

Gestion de Projet Informatique

Partie 2 : Cycles de vie de projet

Licence d'Informatique 3^e Année
Tianxiao Liu
Université de Cergy-Pontoise

*The earliest moment is
when you think it is late.*

Sommaire

- Objectif et définitions des termes utilisés
- Les modèles séquentiels: Cascade, V
- Les modèles agiles : ASD, DSDM, FDD, Crystal

Objectif d'un cycle de vie 1/2

- Définir **la démarche** à appliquer à un projet
- **Attention** : on parle ici des cycles de **développement** de projet, pas de notions marketing
- **Choix : 4 facteurs majeurs à prendre en compte**:
 - Minimiser les évolutions fonctionnelles en cours de projet
 - Soutenir la démarche d'assurance qualité (QA)
 - Maîtriser les coûts et les risques
 - Garantir sa conformité aux règles contractuelles et juridiques

Objectif d'un cycle de vie 2/2

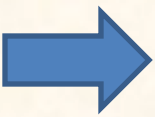
- D'autres éléments pouvant avoir une influence sur le choix de cycle de vie
 - La culture de l'entreprise
 - Les équipes de projet
 - Le niveau de connaissance des utilisateurs
 - Le contexte du projet
 - Les acteurs intervenant
 - Le type du projet

Définitions des termes

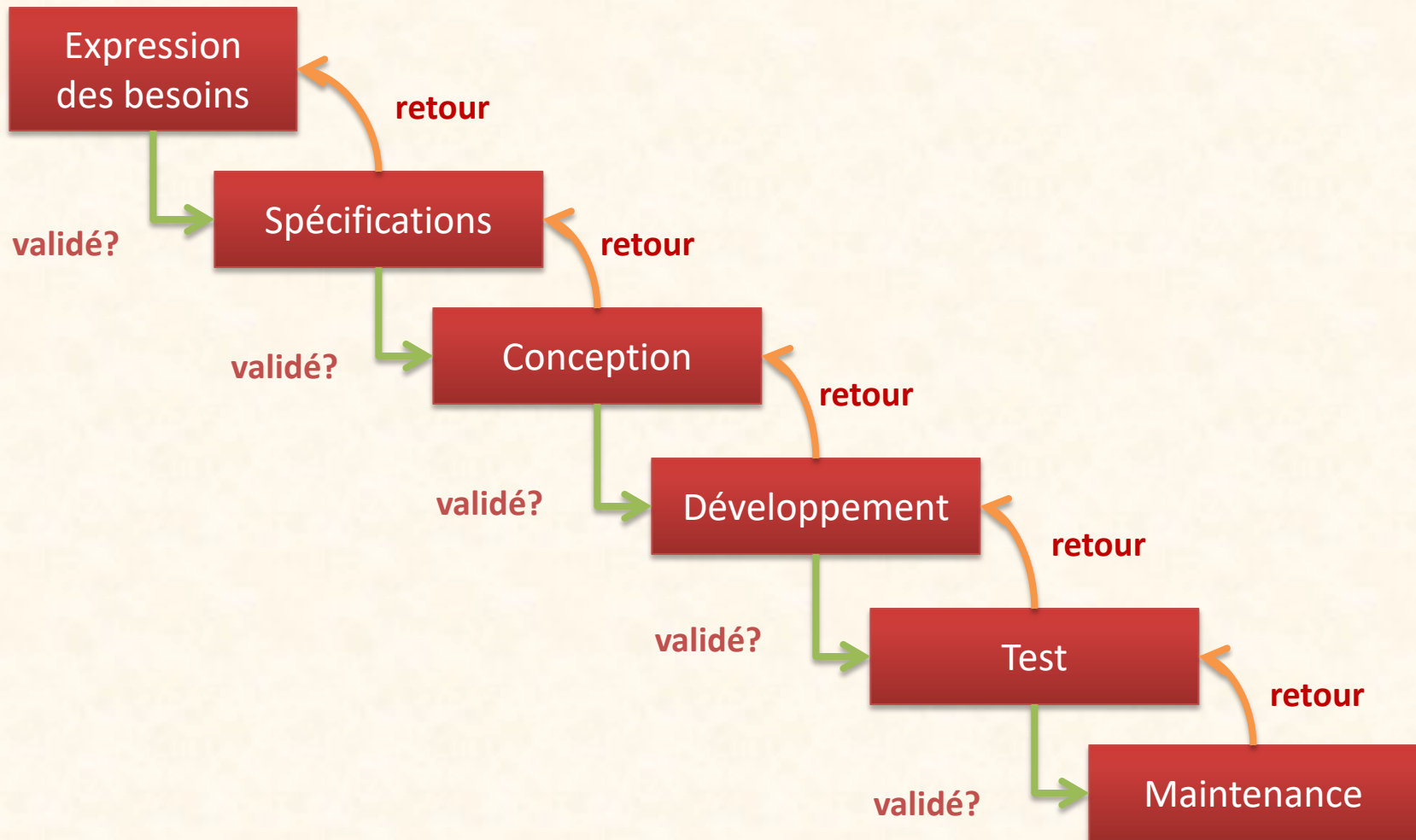
- **Prototype**

- L'embryon du produit initial : "**Je saurai ce que je veux quand je le verrai**"
- Viser à livrer rapidement une maquette de la solution à développer avec un minimum de fonctions **viables**
- Clarifier les besoins afin d'y arriver à une meilleure définition des spécifications fonctionnelles et techniques
- Eviter l'écart entre les besoins réels, ceux exprimés et ceux interprétés

Définitions des termes

- **Effet tunnel**
 - Point de départ : connu
 - Point d'arrivée : **inconnu**  **Rester dans le tunnel noir**
- Pour les clients
 - Pendant très longtemps, pas de communication avec les membres de l'équipe de projet
 - Grand risque d'avoir un résultat non satisfaisant !
- Pour les développeurs
 - De grandes phases de refactoring du code

