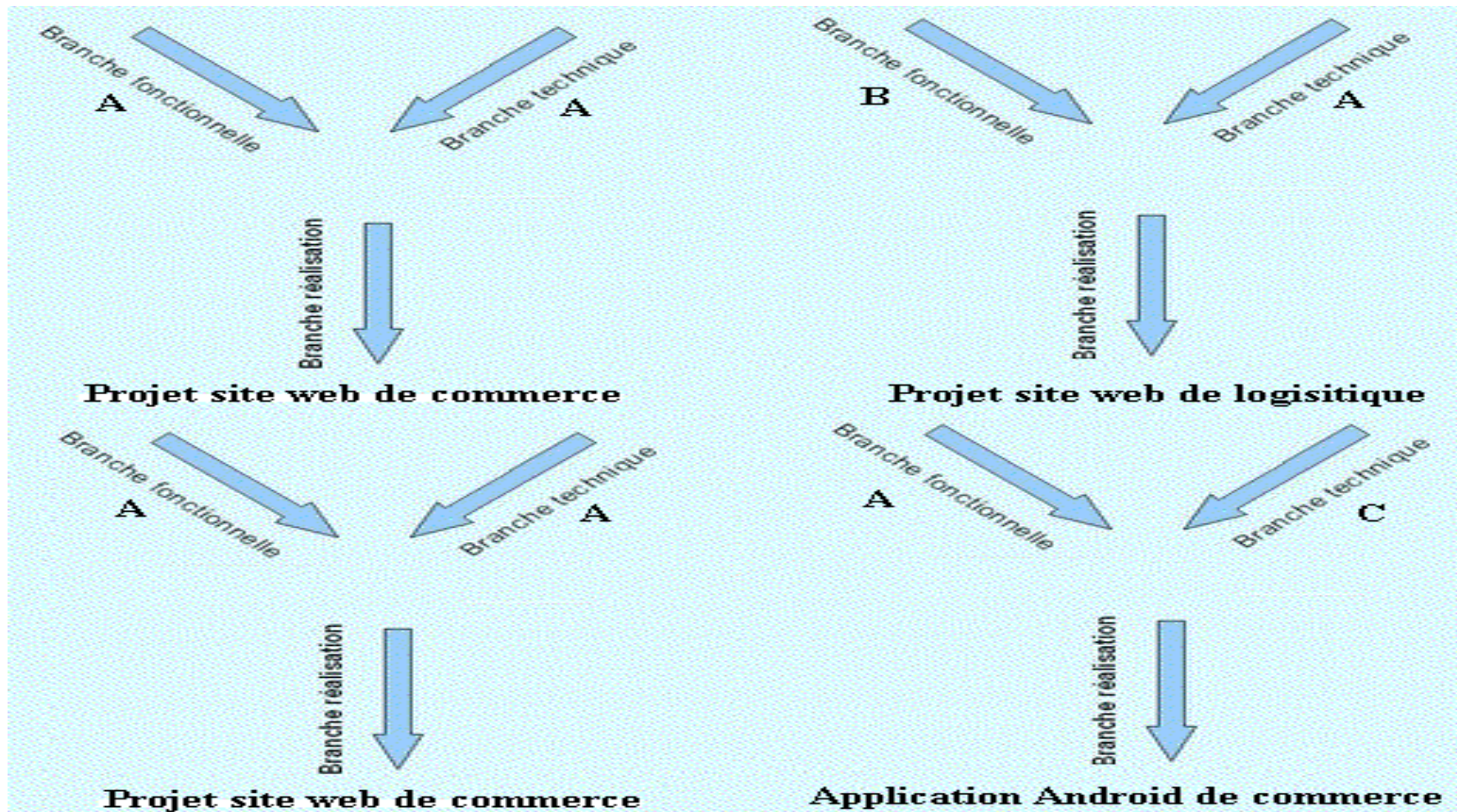
Cycle en Y

- Cycle en Y 1/4



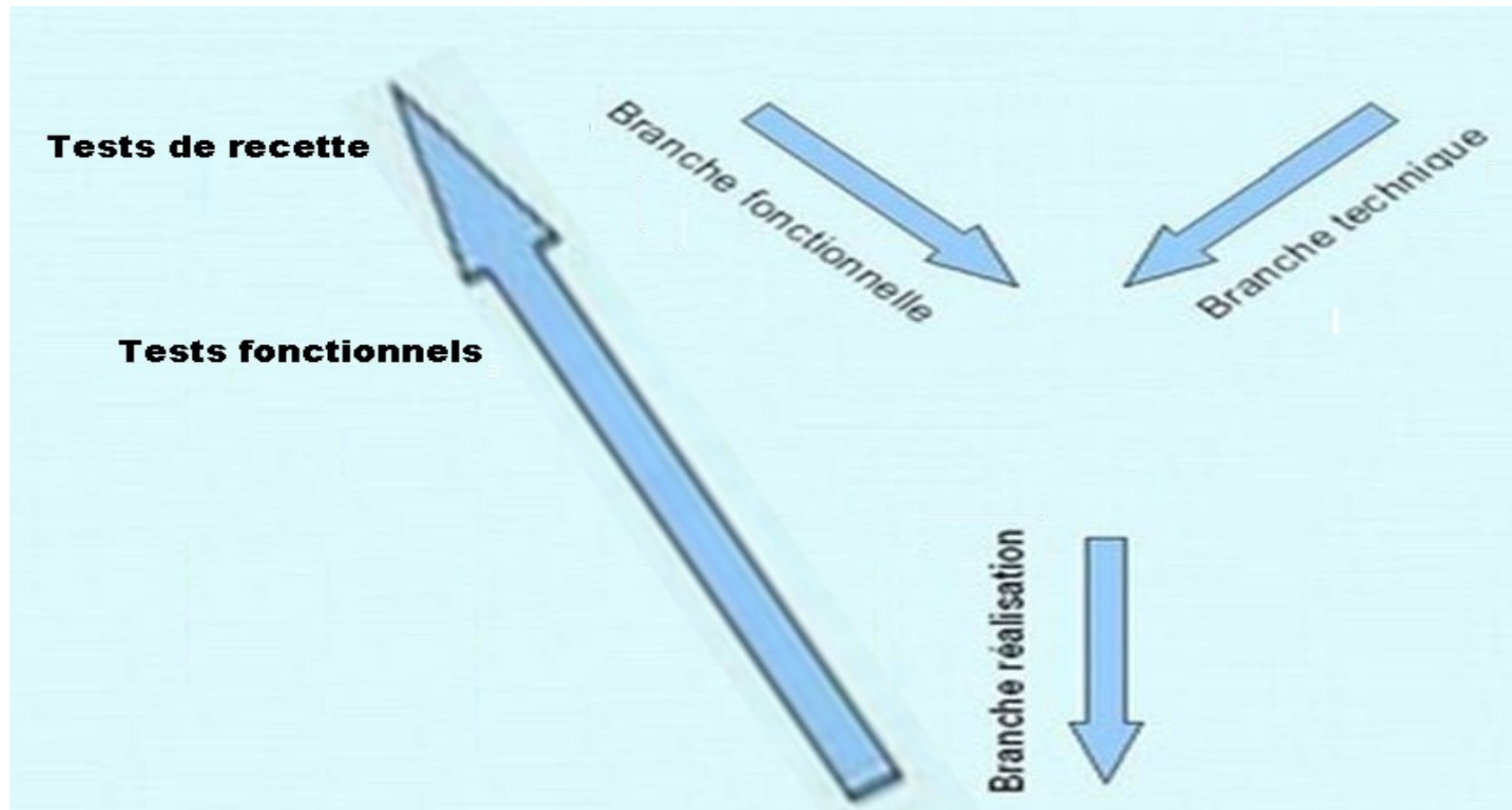
Cycle Y-V

- Cycle en Y 2/4



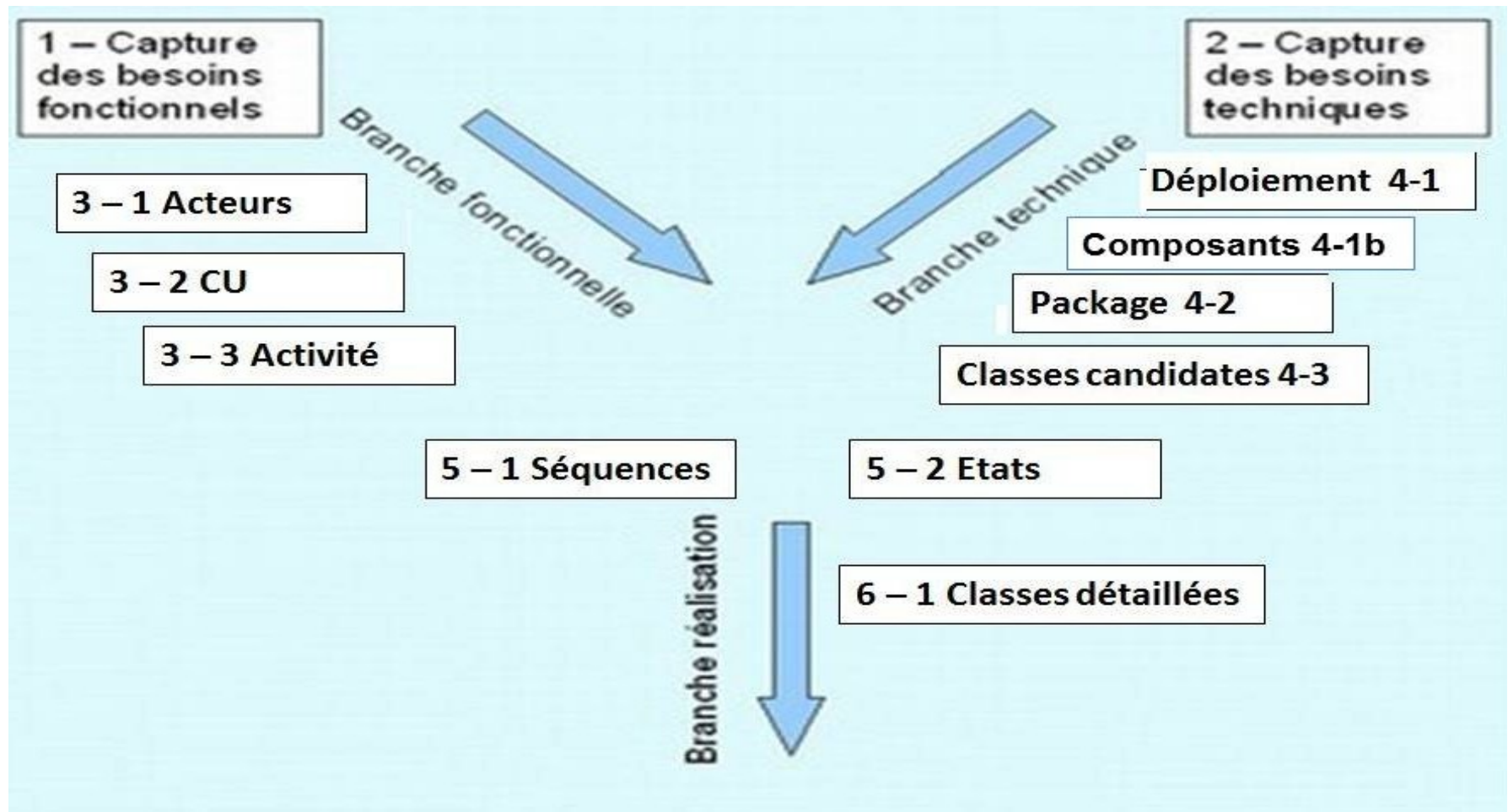
Cycle Y-V

- Cycle en Y 3/4



