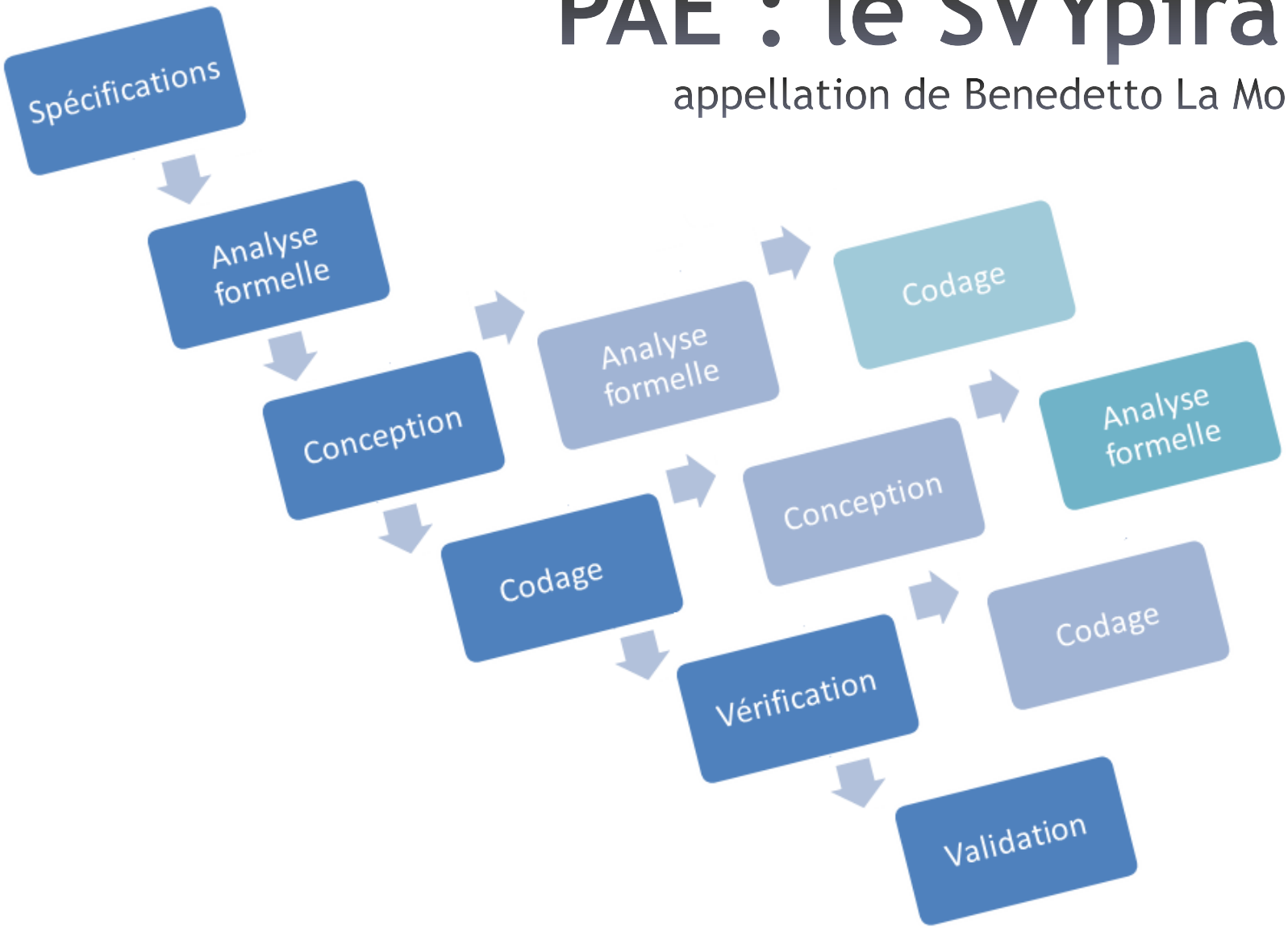
Cycles de vie classiques (2)

