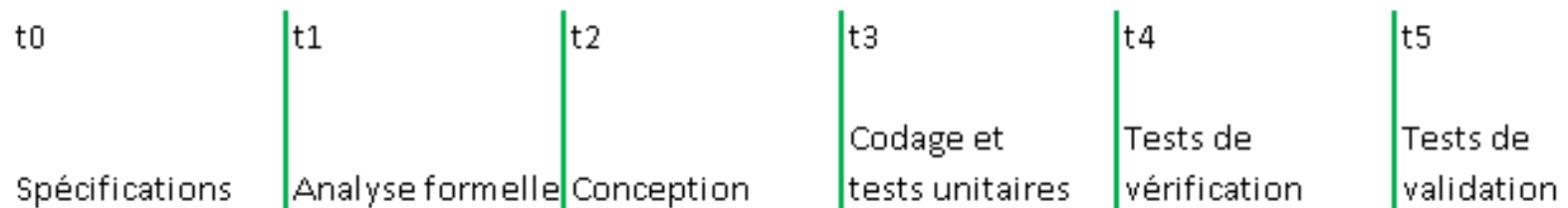


Cycles de vie classiques (suite)

Jusqu'à présent, théoriquement

- On insiste sur la découpe en activités qui se suivent séquentiellement :



Jusqu'à présent, théoriquement

En une étape

Tous
les
UC

Toute la
structure

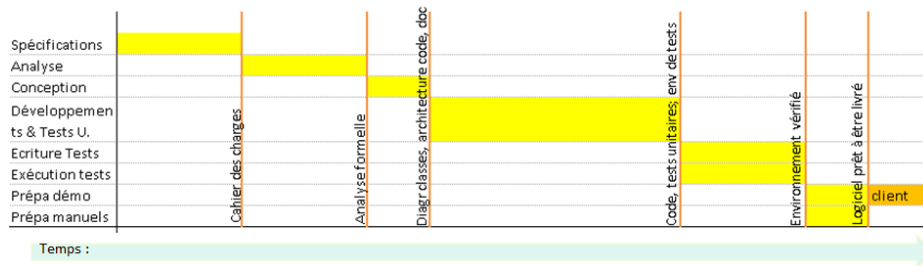
Tout le
développement



Points de synchronisation : attente qu'une étape soit terminée.

2 points de contact avec le client

Jusqu'à présent, théoriquement



Gestion des retards

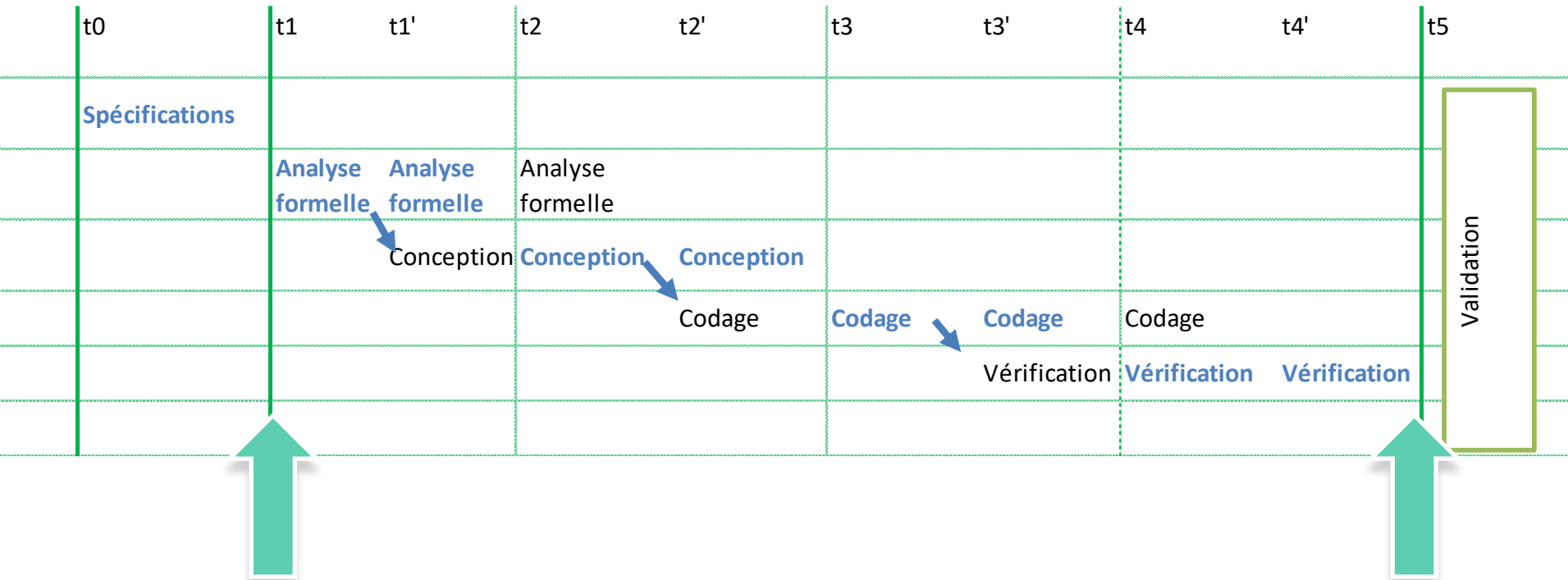
- (exemple) Que se passe-t-il si la conception prend du retard ?
- Imaginons un retard de livraison d'une unité de temps (exemple semaine), de t_6 à t_7 .

