

Cycles de vie classiques (2)

Projet AE : 1^{er} avis

Projet AE : 1er avis estudiantin

Cycle classique	Nombre de votes	Justifications
Cascade	1	Cahier des charges et conception non rejouable
Prototyping	0	
Modèle en Y	1	Technologie connue de suite <i>Avis prof : importance de l'architecture</i>
Modèle en V	1	Test prévus
Spirale	1	Contact avec le client; livrables réguliers <i>Avis prof : 2 validations par le client (les 2 démos)</i>

Spécifications

PAE : le SVYpirale

appellation de Benedetto La Monica

Analyse
formelle

Conception

Analyse
formelle

Codage

Codage

Conception

Analyse
formelle

Vérification

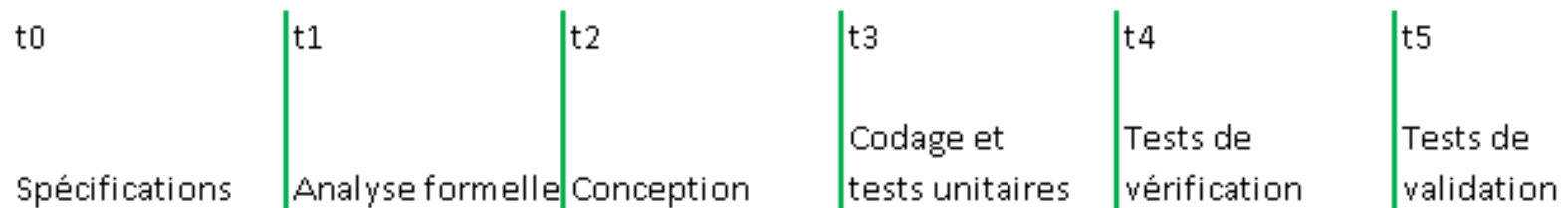
Codage

Validation

Cycles de vie classiques (suite)