PAE : 1er avis estudiantin 2017

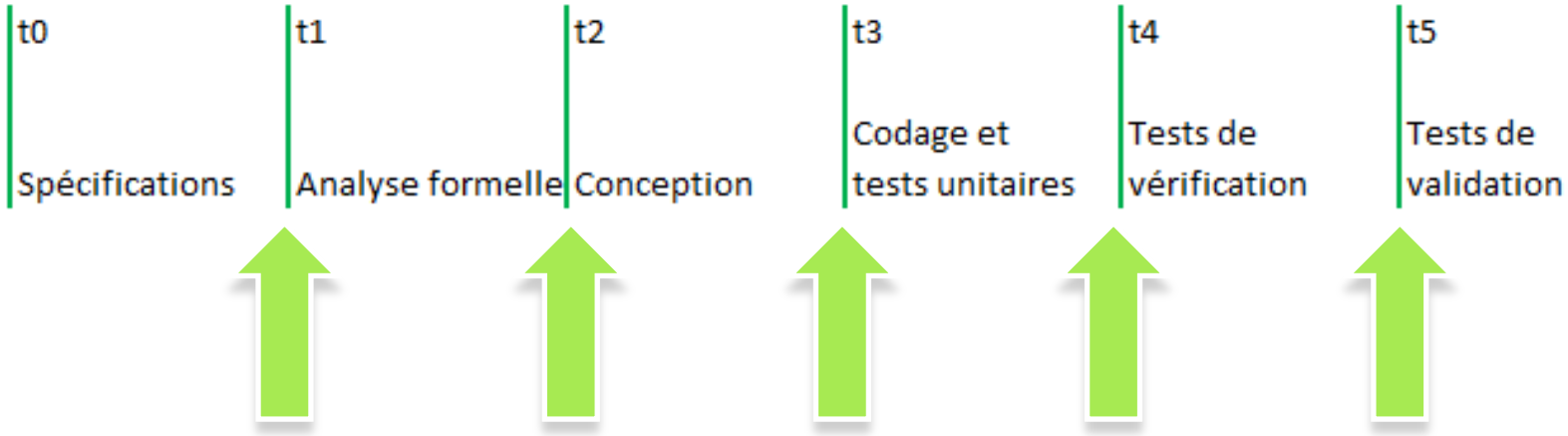
Cycle classique	Nombre de votes	Justifications
Cascade	7	Analyse en 1 fois, sans pouvoir y revenir; note non rejouable
Prototyping	0	
Modèle en Y	0	*Avis prof : importance de la conception*
Modèle en V	2	Test fonctionnels prévus
Spirale	13	Contact avec le client; livrables réguliers