Cycle Y-V

- Cycle en Y 4/4



« Cycle » agile

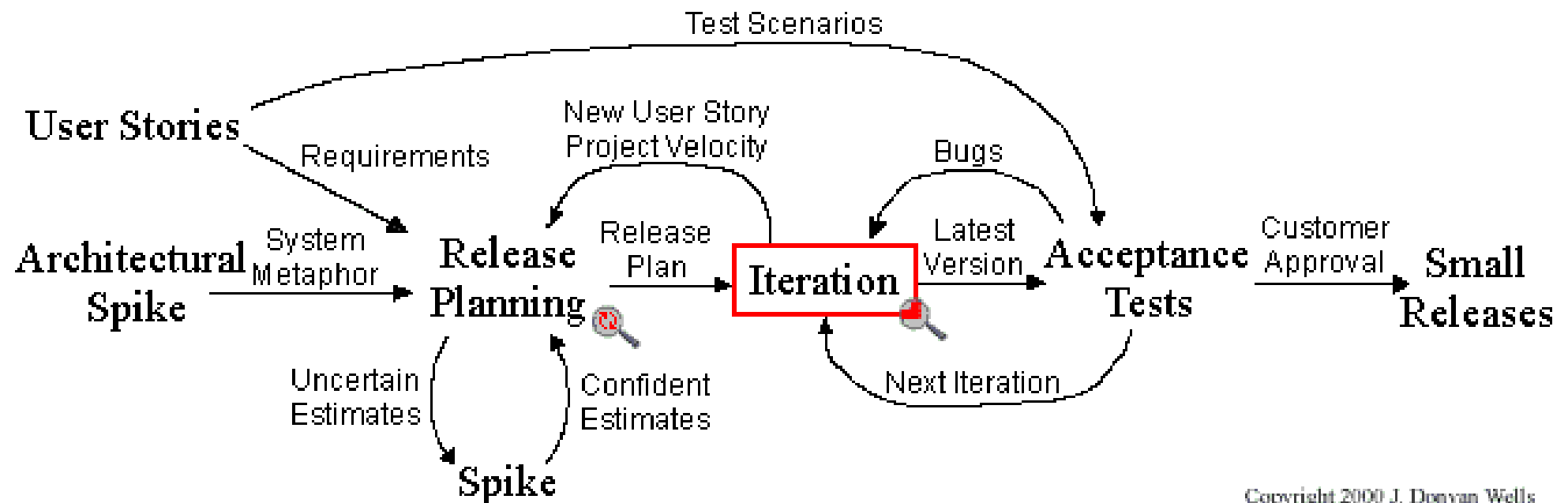
« Cycle » agile

- "Méthodes agiles"
 - un terme à la mode
 - pour toutes les méthodes qui visent un cycle de développement court
 - généralement basées sur un modèle itératif

- eXtreme programming
 - la méthode la plus connue
 - pas forcément la plus pratiquée

« Cycle » agile

Le processus XP : vue d'ensemble



« Cycle » agile

- cycles majeurs: les « releases »
 - cycle d'1 ou 2 mois
- cycles intermédiaires: les « iterations »
 - cycle d'1 ou 2 semaines
- cycles mineurs: l'intégration permanente
 - cycle de quelques minutes à une journée