Modèle en cascade



Modèle en cascade

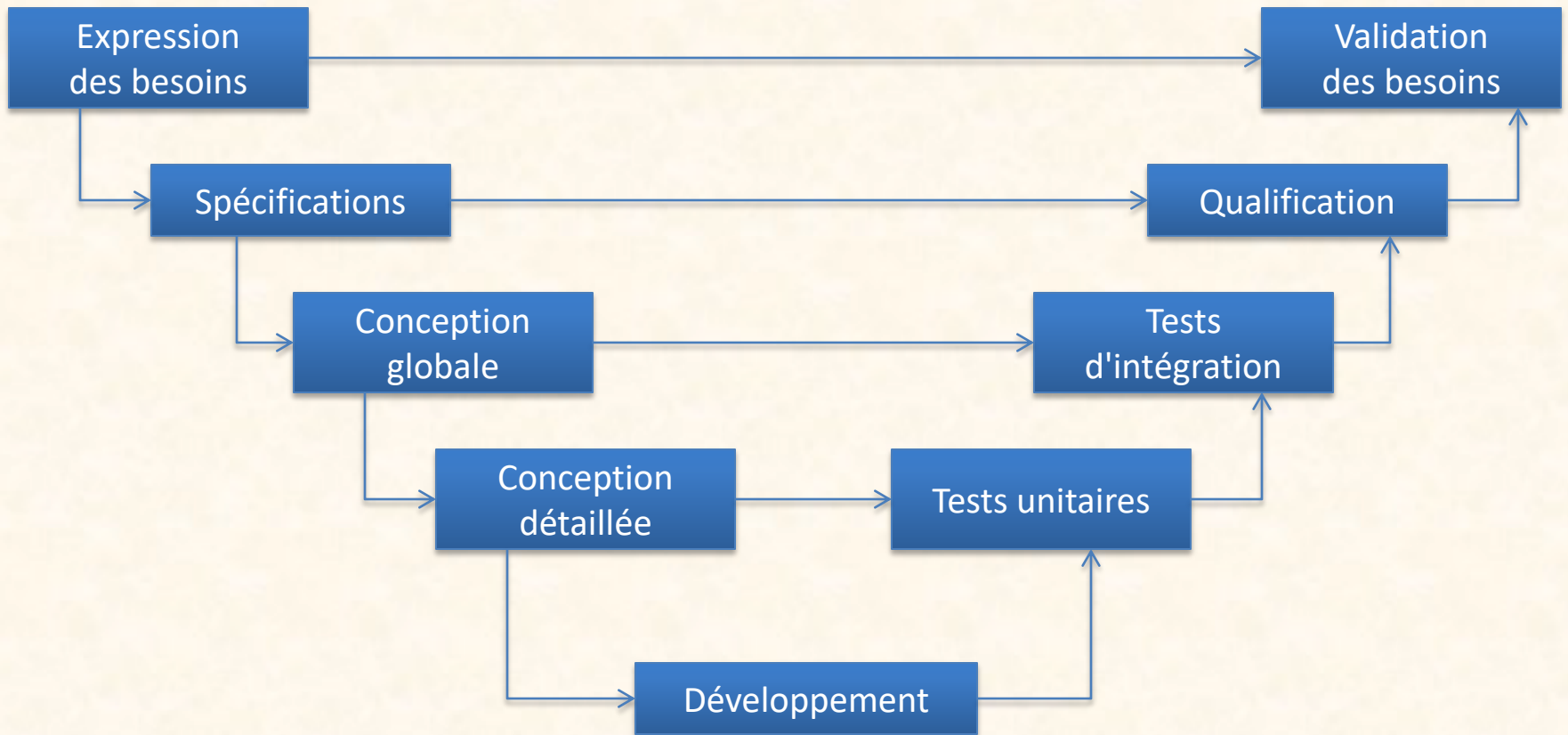
- **Principe**

- Une suite de phases dans un déroulement **linéaire**
- Validé → avancer, non validé → retourner

- **Avantages et inconvénients**

- Bonne réduction de risques en minimisant l'impact des incertitudes
- Bonne solution pour les projets peu complexes
- Problèmes non découverts avant les tests
- Pas de prise en compte des évolutions
- Difficulté d'amélioration des performances
- Durée du projet inférieure à **un an**