PAE : le SVYpirale

appellation de Benedetto La Monica

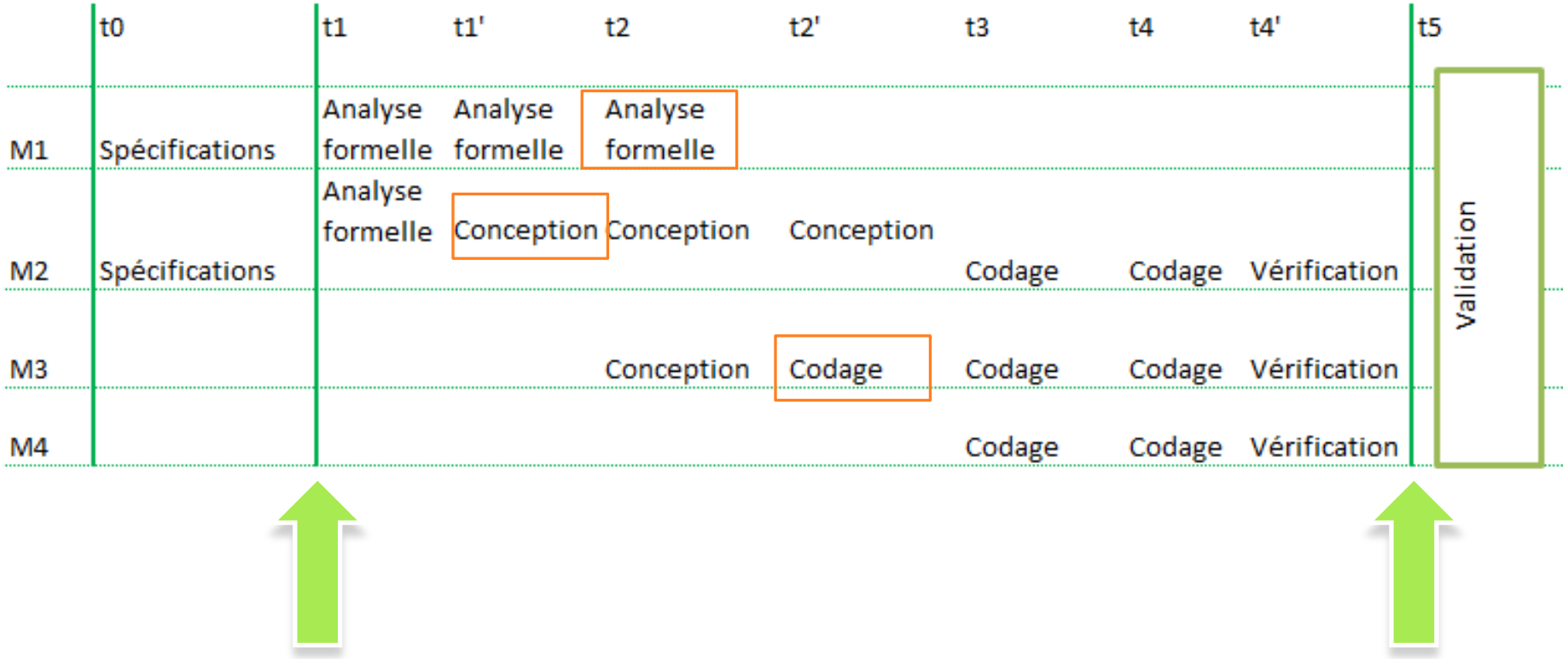


Jusqu'à présent, théoriquement



« Points de synchronisation » : « attente » qu'une étape soit terminée

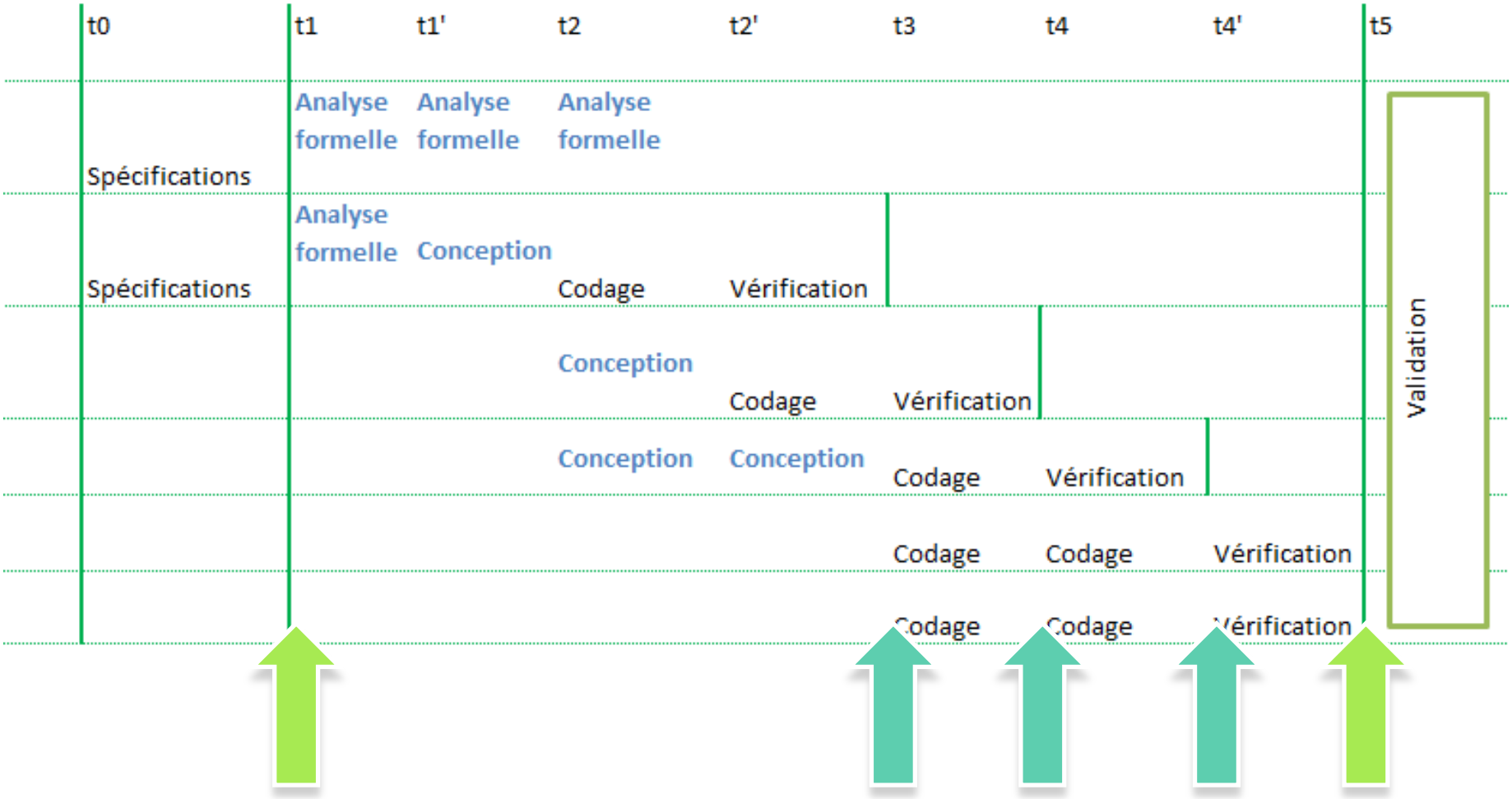
Pratiquement



2 points de contact avec le client

A l'intérieur de ces points, pas ou peu d'interactions avec le client

Un pas plus loin…



Parties de produit livrées plus tôt + feedback utilisateur
Travail en parallèle possible