Toutes les étapes suivantes reculent d'une semaine ;

Ou, si possible, un concepteur vient renforcer l'équipe pour récupérer le retard de l'étape précédente dans celle-ci.

Que font les programmeurs pendant le temps t_6 ?

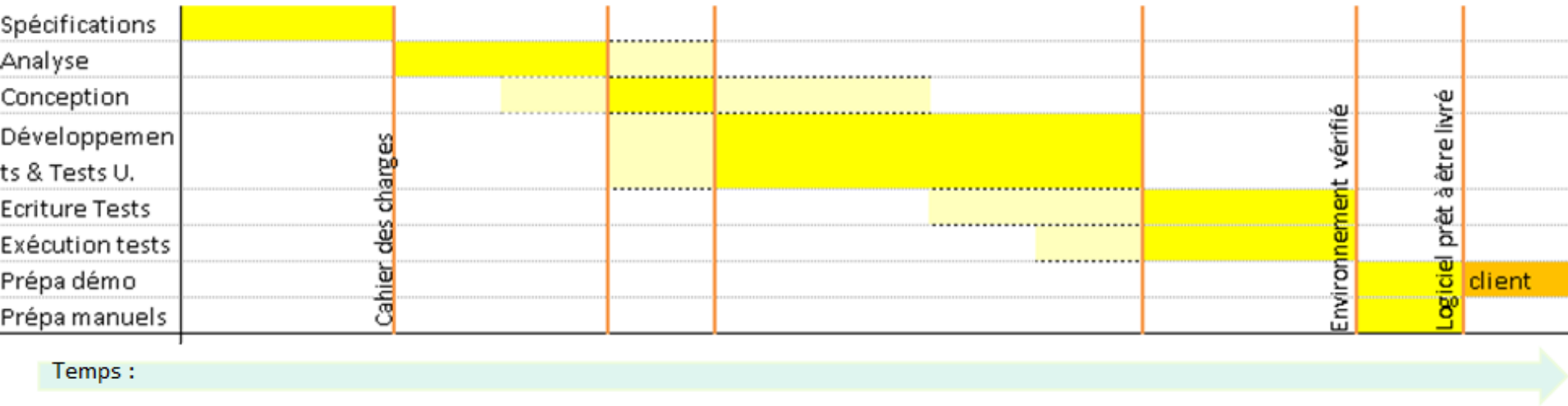
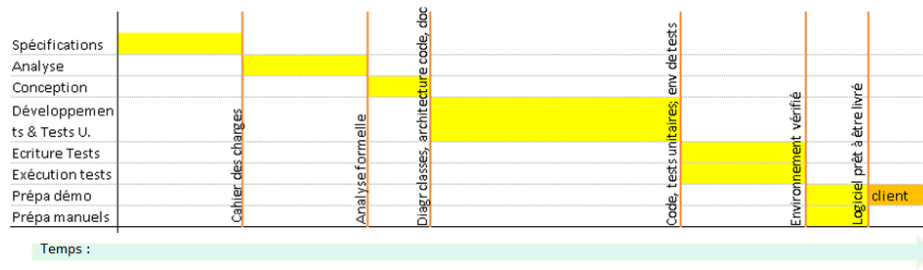
Pratiquement