Jusqu'à présent, théoriquement

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



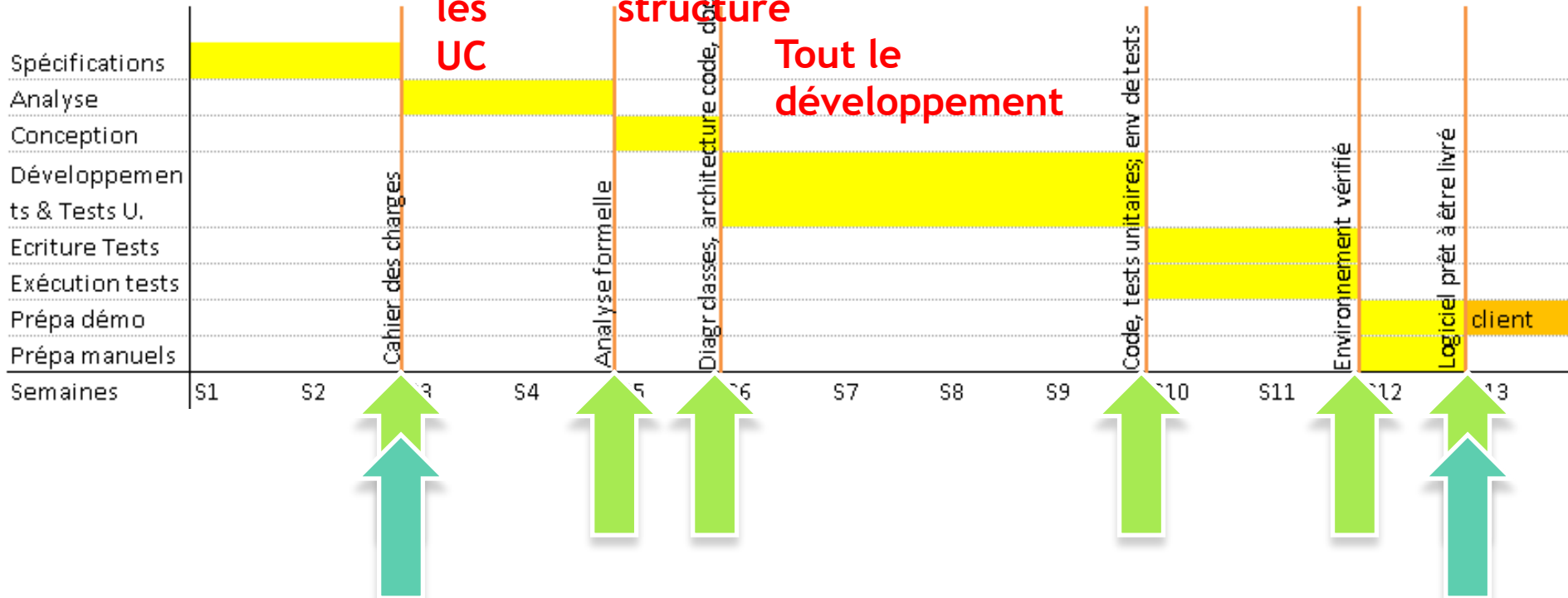
Jusqu'à présent, théoriquement

En une étape

Tous
les
UC

Toute la
structure

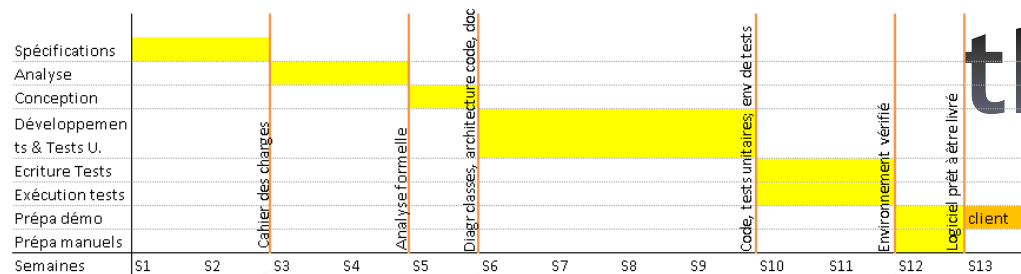
Tout le
développement