vers …

les modèles à incréments

Processus Unifié
(Unified Process)

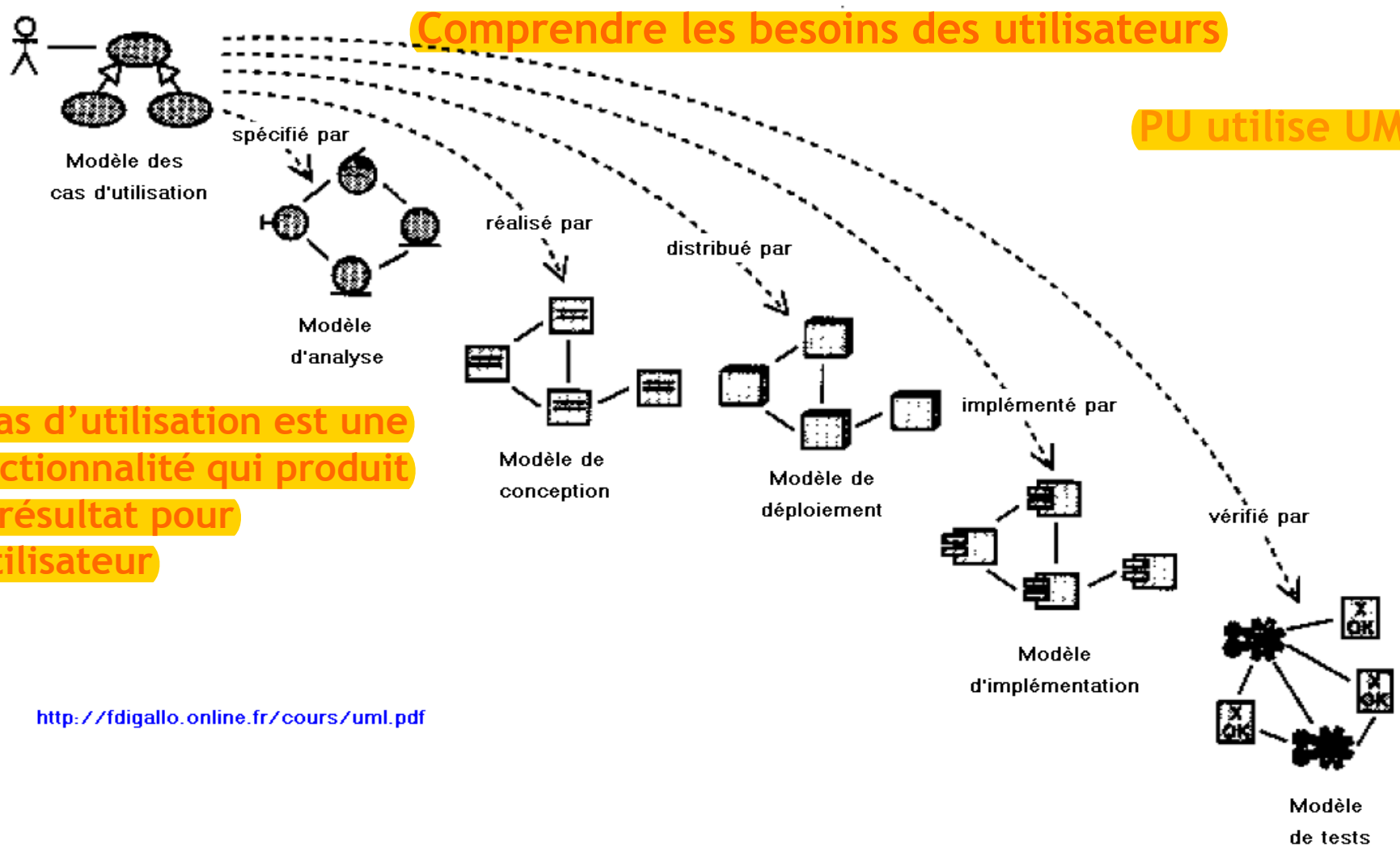
Unifié : pourquoi ?