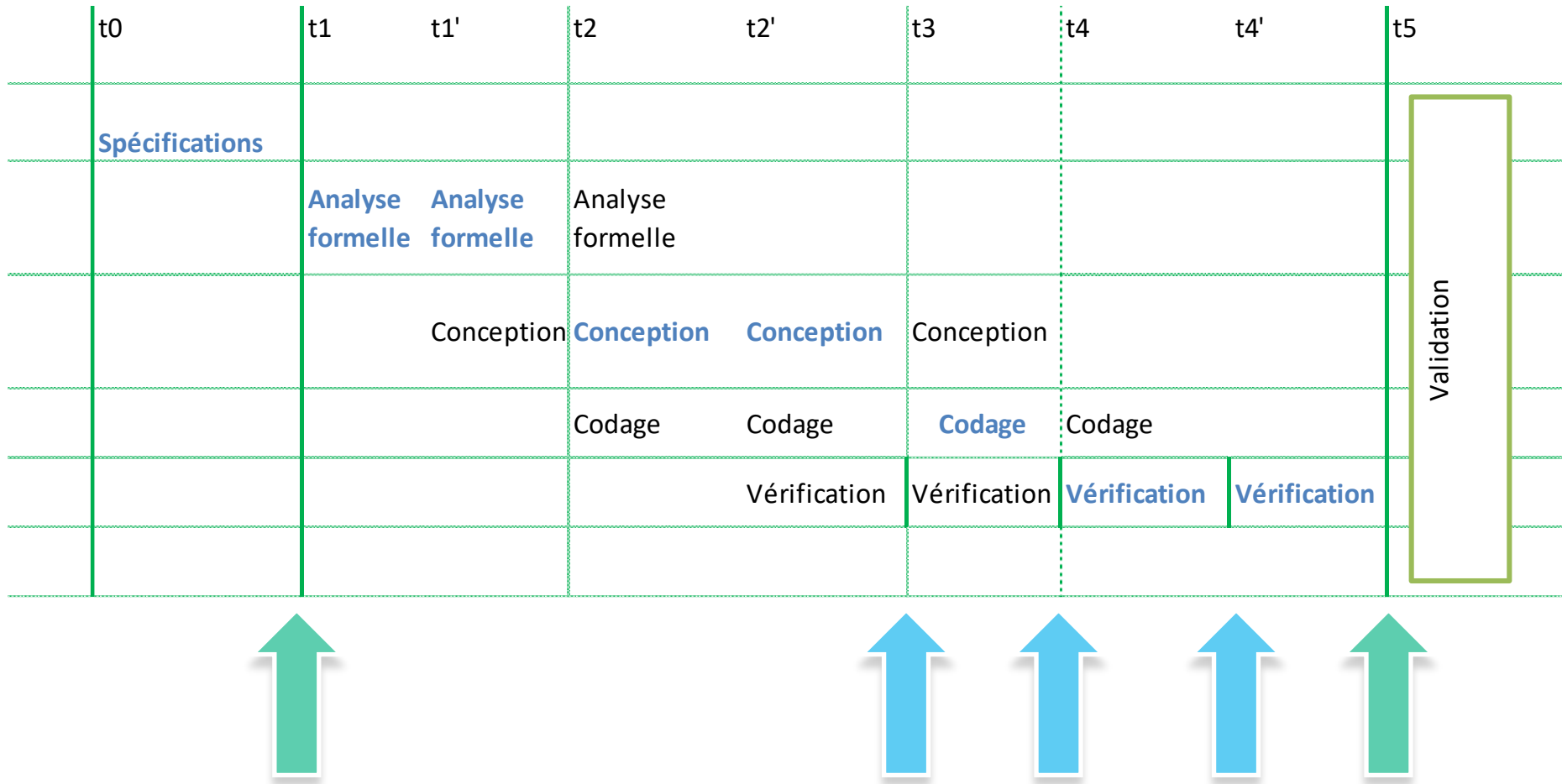
Toujours 2 points de contact avec le client

Mais à l'intérieur de ces points, « **élasticité** » et peu de réels points d'attente

Pratiquement



Un pas plus loin...