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

2 points de contact avec le client

Jusqu'à présent, théoriquement



Gestion des retards

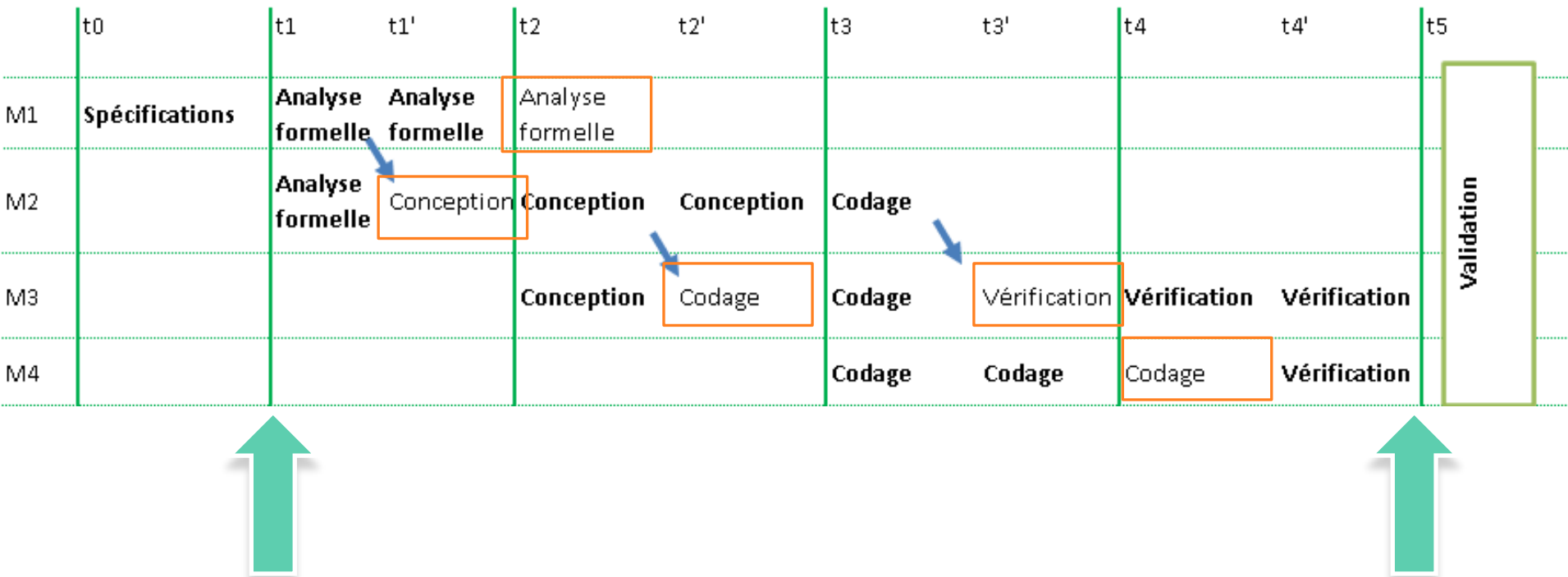
- (exemple) Que se passe-t-il si la conception prend du retard ?
- Livraison en semaine 7, plutôt qu'en semaine 6 ?