Modèle en V

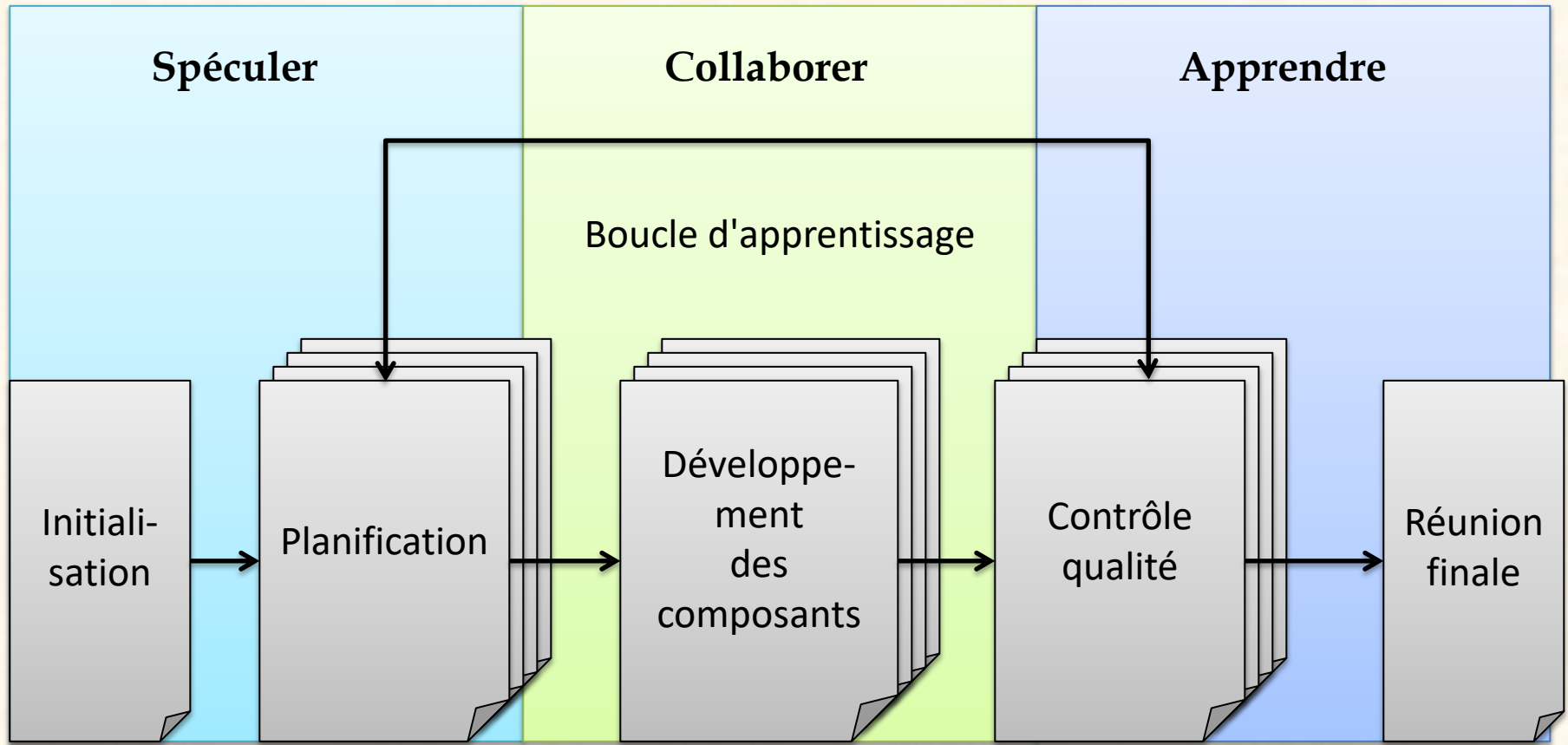


Modèle en V

- **Principe**
 - Modèle orienté tests
 - Décomposition et recomposition
- **Avantages et inconvénients**
 - Proposer au fur et à mesure une démarche de réduction des risques, en minimisant progressivement l'impact des incertitudes
 - Exclusion de l'utilisateur dès la phase de conception car trop technique
 - Contrôle qualité significatif seulement en fin de projet
 - Dans un contexte compétitif, risque éventuel de ne pas respecter les délais

Modèle agile : ASD

Adaptive Software Development



Modèle agile : ASD

- **Spéculation**
 - Initialisation (cahier des charges) : Objectifs, contraintes, intervenants, dates, estimation des charges, risques critiques
 - Détermination du nombre optimal d'itérations et de leurs contenus (composants) → Un **tableau de bord**