Parties de produit vérifiées plus tôt + feedback (testeurs -utilisateurs)
 Travail en parallèle possible

vers ...

les modèles à incréments

Processus Unifié (Unified Process)

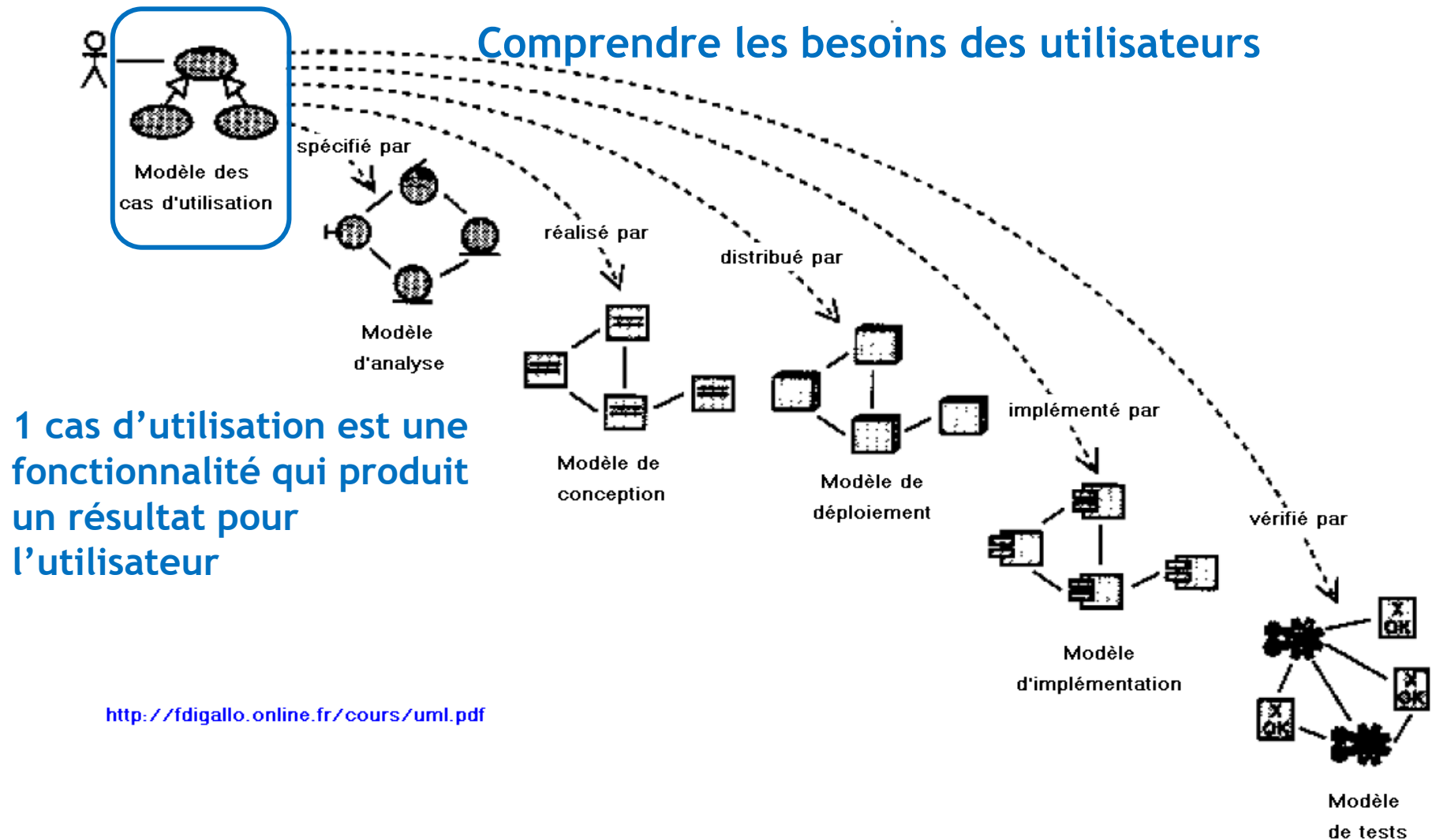
Unifié : pourquoi ?