Toutes les étapes suivantes reculent d'une semaine ;

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

Que font les programmeurs pendant la semaine 6 ?

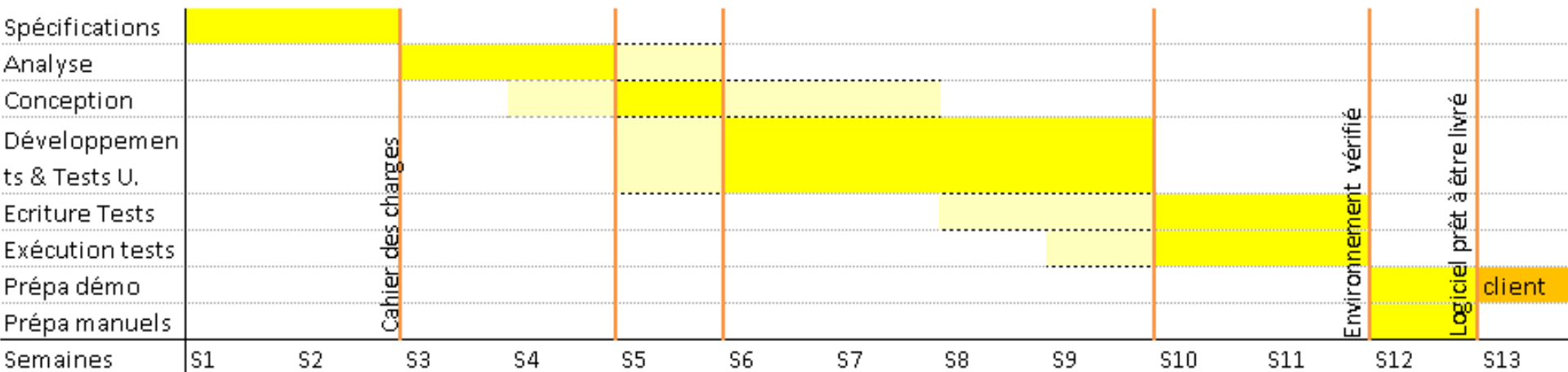
Pratiquement



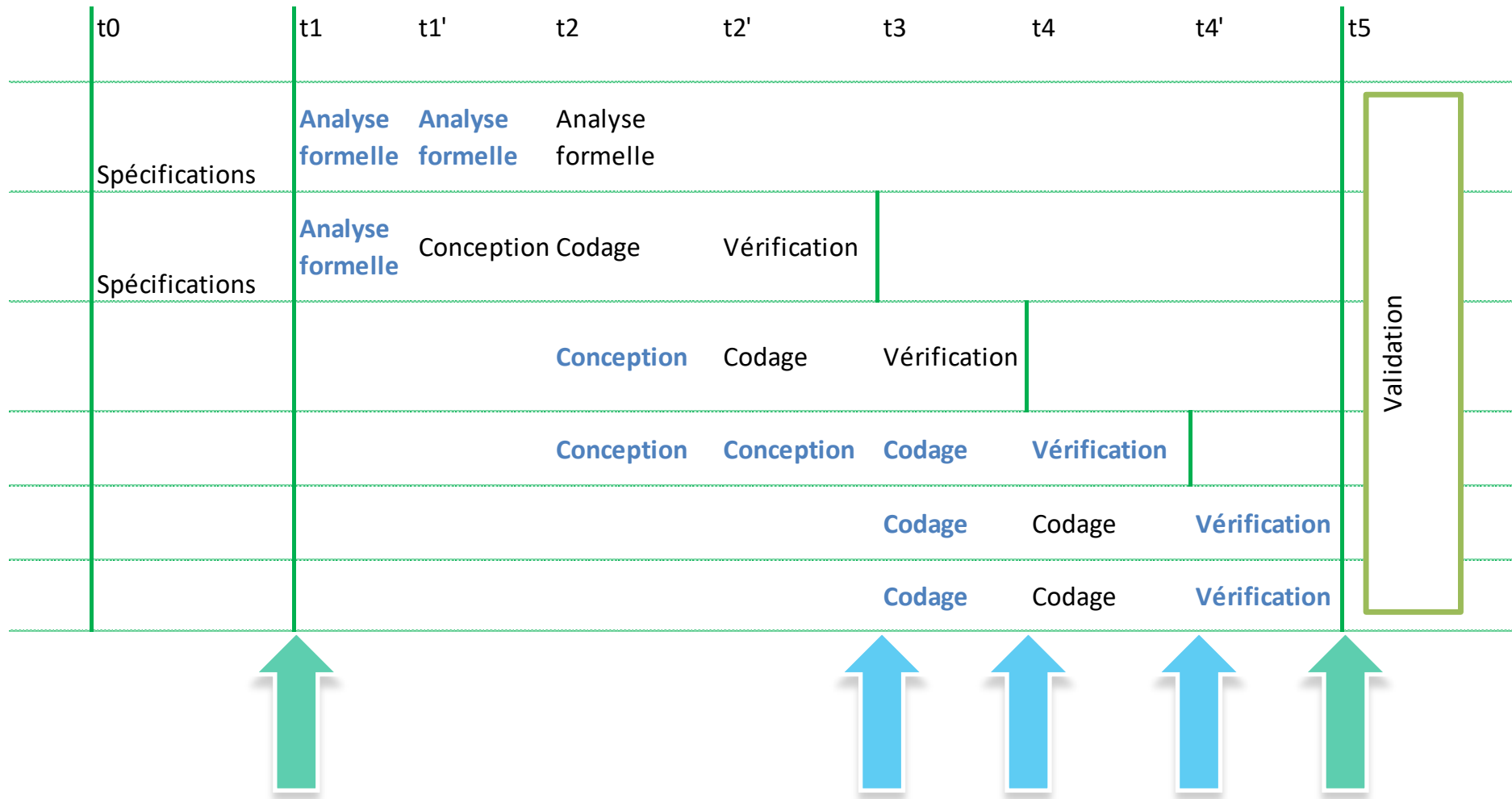
Toujours 2 points de contact avec le client

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)

