

Génie Logiciel 2

– Chapitre 1 – Processus de développement UP

Dr. Sahar SMAALI

sahar.smaali@univ-constantine2.dz

Etudiants concernés

Faculté/Institut	Département	Niveau	Spécialité
NTIC	TLSI	Licence 3	GL

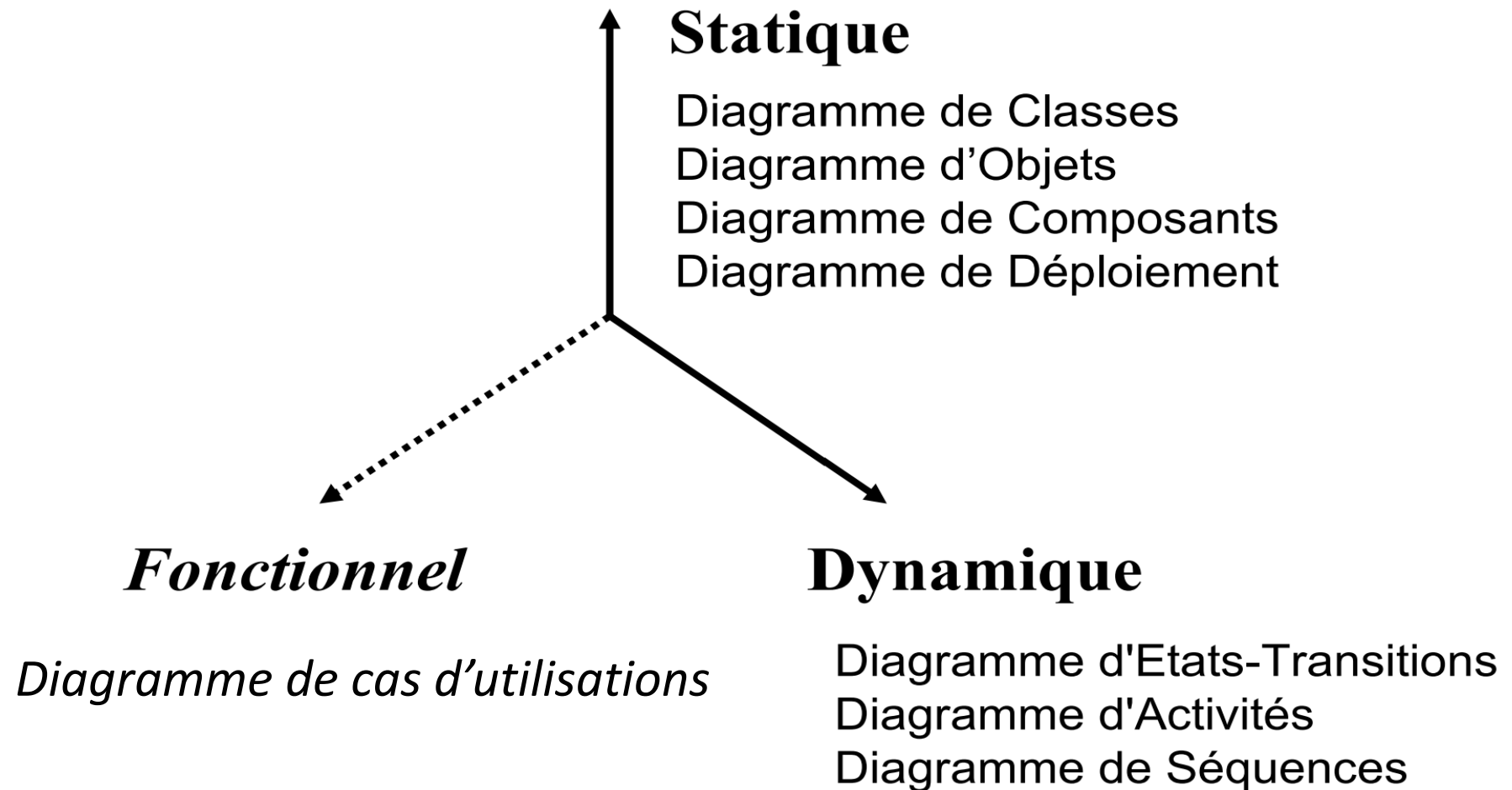
Unified Modeling Language

Modélisation

- Un modèle est une **abstraction** permettant de mieux comprendre un objet complexe (bâtiment, économie, atmosphère, cellule, logiciel, ...).
- Un petit **dessin** vaut mieux qu'un long discours. Les modèles sont donc souvent graphiques, même si l'objet à créer n'est pas matériel.
- Le modèle d'un système informatique sert :
 - de document d'échange **entre clients et développeurs**
 - d'outil de **conception**
 - de **référence** pour le développement
 - de **référence** pour la maintenance et l'évolution

Unified Modeling Language

- UML est le langage standard pour la modélisation objet.