- Années 90, une 50aine de méthodes orientées Objet
- Pas de consensus → recherches d'un langage commun :
 - **UML**
- UML = ensemble d'outils normalisés ; MAIS besoin d'une méthode
- Processus Unifié (**PU** - Unified Process **UP**) :
 - Méthode
 - Couverture complète du SDLC pour les développements orientés Objet
 - Lien avec UML.

PU méthode

- PU est **piloté par les cas d'utilisation**.
- PU est **centré sur l'architecture logicielle**.
- PU est à base de composants.
- PU est une méthode de développement de logiciels **itérative et incrémentale**.

PU piloté par les cas d'utilisation



<http://fdigallo.online.fr/cours/uml.pdf>

PU centré sur l'architecture

- Architecte dessine une image complète d'un bâtiment avant le début de la construction.

**→ Image complète
du système avant
son implémentation.**

PU itératif et incrémental

- L'idée de base :
 - Développer un système au travers de **cycles répétés (itération)** et en **petites avancées (incrément)**.
 - Chaque itération peut reprendre **plusieurs activités** (activités qui vont de l'analyse formelle jusqu'à la vérification).
 - Chaque incrément va ajouter de nouvelles fonctionnalités; c'est une construction morceau par morceau.
- Avantage majeur :
 - On peut tirer avantage de ce que l'on a appris durant l'itération précédente.
 - On réduit les risques.

PU réduit les risques

- Prendre en charge les risques importants **très tôt** dans le processus de développement.
- Définir une **architecture** qui guidera le développement logiciel.
- Fournir une **infrastructure** préfabriquée (framework) pour prendre en compte non seulement les exigences de base mais aussi les changements futurs.
- Développer **progressivement** le système, de façon incrémentale.

http://lgl.isnetne.ch/methodologie-2005/chap_06/chapitre6.pdf

03.2014

PU : 4 phases

- **La création** (inception) : la vision du projet est encore approximative. On y élaborera surtout les cas d'utilisation.
- **L'élaboration** : la vision y est plus élaborée. Le noyau du projet sera implémenté, les risques élevés résolus. La plupart des besoins seront identifiés.
- **La construction** : implémentation des éléments de risque et complexité plus faibles. Préparation du déploiement.
- **La transition** : B-tests et déploiement.

Phase 1 : création

- Développer la vision du projet
- Définir la portée du projet
- Réduire les risques majeurs
- S'assurer de la viabilité commerciale
- 1 seule phase - pas d'itération

Activités

- Comprendre les besoins du client
- Spécifier
- Analyser
- Concevoir
- Développer

Phase 2 : élaboration

- Développer l'architecture de référence
- Avoir compris l'essentiel des besoins
- Réduire les risques élevés (risques de moindre gravité qu'en phase de création)
- Peut avoir plusieurs itérations

Activités

- Comprendre les besoins du client
- Spécifier
- Analyser
- Concevoir
- Développer
- Tests

Phase 3 : construction

- Développer le système
- Réduire les risques
- Vérifier l'utilisabilité du produit
- Peut avoir plusieurs itérations

Activités

- Comprendre les besoins du client
- Spécifier
- Analyser
- Concevoir
- Développer
- Tests