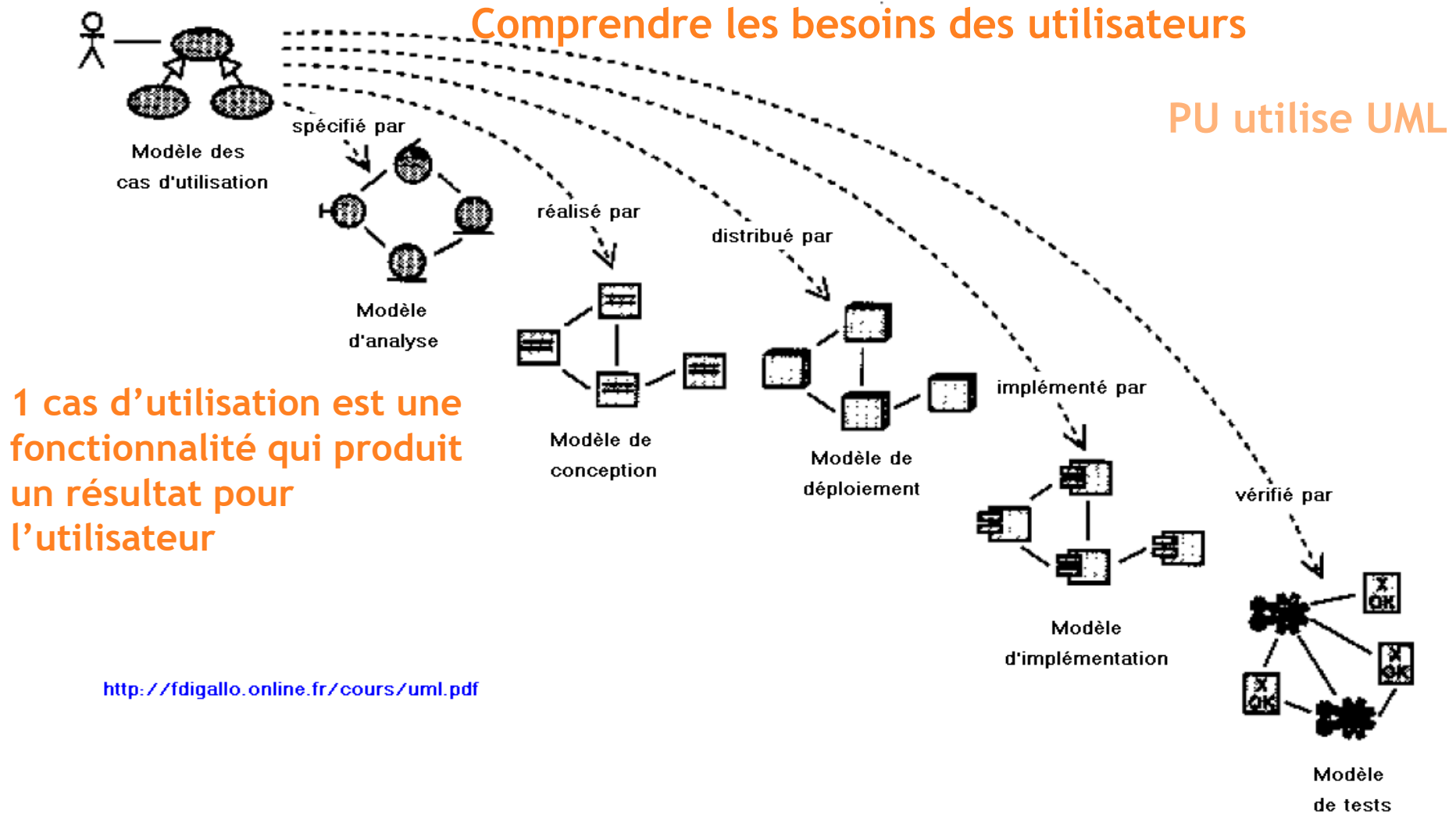
Unifié : pourquoi ?

- 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



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 des spécifications jusqu'à la vérification (/validation)).
 - 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