UML n'est qu'un langage

UML

- Notation ou formalisme pour représenter, partager l'information, uniformiser, mécaniser (transformation vers le code)

UML n'est pas un processus

- Il ne définit pas le processus d'élaboration des modèles.
- Dans quel ordre doit-on utiliser les différents types de diagrammes?
- A quel moment de la conception d'un système doivent-ils intervenir?

Processus ou Méthode ?

Définition

Une séquence d'étapes, en partie ordonnées, qui concourent à l'obtention d'un système logiciel ou à l'évolution d'un système existant.

Unified Process (UP)

Unified Process (UP)

- Processus **générique** de développement de logiciel .
- Développée par les concepteurs d'UML .
- Regroupe les activités à mener pour **transformer les besoins des utilisateurs en système logiciel**.
- Répond à **4** questions :
 - **QUI** participe au projet?
 - **QUOI**, qu'est-ce qui est produit durant le projet?
 - **QUAND** est réalisé chaque livrable?
 - **COMMENT** doit-il être réalisé?

Concepts de base

Artefacts (quoi?)

- Éléments produits au cours du projet
- Documentent le système et le projet
- Ex: Architecture, Code source, Modèles, Exécutables, etc.

Rôles(qui?)

- Joués par les intervenants dans un projet
- Ex: Architectes, Analystes, Développeurs, Testeurs, Managers, autres.

Activités (comment?)

- 5 grandes activités : Expression des besoins, Analyse, Conception, Implémentation, Test.
- Multiples sous-activités: Concevoir une classe, Dessiner un DCU.

Workflow(quand?)

- Enchainement des activités qui produisent des artefacts.

Activités d'UP

Activités 1

Expression des besoins

- **Objectif**

- Capturer les besoins des utilisateurs et clients.

- **Artefacts**

- Modèle de cas d'utilisations
- Spécifications non fonctionnelles
- Listes des Risques
- Maquette IHM

- **Rôles**

- Analystes systèmes

- **Objectif**

- Comprendre les besoins et les exigences formulés par le client

- **Artefacts**

- Modèle de d'analyse
 - Modèle Fonctionnel
 - Modèle statique
 - Modèle dynamique

- **Rôles**

- Analystes et concepteurs

Activités 2

Analyse et Conception

- **Objectif**
 - Concevoir l'architecture du système
- **Artefacts**
 - Modèle de conception
 - Modèle de déploiement
- **Rôles**
 - Concepteurs et Architectes

Activités 3

Implémentation

- **Objectif**

- Raffiner le modèle de conception et produire des composants logiciels.