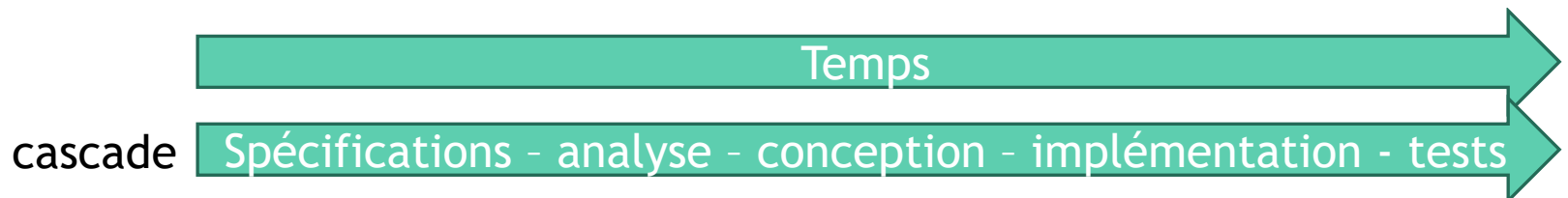
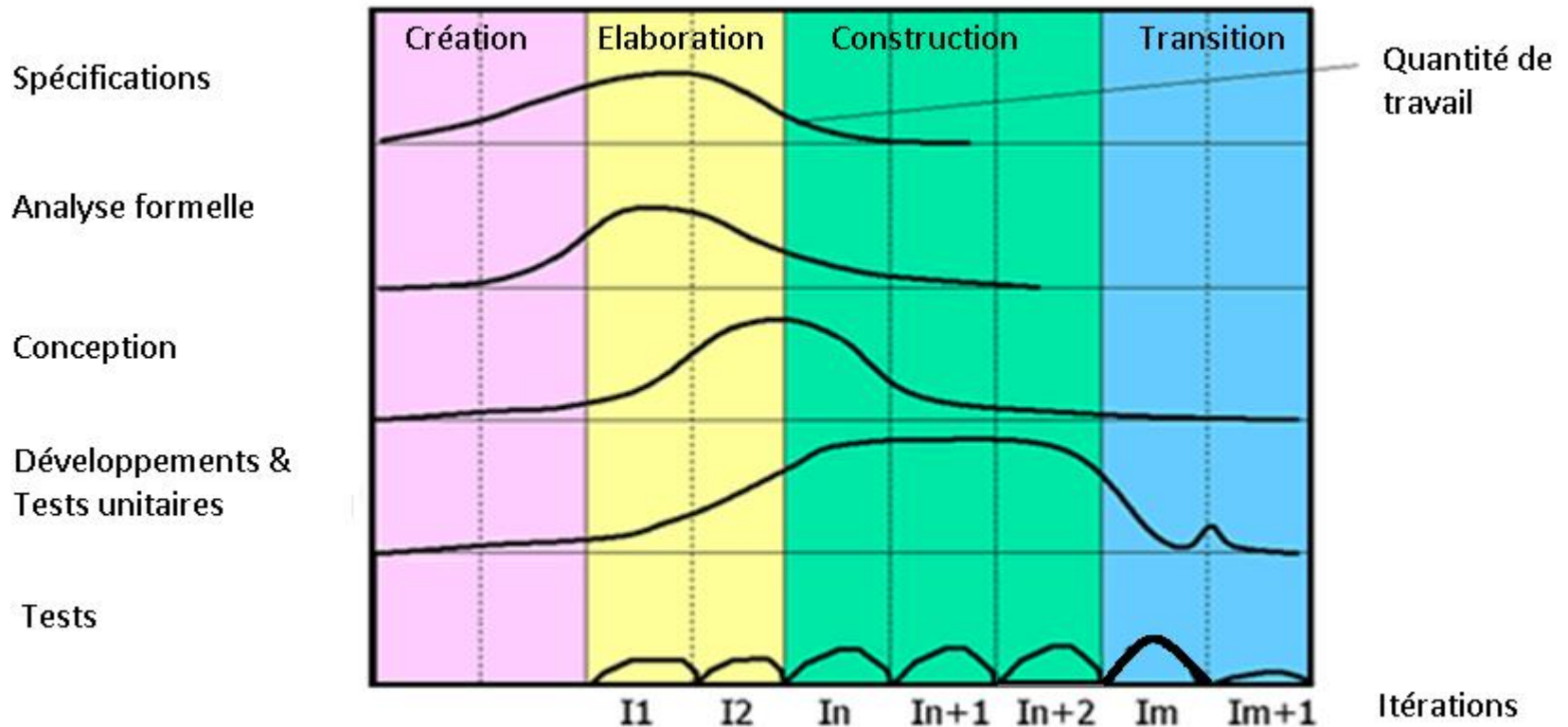
Phase 4 : transition