- 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



1 cas d'utilisation est une fonctionnalité qui produit un résultat pour l'utilisateur

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

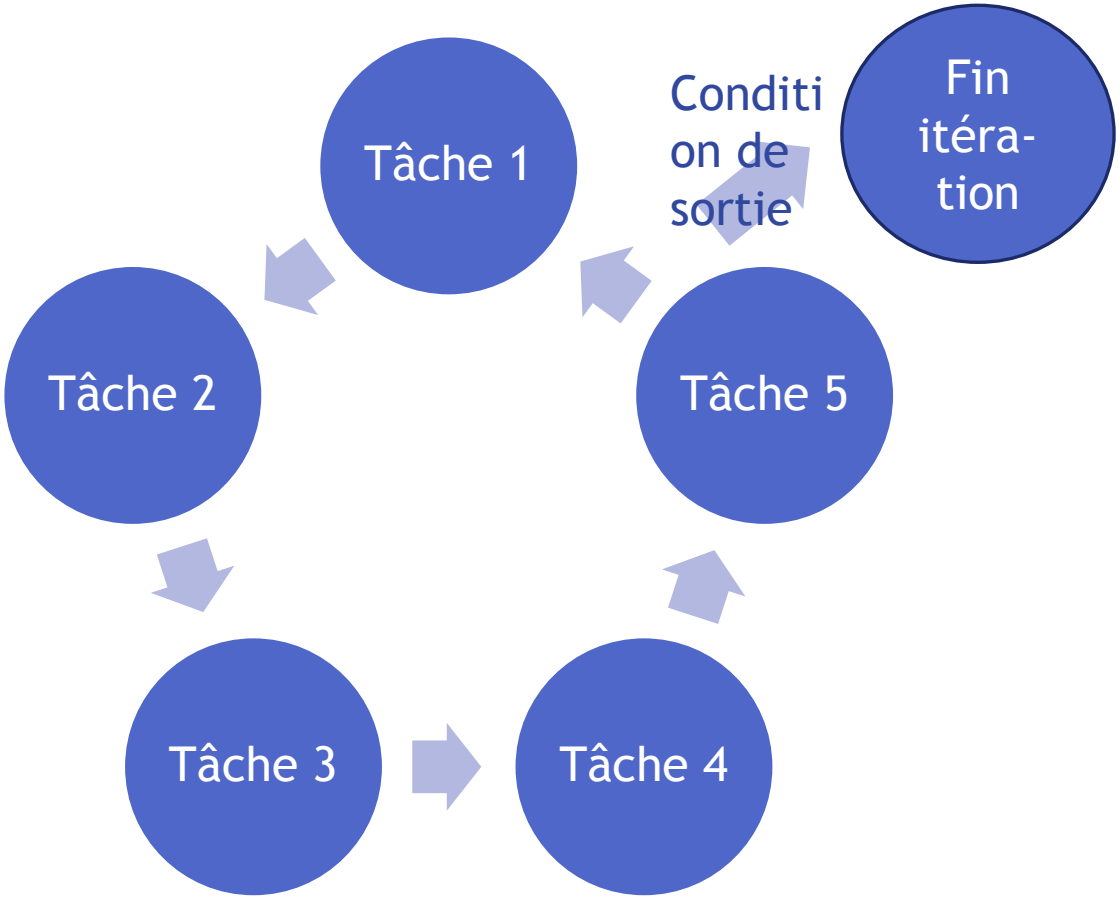
- L'idée de base :
  - Développer un système au travers de cycles répétés (itération) et en petites avancées (incrément)
  - Chaque itération peut reprendre plusieurs activités (activités qui vont des spécifications jusqu'à la vérification (/validation))
  - Chaque incrément va ajouter de nouvelles fonctionnalités.
- Avantage majeur :
  - On peut tirer avantage de ce que l'on a appris durant l'itération précédente
  - Réduction des risques

PU itératif et incrémental

- Réduire les risques :
  - Prendre en charge très tôt dans le processus de développement, les risques importants
  - 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

Itératif


Tâche 1
Tâche 2
Tâche 3
Tâche 4
Tâche 5
Condition de sortie
Fin itération

Itératif : Pros

- **Gestion précoce des risques** : limitation des coûts, en termes de risques, à une itération.
- **Progrès visibles rapidement**
- **Tests et intégration** se font de manière « continue »
- **Avancées** évaluées au fur et à mesure de l'implémentation
- **Gestion de la complexité** et rythme de développement soutenu grâce à des objectifs clairs et à CT
- **Feed-back rapide des utilisateurs** – besoin des utilisateurs se dév. au cours des itérations successives