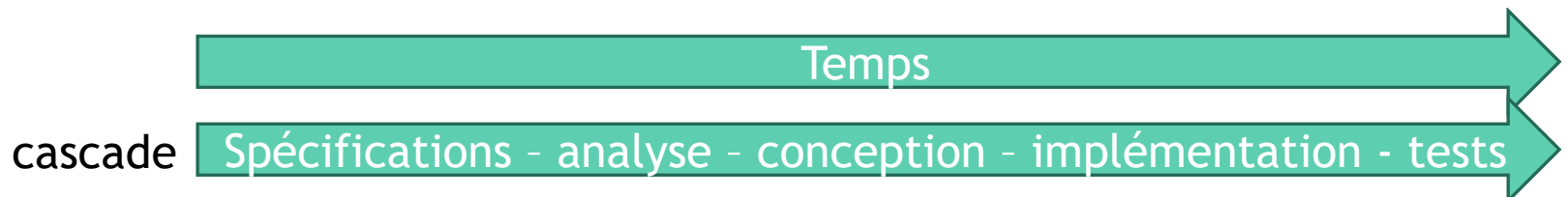
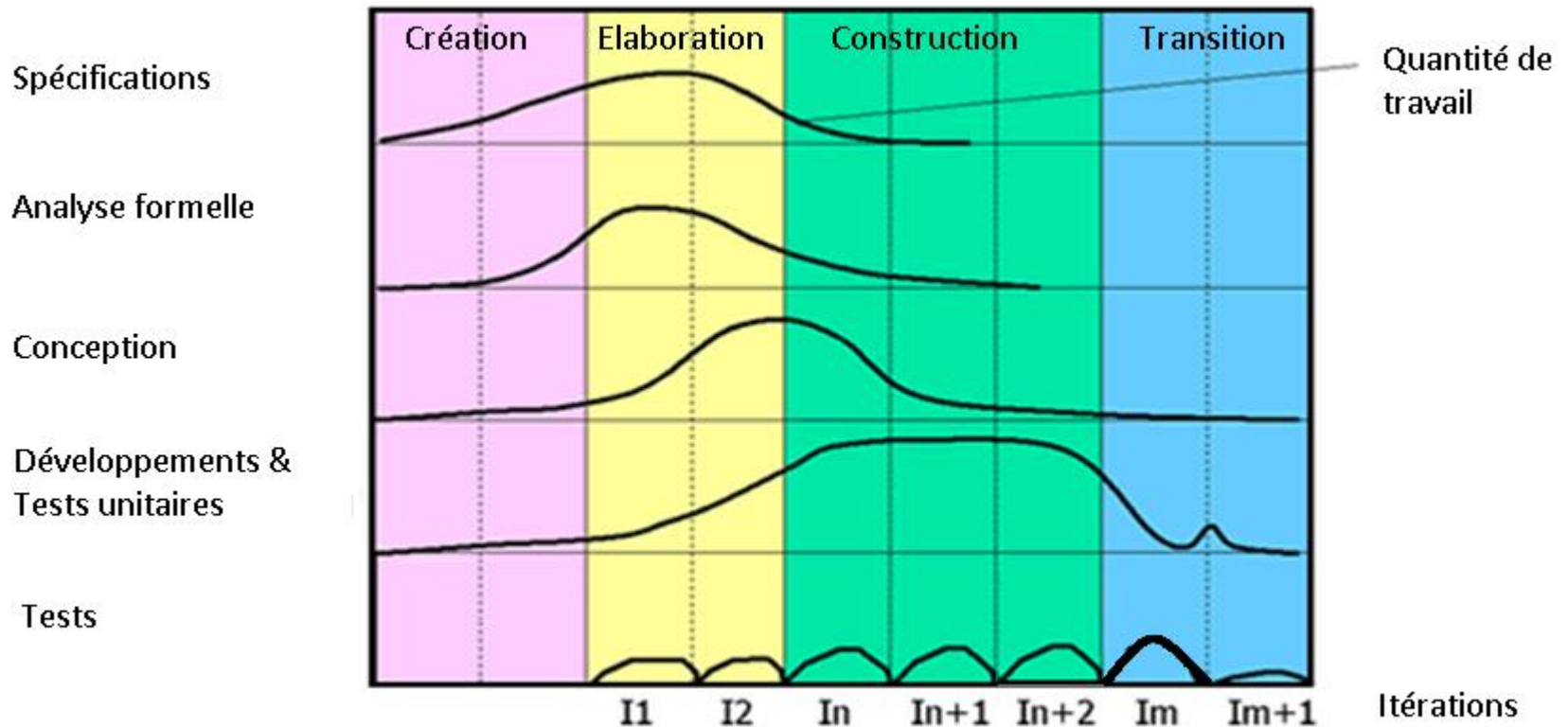
Phase 4 : transition