- S'assurer que le produit est livrable
- Déployer
- Former les utilisateurs
- Mettre en production
- Peut avoir plusieurs itérations

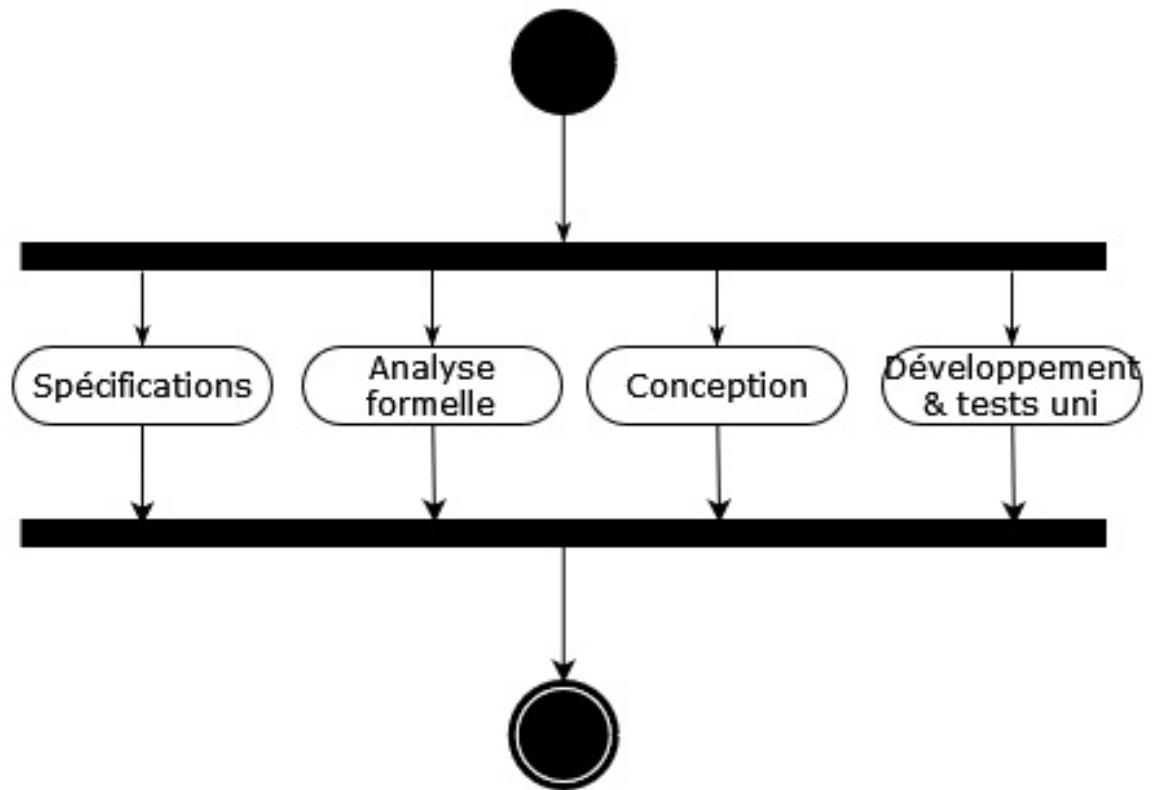
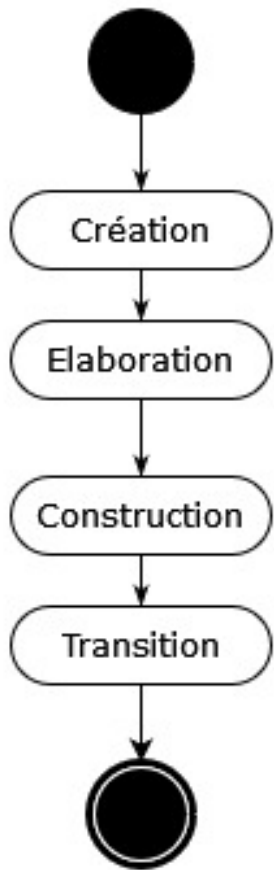
Activités