- **Artefacts**

- Modèle de composants
- Prototype ou version du logiciel

- **Rôles**

- Développeurs et Intégrateurs

Activités 4

Test

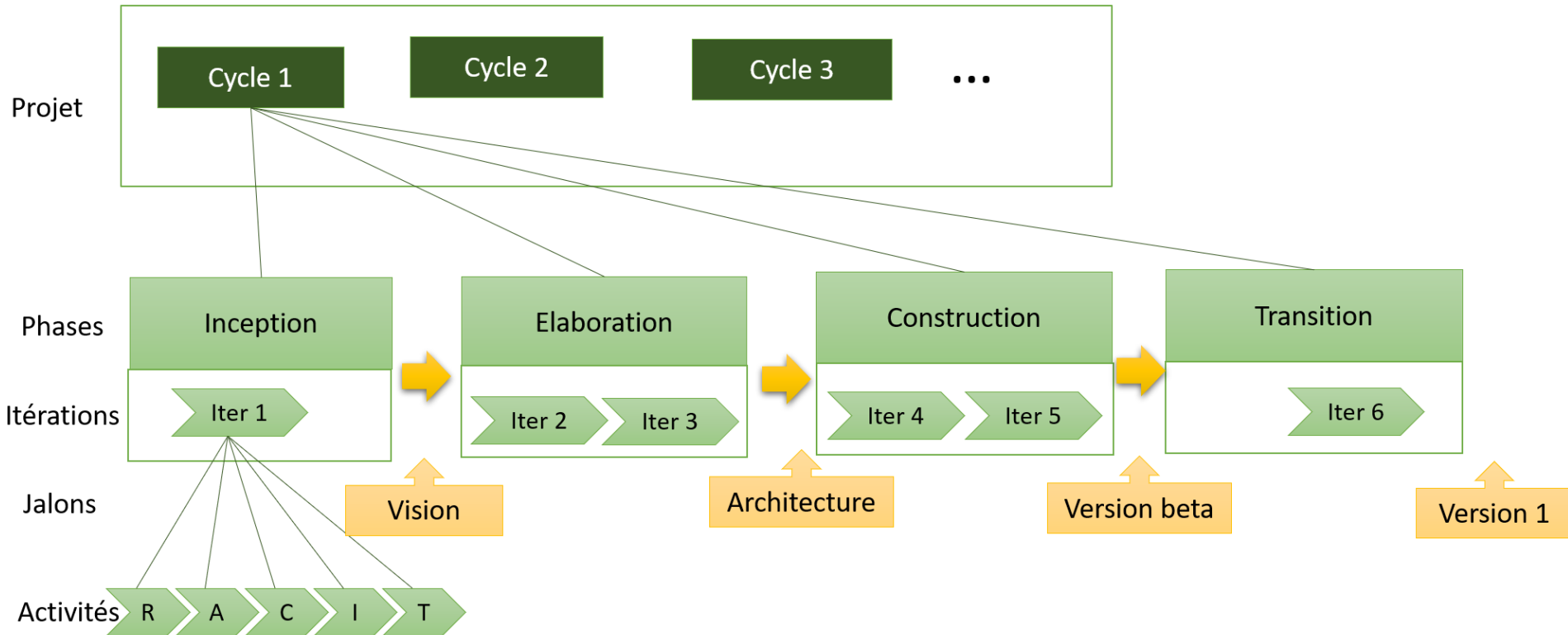
- **Objectif**
 - Test des composants logiciels développés
- **Artefacts**
 - Plan du test
 - Script de test
- **Rôles**
 - Testeurs

Phases d'UP

4 Phases du Cycle de vie

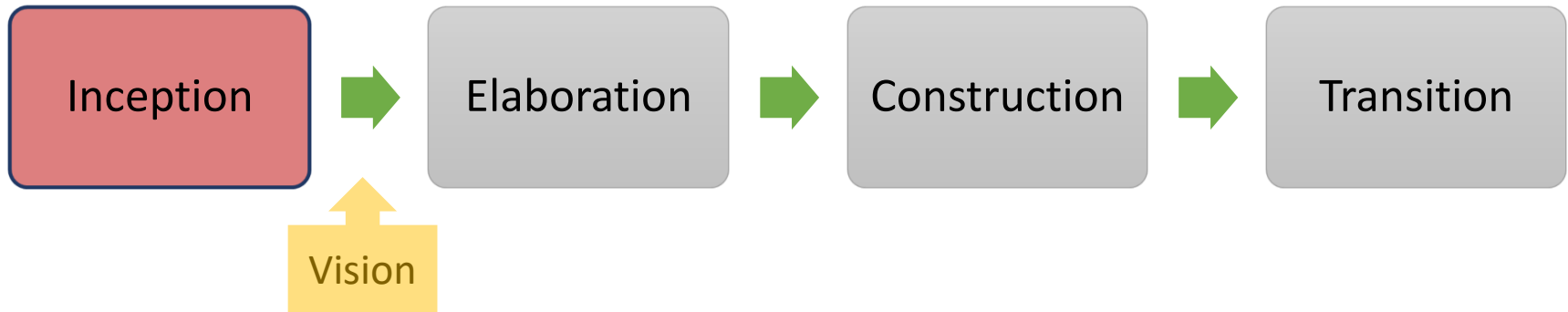
- Considerer un produit logiciel par rapport à ses versions :
 - **Un cycle produit une version du logiciel.**
- Gérer chaque cycle de développement comme un projet ayant **quatre phases**.
 - chaque phase se termine par un **jalón** permettant de prendre des décisions
 - Les **Jalons** correspondent à des **étapes d'évaluation** de la phase terminée, et de lancement de la phase suivante

4 Phases du Cycle de vie



Phases 1

Etude préliminaire (Inception)



● Déterminer:

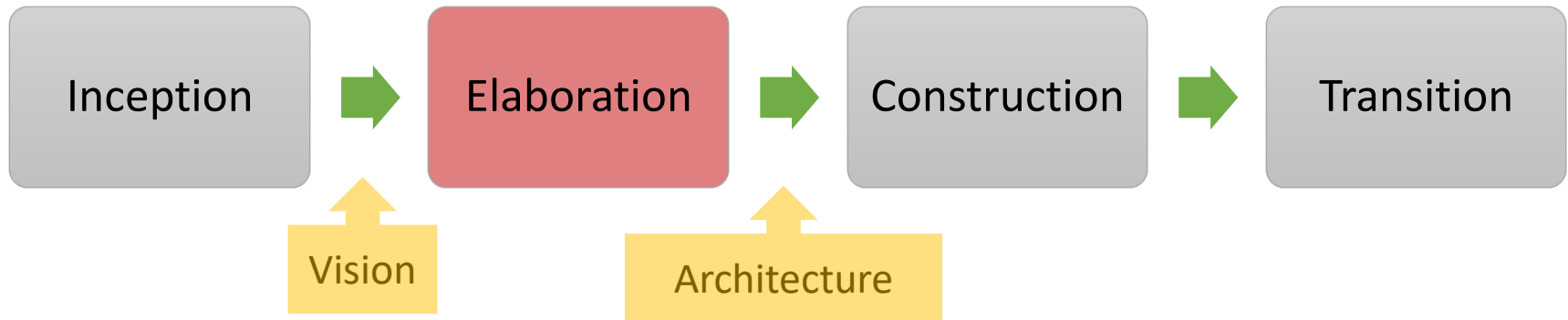
- que fait le système ?
- à quoi pourrait ressembler l'architecture ?
- quels sont les risques ?
- quel est le coût estimé du projet ? Comment le planifier ?

● Jalon

- « vision du projet » = document
- Accepter le projet ?
- Coût faible

Phase 2

Elaboration



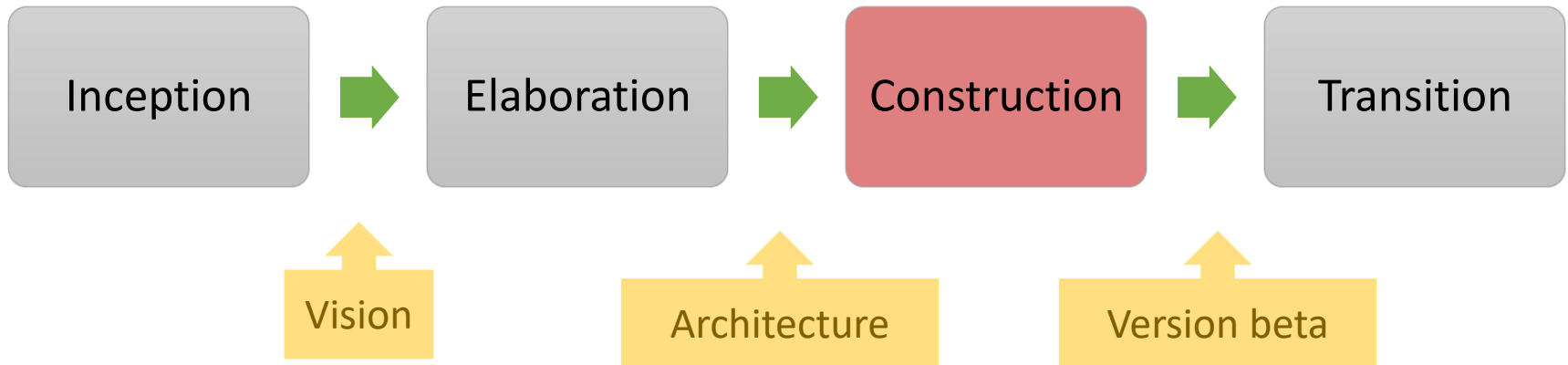
- Spécifier la plupart des cas d'utilisation
- Concevoir l'architecture de base (squelette du système)
- Mettre en œuvre cette architecture
- Faire une planification complète

● Jalon

- « **architecture** » = premier prototype
 - démontre la validité des choix architecturaux
- Peut-on passer à la construction ?
- besoins, architecture, planning stables ? Risques contrôlés ?