- **Collaboration**
 - Collaboration optimale entre les membres de l'équipe
 - Assurer **l'efficacité** de la production
- **Apprentissage**
 - Vue de la maîtrise d'ouvrage (clients)
 - Vue de l'équipe : points positifs et négatifs
 - Vue de progression du projet
 - **Où en est-il par rapport à ce qui était prévu?**

Modèle Agile : ASD

- **Caractéristiques**

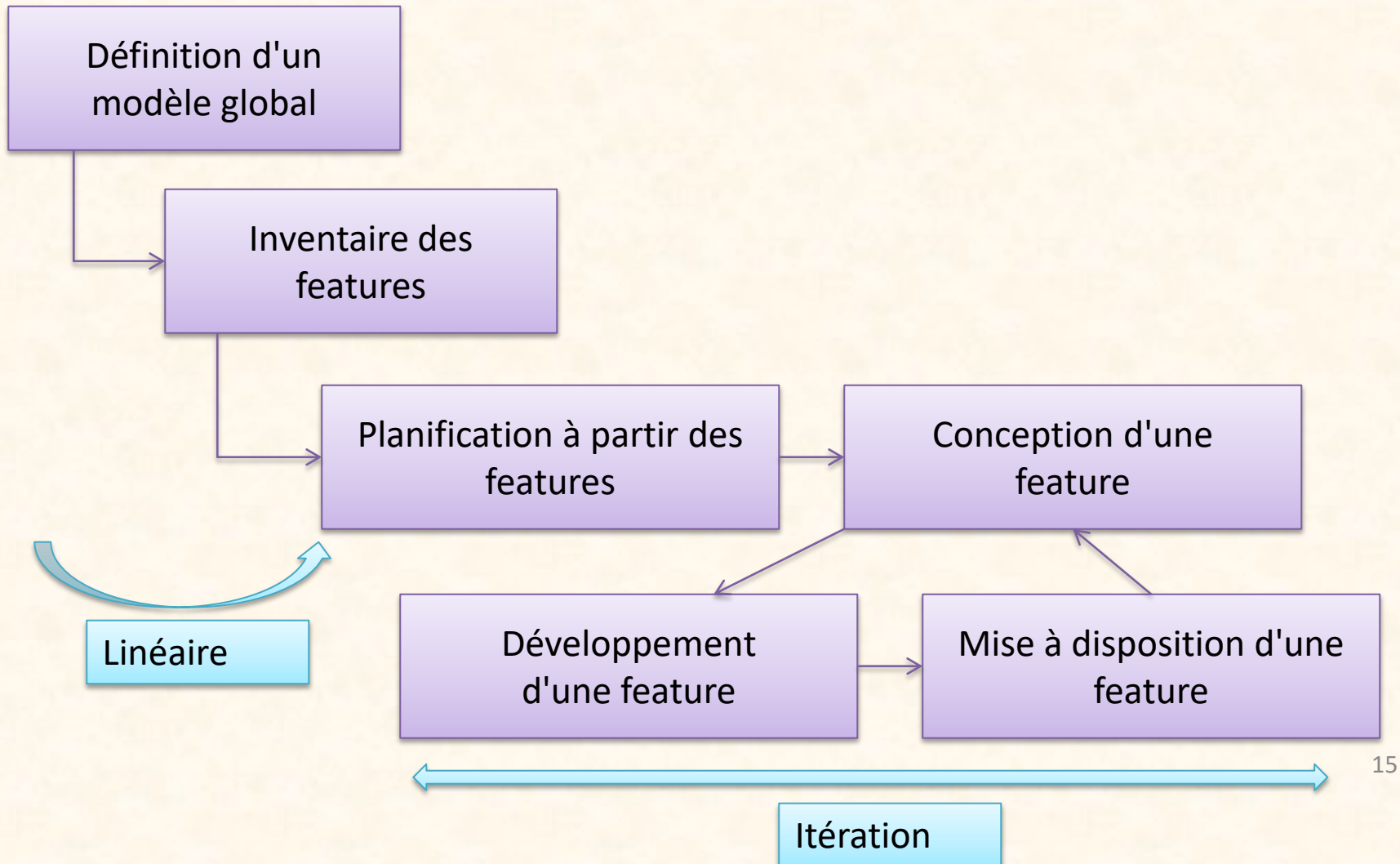
- S'adapte particulièrement aux projets **e-business**
 - Réalisation en des temps très courts
 - Support de nombreux changements et incertitudes
- Focalisation : viser les résultats plutôt que les tâches
- Itération : évolution en fonction des retours d'utilisateurs
- Changement : capacité à supporter un changement fonctionnel ou technique en cours de développement