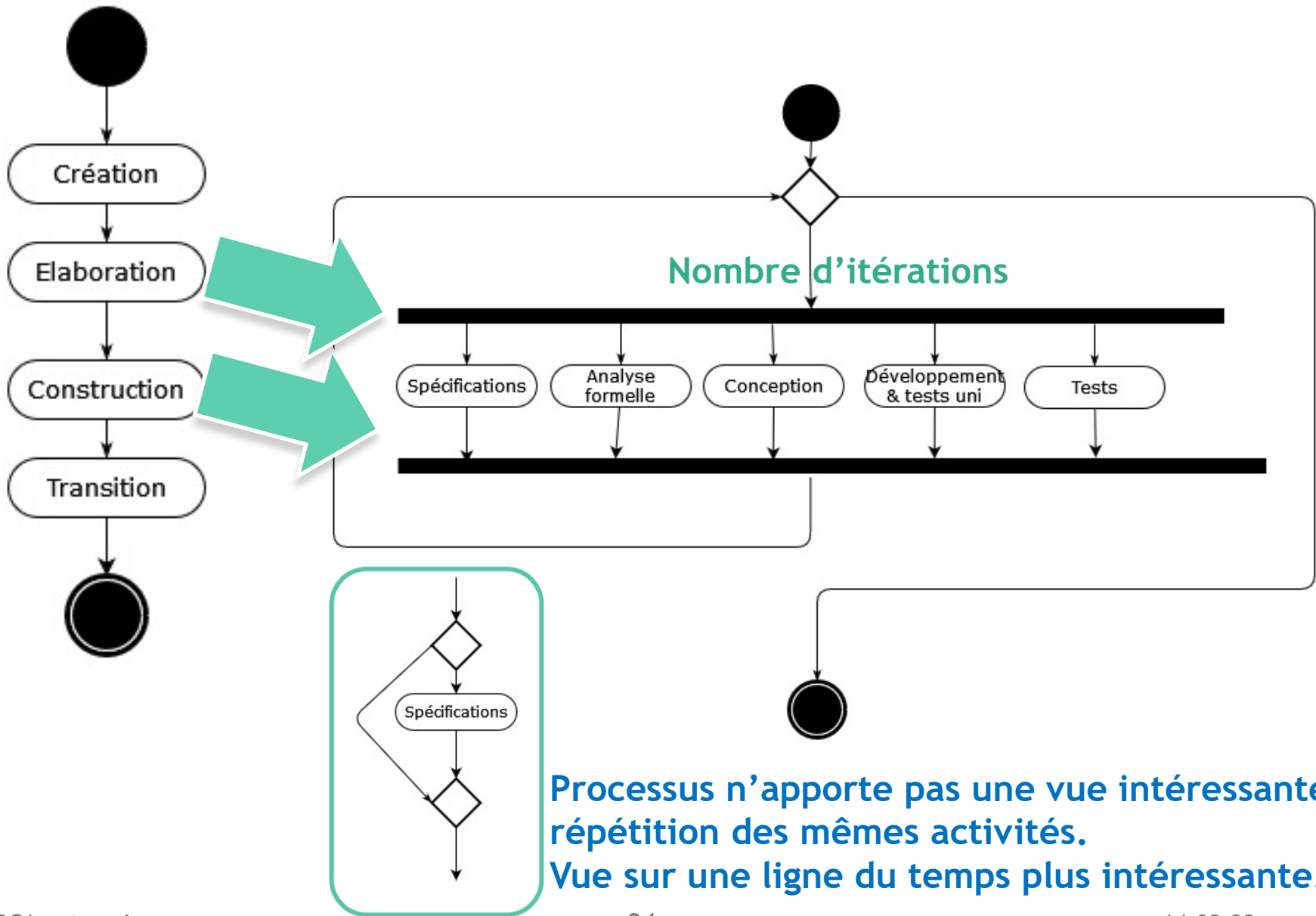
- Comprendre les besoins du client
- Spécifier
- Analyser
- Concevoir
- Développer
- Tests
- Déployer, livrer...

PU : cycle de vie



Processus ?





Conclusion

- Le PU met en évidence :
 - Le cycle de vie, avec des **phases** et des itérations.
 - Les **disciplines** qui ne sont pas limitées au temps d'une seule phase.
 - Le fait que les logiciels ne sont pas développés de la façon strictement séquentielle exprimée dans la Cascade ou le V...