« Cycle » agile

XP: Méthodes de travail

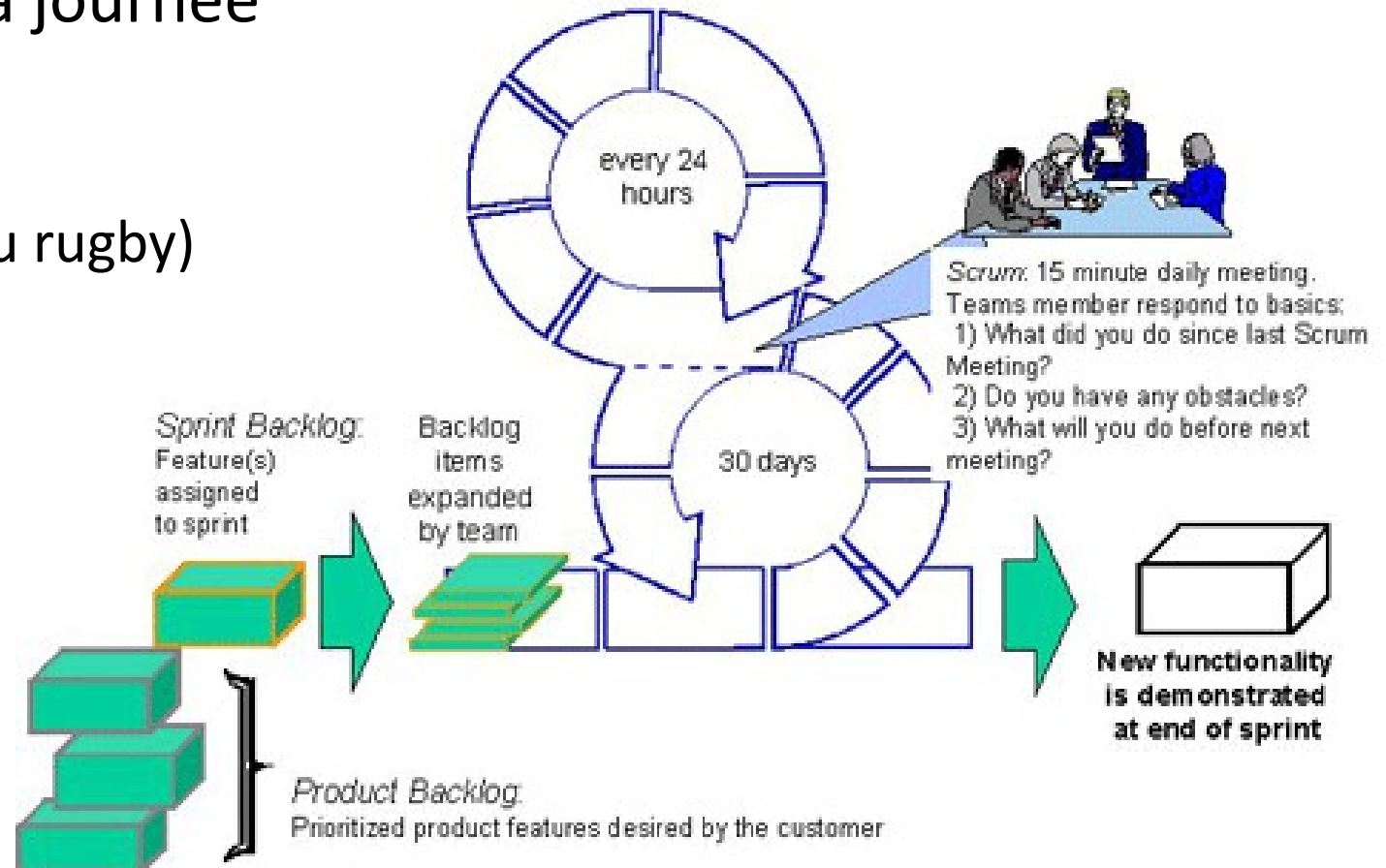
- pilotage par les besoins
- pilotage par l'architecture
- contrôle permanent de la qualité
- modélisation et documentation légères
- estimation des charges et planification empiriques

le workflow est indissociable des méthodes

« Cycle » agile SCRUM

- itératif sur
 - un cycle de 30 jours
 - un cycle de la journée

"Scrum": la mêlée (au rugby)



« Cycle » agile

Caractéristiques de SCRUM

- cycle de 30 jours
 - Sprint
 - livraison au client en fin de cycle
- cycle de la journée
 - Scrum: réunion quotidienne de 15'
- développement en 3 phases
 - initialisation
 - ♦ Sprint Backlog: liste de tâches très détaillée (une tâche = 4 à 16 heures)
 - réalisation: un sprint
 - clôture du sprint

« Cycle » agile

Positionnement de SCRUM

- différences par rapport à XP
 - organisation plus simple (et plus rigide) que celle de XP
 - fonctionnement entre collectif et hiérarchique
 - ♦ responsabilité de l'équipe
 - ♦ rôle du Scrum Master
- points communs
 - pilotage par les besoins ("features")
 - ♦ sprint défini par une liste de fonctionnalités
 - ♦ planification en nombre de fonctionnalités réalisées / à réaliser
 - itérations courtes