- 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



Exercices

Projet AE

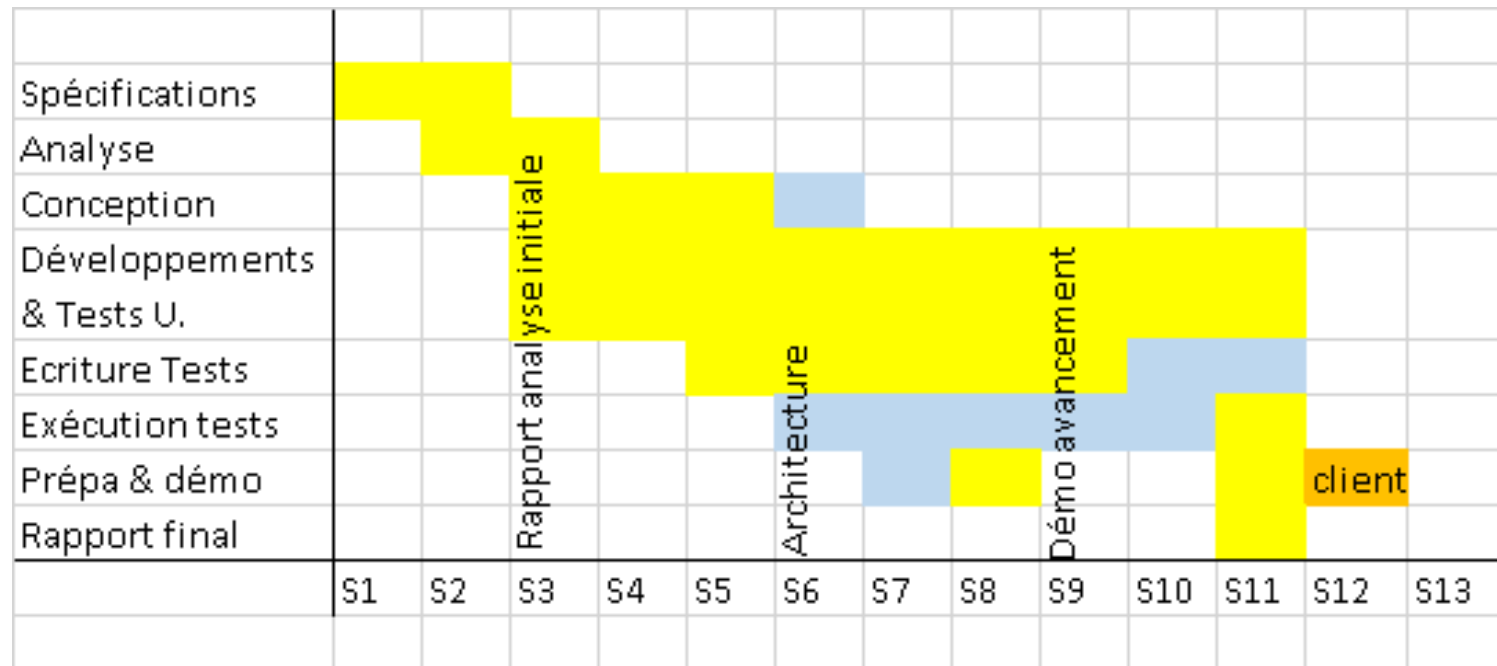
Y a-t-il un cycle de vie appelé « classique » qui décrit ce que vous avez fait en PAE ?

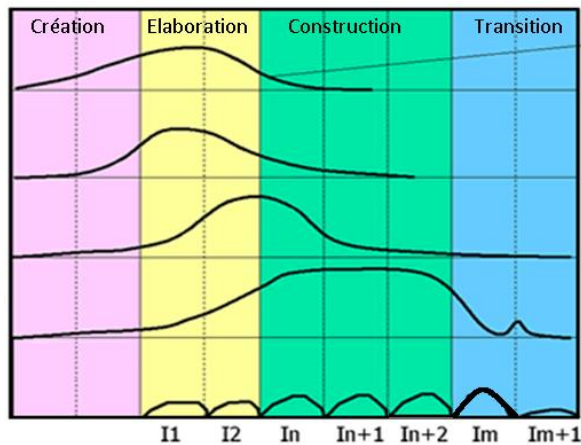
- Cascade
- Prototyping
- Y
- V
- Spirale
- SVYpirale
- PU ?