Phase 3

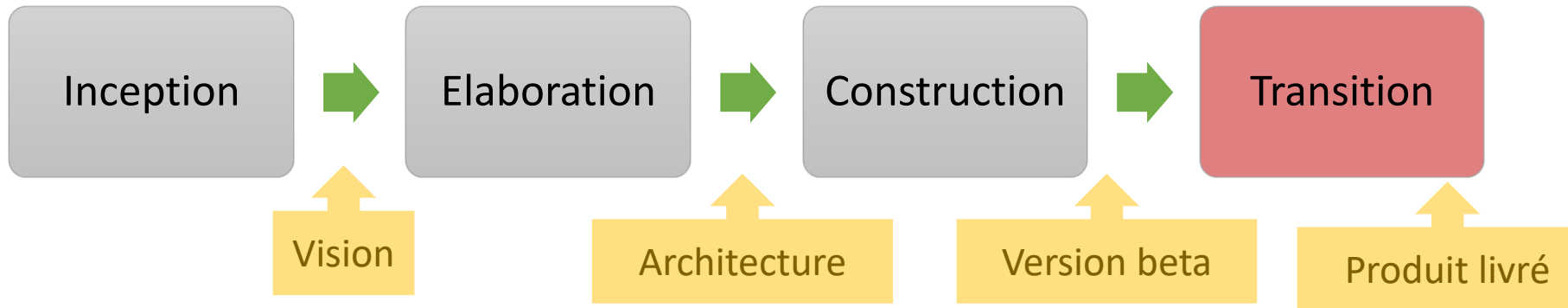
Construction



- Développer par **incréments**
- L'architecture reste stable malgré des changements mineurs
- Le produit contient au final tout ce qui avait été planifié, il reste quelques erreurs
- **Jalon**
 - « capacité opérationnelle initiale » = **version bêta**
 - Le produit est-il suffisamment correct pour être installé chez le / un client ?
 - Phase la plus coûteuse (> 50% du cycle)
 - englobe conception, codage, tests, intégration, etc.

Phase 4

Transition



- Livrer / déployer le produit
- Corriger le reliquat d'erreurs
- Améliorer le produit
- Former les utilisateurs

- **Jalon**

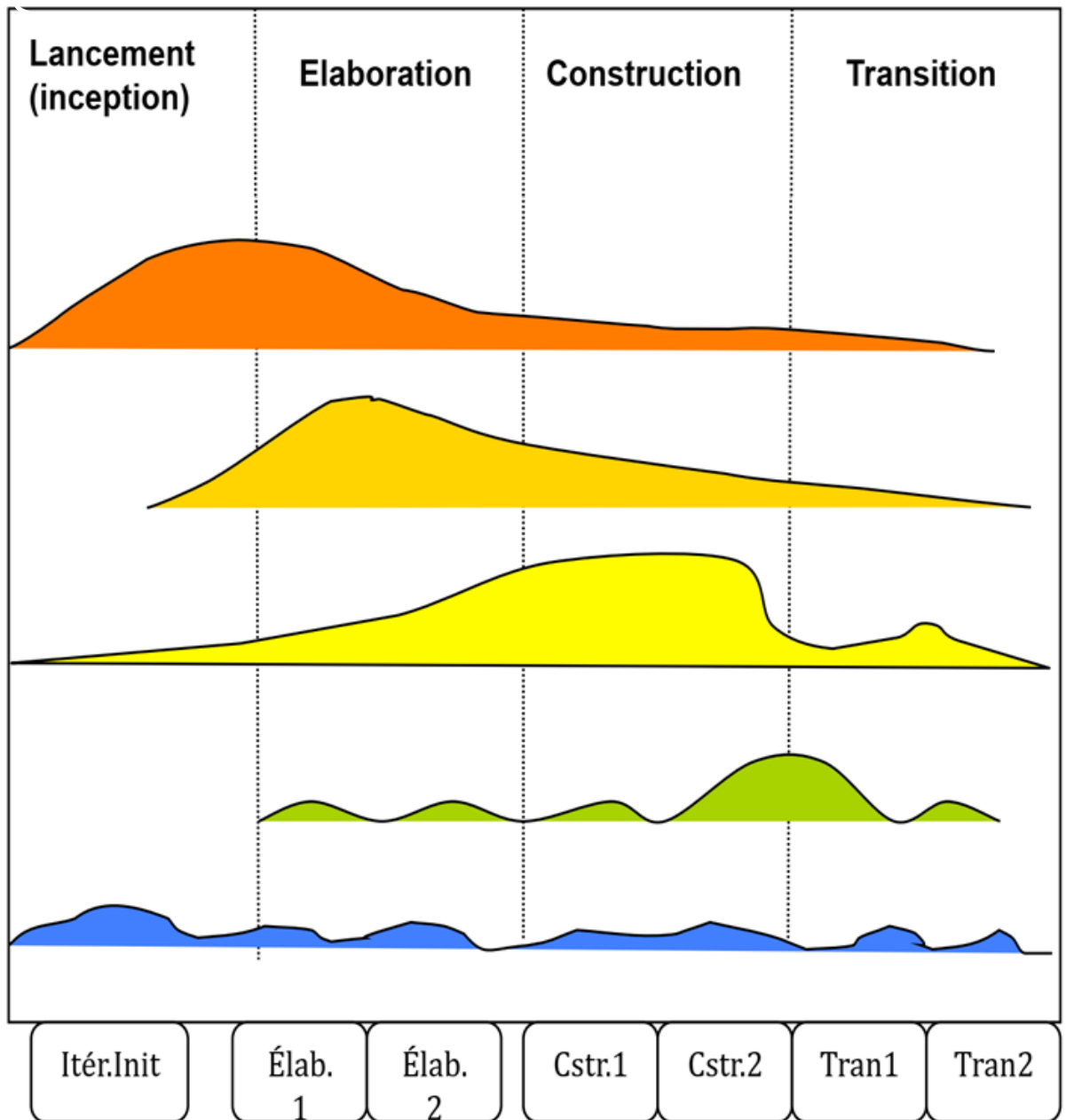
- « Livraison du produit » = produit déployé chez le client
- Tests suffisants ?
- Produit satisfaisant ?
- Manuels prêts ?