- **Avantages / Inconvénients**

- Grande souplesse dans le changement
- Rapidité, respect des délais
- Implication du client
- **Gestion des risques : absence totale de stabilité !**

Modèle agile : FDD

Feature Driven Development



Modèle agile : FDD

- **Feature**
 - *Client-valued functionality*
- **Description des phases**
 - Développer un modèle global
 - Etablir une liste détaillée de *features* classées par priorité (inventaires de *features*)
 - Planifier à partir des *features*
 - Concevoir à partir des *feature*
 - Construire (développement) à partir des *features*
 - Mise à disposition des *features*

Modèle agile : FDD

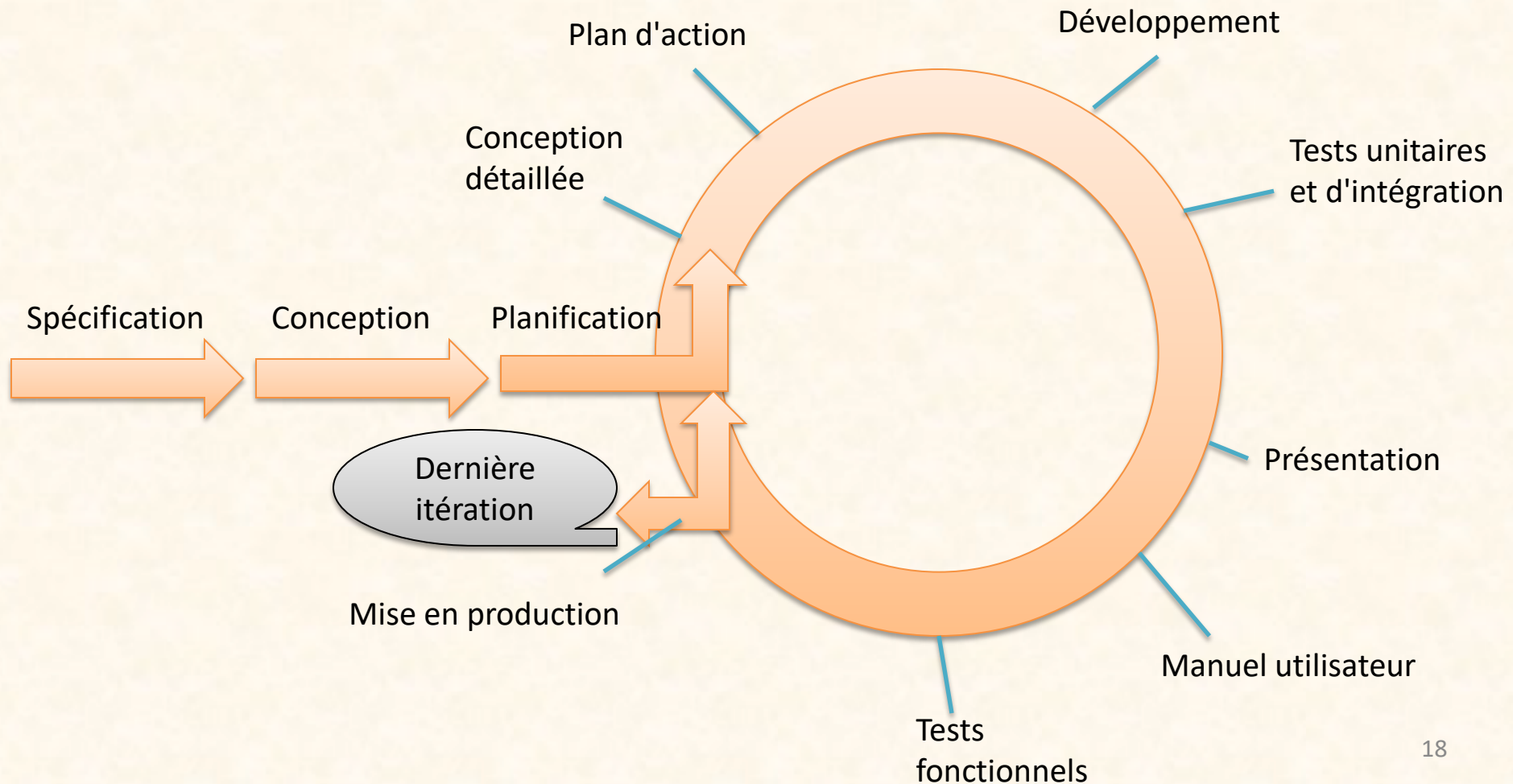
- **Principe**

- Mise en places des itérations très courtes
- Chaque itération produit un livrable fonctionnel
- Bases : caractéristiques de l'application (*features*)

- **Avantages / Inconvénients**

- Motiver les développeurs : livrable utilisable
- Sécuriser le chef de projet : état de l'avancement visible au gré des itérations
- Satisfaire les clients : livrables concrets, planning clair
- Maintenance de l'ensemble des composants peut parfois se muer en challenge quotidien → programmation par binôme (*pair-programming*)

Modèle agile : Crystal



Modèle agile : Crystal

- **Description des phases**
 - Spécification : classer et prioriser les besoins
 - Conception : choix des technologies et de l'architecture
 - Planification des fonctionnalités à réaliser
 - Itération de développement
 - Conception détaillée → plan d'action
 - Présentations périodiques