Livrable	Quand
1. Rapport d'analyse initiale	S3
2. Implémentation architecture : revue du code en séance	S6
3. Revue du code en séance	S9
4. Démo d'avancement en séance	S9
5. Livrable supprimé	S10
6. Code de tout le projet + Rapport + Démo	S12

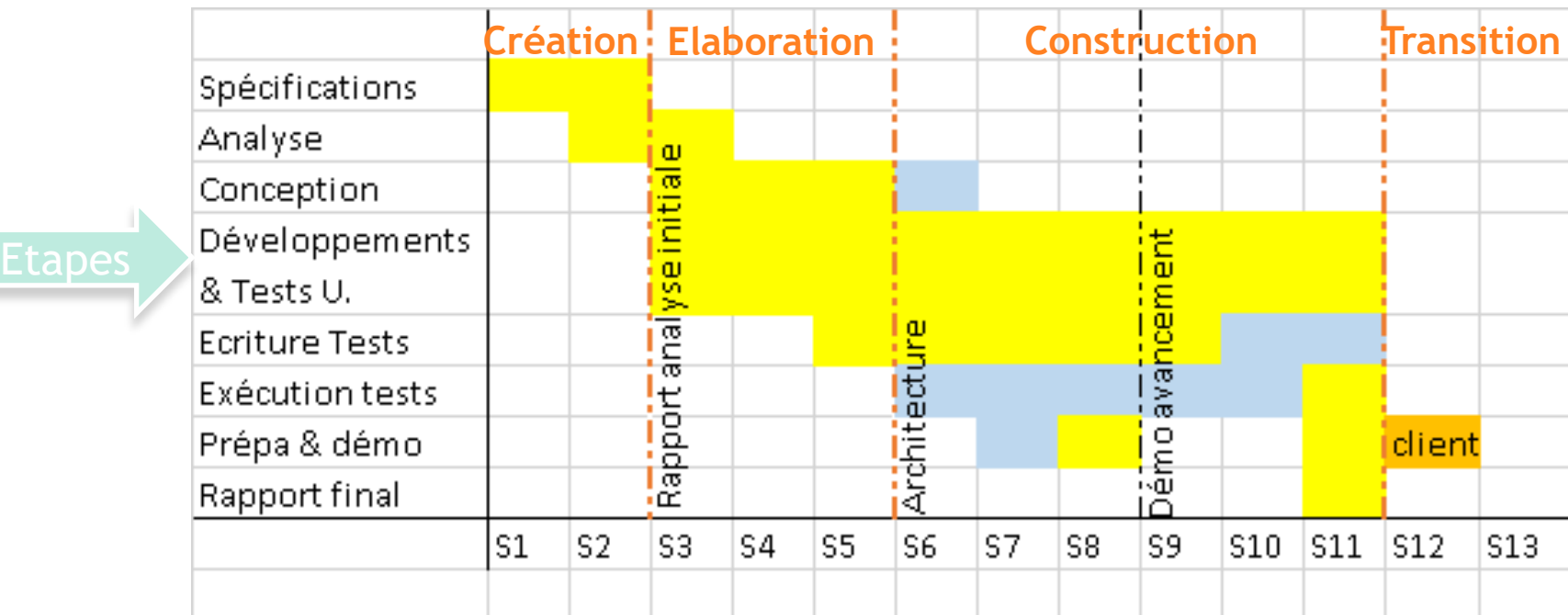
Livrable	Quand
1. Rapport d'analyse initiale	S3
2. Implémentation architecture : revue du code en séance	S6
3. Revue du code en séance	S9
4. Démo d'avancement en séance	S9
5. Livrable supprimé	S10
6. Code de tout le projet + Rapport + Démo	S12

Dessinez le cycle de vie employé, à votre avis, dans le cadre du cours de Projet AE.





Plaçons les 4 phases du PU dans le cadre du cours de Projet AE.



Modèle à incréments : suppression des points de synchronisation et des attentes entre étapes