Phases et Activités

Architecture bidimensionnelle

- **Axe vertical** → représente les principaux enchaînements d'activités (workflows)
 - rend compte de l'aspect statique du processus qui s'exprime en termes de composants, de processus, d'activités, d'enchaînements, d'artefacts et de travailleurs.
- **Axe horizontal** → représente le temps et montre le déroulement du cycle de vie du processus
 - rend compte de l'aspect dynamique du processus qui s'exprime en terme de cycles, de phases, d'itérations et de jalons (points de repère)

Phases



Principales
activités du
processus

Recueil
des
besoins

Analyse et
Conception

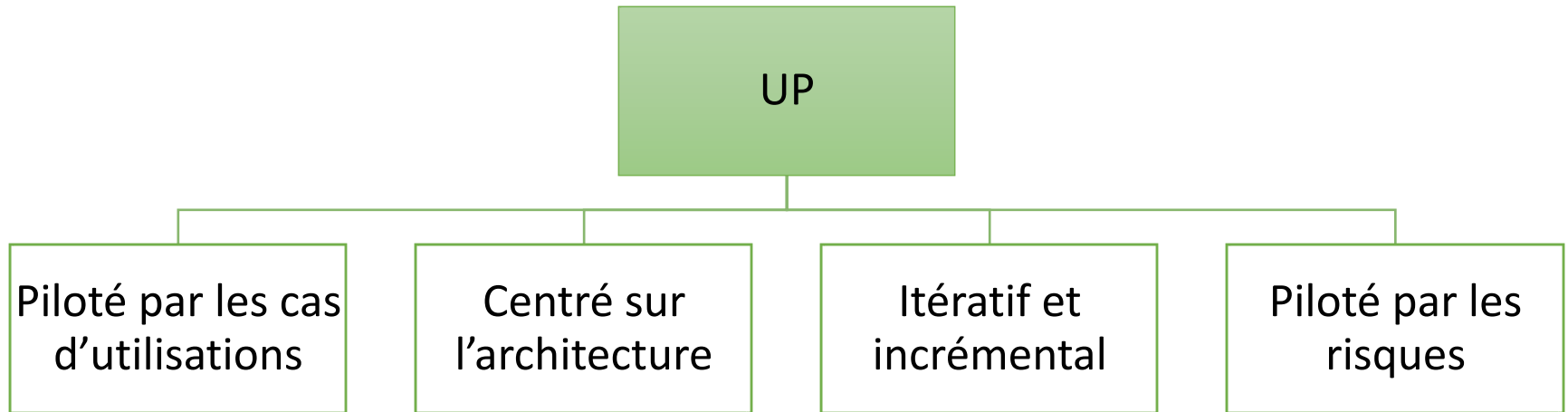
Implémentation

Tests

Gestion de
projet

Caractéristiques d'UP

Caractéristiques du processus UP



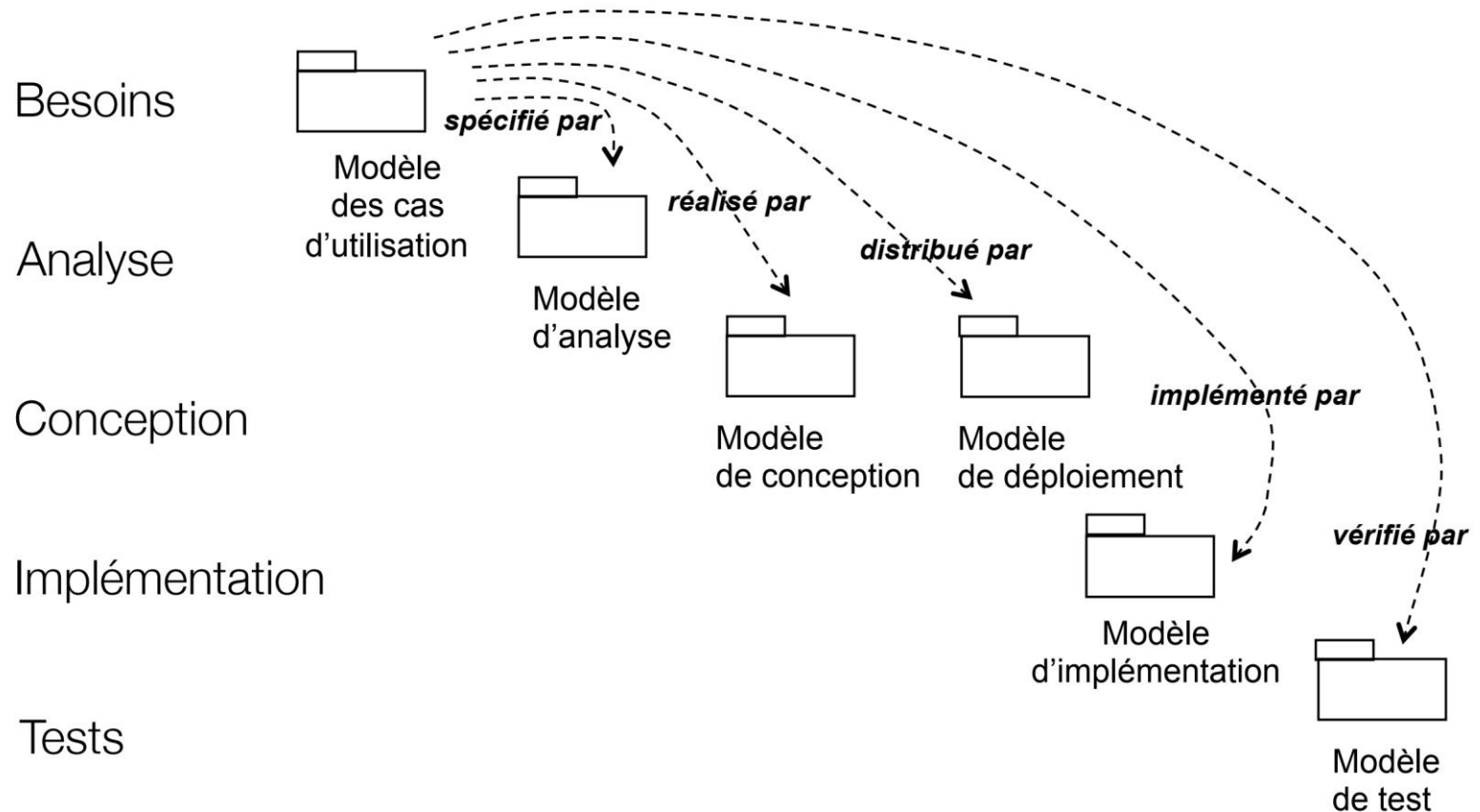
Caractéristiques du processus UP

UP Piloté par les cas d'utilisations

- Utilisés tout au long du cycle
- Validation des besoins / utilisateurs
- Point de départ pour l'analyse (découverte des objets, de leurs relations, de leur comportement) et la conception (sous-systèmes)
- Guide pour la construction des interfaces
- Guide pour la mise au point des plans de tests

Caractéristiques du processus UP

UP piloté par les cas d'utilisations



Caractéristiques du processus UP

Centré sur l'architecture

- **L'architecture regroupe les différentes vues du système**
 - fonctionnel, structurel, comportemental et technique
- **L'architecture doit prévoir la réalisation de tous les cas d'utilisation.**
- **Elle est souvent influencée par :**
 - Les langages et frameworks envisagés pour le développement;
 - La plate-forme sur laquelle devra s'exécuter le système;
 - Les composants et les bibliothèques réutilisables disponibles pour le développement;
 - Les considérations de déploiement, les systèmes existants et les besoins non fonctionnels (performance, fiabilité, sécurité, etc)

Caractéristiques du processus UP

Itératif et incrémental

- Projet découpé en mini-projets :
 - des itérations qui donnent lieu à des incréments
- Chaque itération :
 - donne lieu à un incrément
 - comprend un certain nombre de cas d'utilisation et doit traiter en priorité les risques majeurs.
 - reprend les livrables dans l'état où les a laissés l'itération précédente et les enrichit progressivement (incrémental).