Itératif : Cons

- **Définition de l'itération** : demande du temps, risqué
- **Lourd** à mettre en œuvre

Inadéquat pour les petits projets

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 : ß-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

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

Phase 3 : construction

- Développer le système

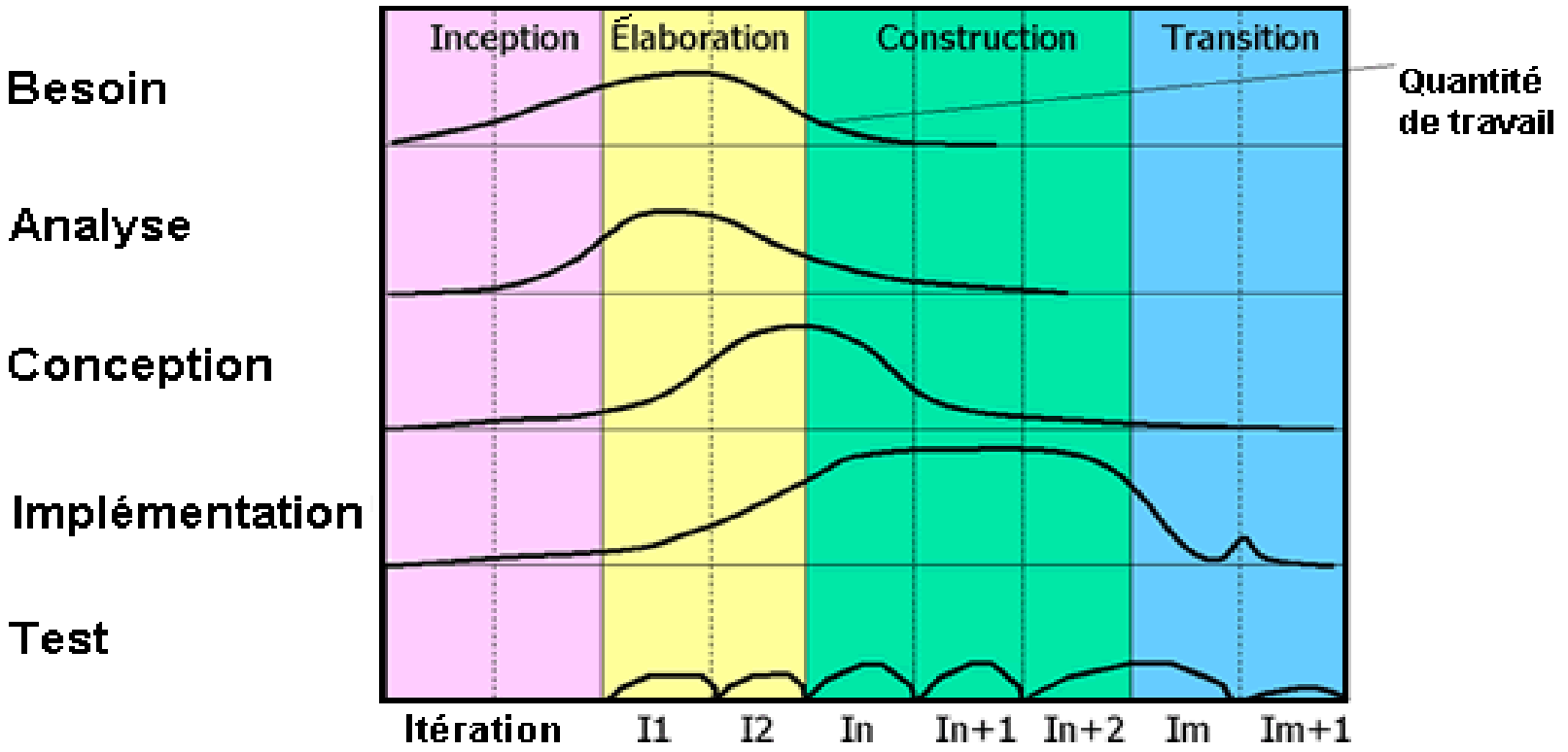
- Réduire les risques
- Vérifier l'utilisabilité du produit
- Peut avoir plusieurs itérations

Phase 4 : transition

- S’assurer que le produit est livrable

- Déployer
- Former les utilisateurs
- Mettre en production
- Peut avoir plusieurs itérations