 - Eviter toute dérive et assurer la conformité avec les exigences des utilisateurs
 - **Documentation régulière** avant le lancement de chaque itération

Modèle agile : Crystal

- **Principe**

- Communication, promiscuité des développeurs et les rencontres avec les utilisateurs
→ améliorer la qualité des échanges
- Livraisons périodiques → fonctionnalités mises à disposition des utilisateurs
- Peu de normes, voire quasi inexistantes mais beaucoup de documents !

- **Avantages / Inconvénients**

- Adapté aux petits projets (moins de six personnes)
- Flexibilité, rapidité, livraisons fréquentes
- Inadéquation pour des projets plus importants

Les modèles de cycle de vie

- **Modèles séquentiels**
 - Cascade, Intégration, En V
- **Modèles itératifs**
 - Incrémental, Spiral de Boehm
- **Modèles agiles**
 - ASD, DSDM, FDD, Crystal, **Scrum, XP**, Lean
- D'autres modèles
 - RAD, UP

Conclusion

La production d'un projet informatique est :

- choisir un cycle de vie
- ne pas rester figé dans un modèle (**changer, adapter**)
- planifier (découpage, contrôles qualité, revues)
- autoriser une certaine **souplesse**
- prévoir : **ce qui n'est pas fait aujourd'hui risque d'être oublié demain...**
- assurer une forte communication entre tous les intervenants
- **adhérer à la méthodologie utilisée**
- faire preuve de bon sens

Activités de la séance

Revue du CdC et organisation programmation

- Chef + MOA
 - Revue du cahier des charges avec l'enseignant
 - **Cycle de vie choisi ou adapté**
- MOE et les autres
 - Organisation de travail de programmation
 - Répartition + intégration
 - **Présentation avec une slide**