PU : cycle de vie

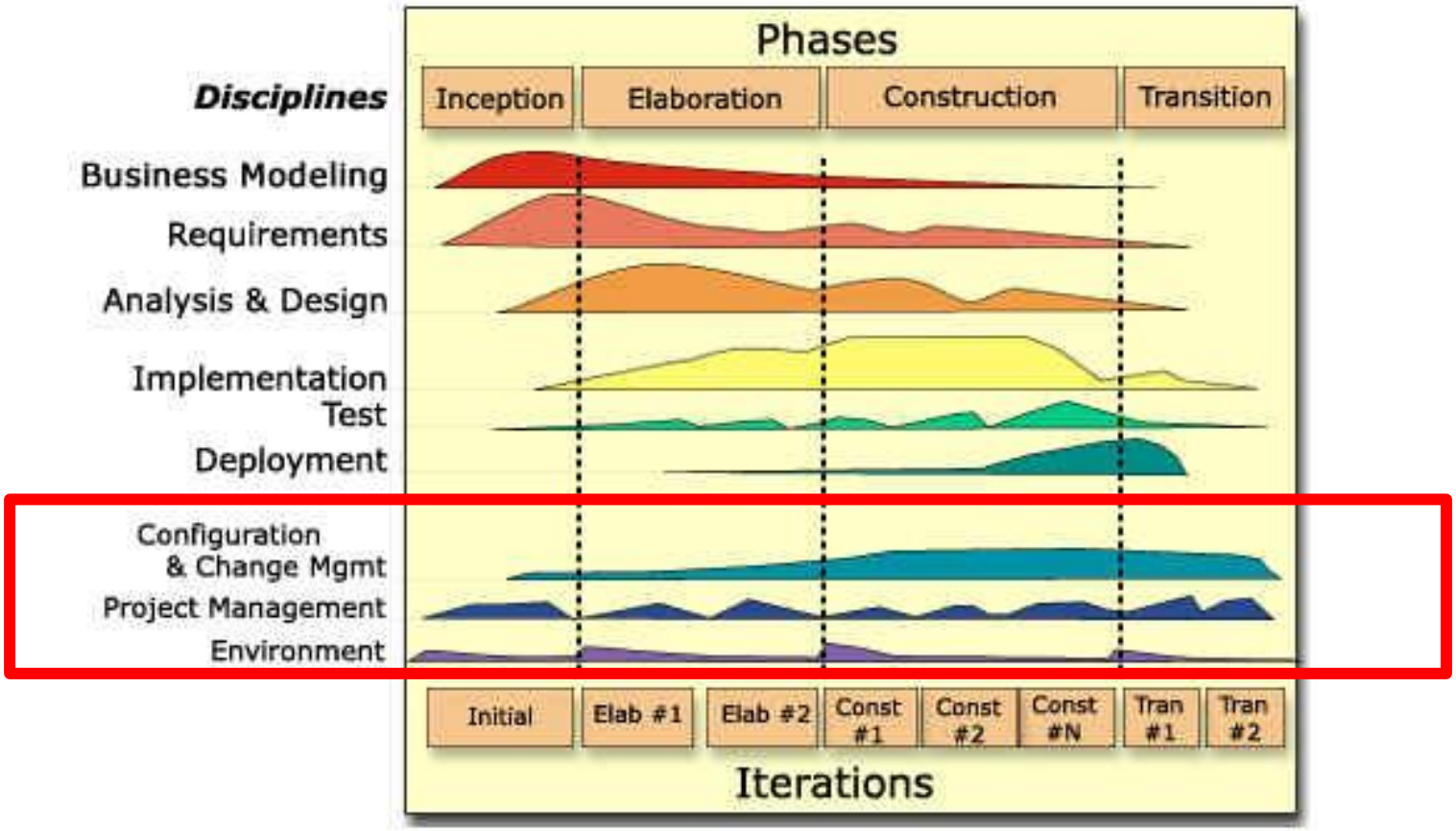


Temps

cascade

Besoin – analyse – conception – implémentation - tests

PU : cycle de vie & management

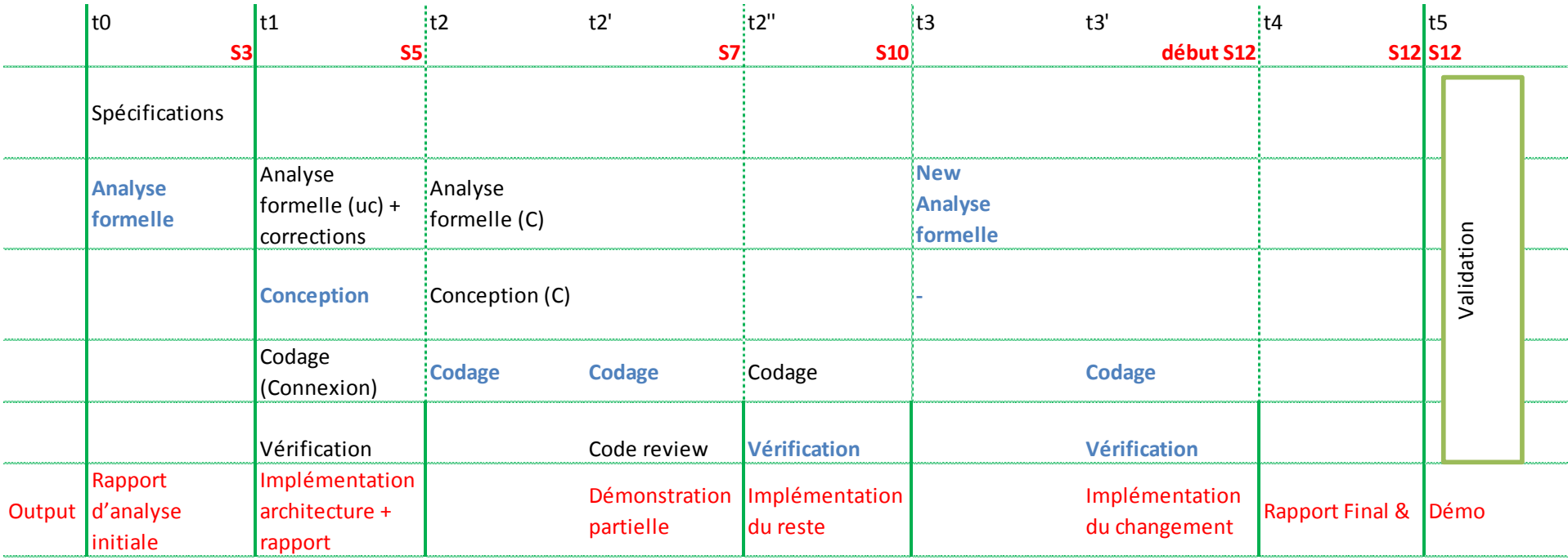


Exercices

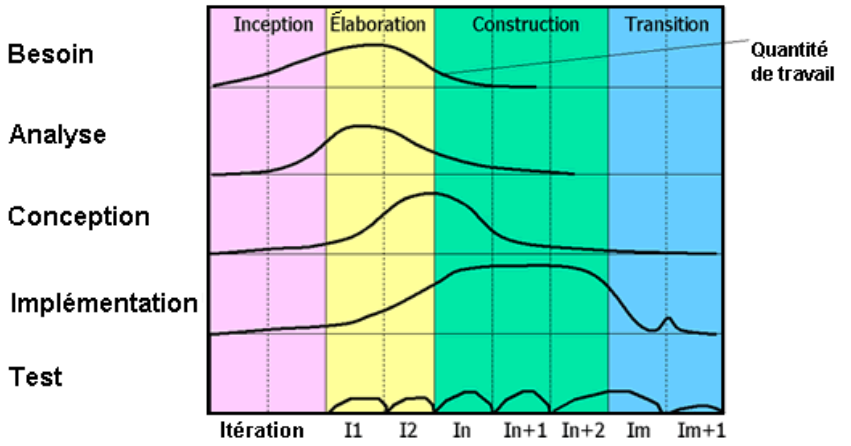
Exercice 2 : dessinez le cycle de vie employé, *à votre avis*, dans le cadre du cours de PAE.

Livrable	Date Livraison
Rapport d'analyse initiale	Début S3
Implémentation architecture & analyse uc	Fin S5
Revue du code / Démo d'avancement	S7-S9
Implémentation du reste	Fin S10
Implémentation demande changement	Début S12
Rapport final & démo	S12

PAE : le SVYpirale



PAE : le SVYpirale



	t0 S3	t1 S5	t2	t2' S7	t2'' S10	t3	t3' début S12	t4 S12	t5 S12
	Spécifications								Validation
	Analyse formelle	Analyse formelle (uc) + corrections	Analyse formelle (C)			**New Analyse formelle**			
		Conception	Conception (C)			-			
		Codage (Connexion)	**Codage**	**Codage**	Codage		**Codage**		
		Vérification		Code review	**Vérification**		**Vérification**		
Output	Rapport d'analyse initiale	Implémentation architecture + rapport		Démonstration partielle	Implémentation du reste		Implémentation du changement	Rapport Final &	Démo

Est un PU, pas une cascade, ni Y ou V, ni spirale !