Caractéristiques du processus UP

Itératif et incrémental

- Gestion de la **complexité**
- Accélération du rythme de développement grâce à des **objectifs clairs et à court terme.**
- Maîtrise précoce des **risques** élevés
 - **diminution de l'échec**
 - architecture mise à l'épreuve rapidement (**prototype réel**)
- Intégration continue et progrès immédiatement visibles
- Prise en compte des **modifications de besoins**
 - feedback, implication des utilisateurs et adaptation précoce

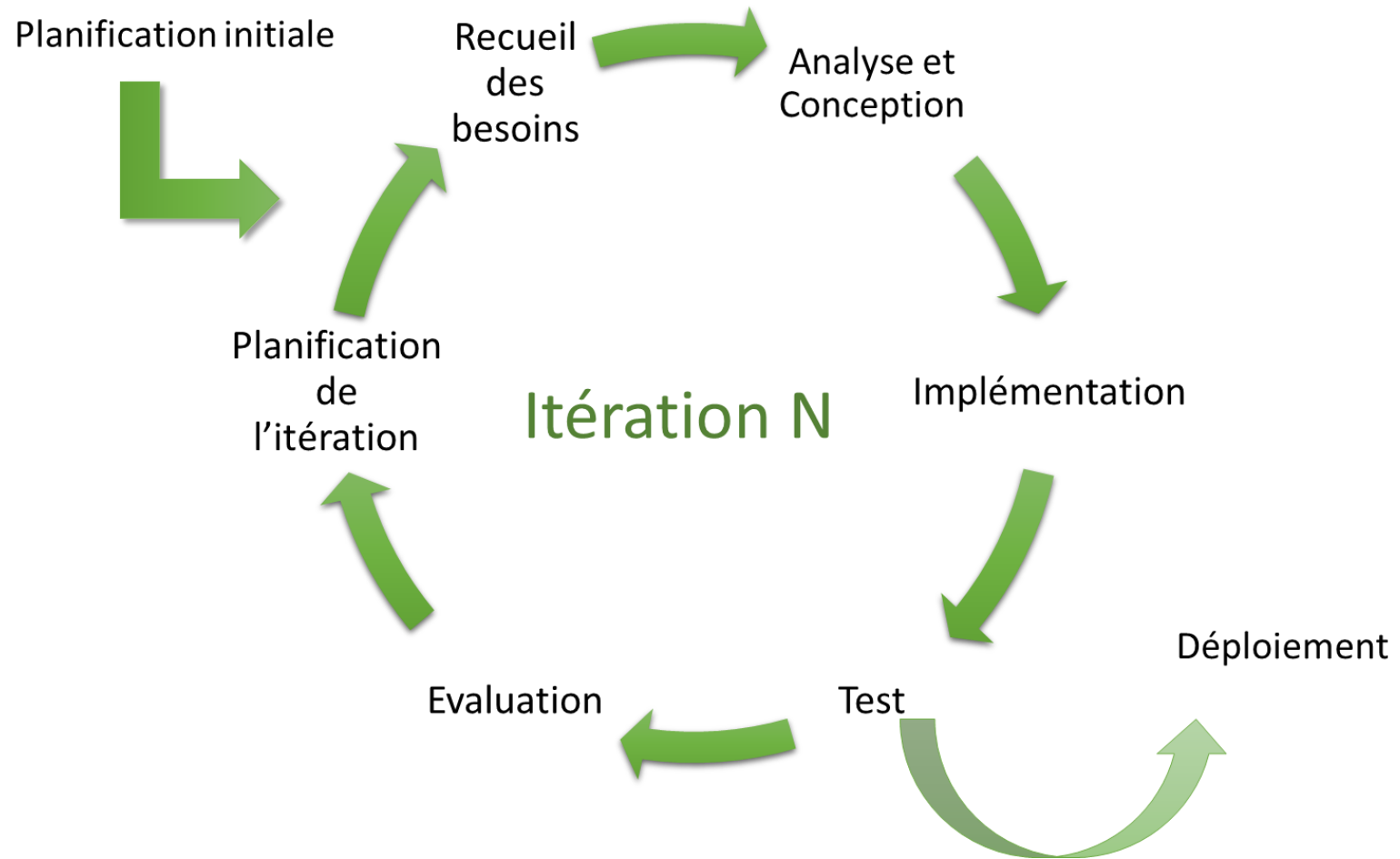
Caractéristiques du processus UP

Piloté par les risques

- Risques de différentes natures (fonctionnels, techniques, financiers, ou autres) .
- L'analyse des risques doit être présente à tous les phases et activités du processus de développement.
- S'attaquer en priorité aux risques les plus cruciaux qui menacent le plus la réussite du projet.

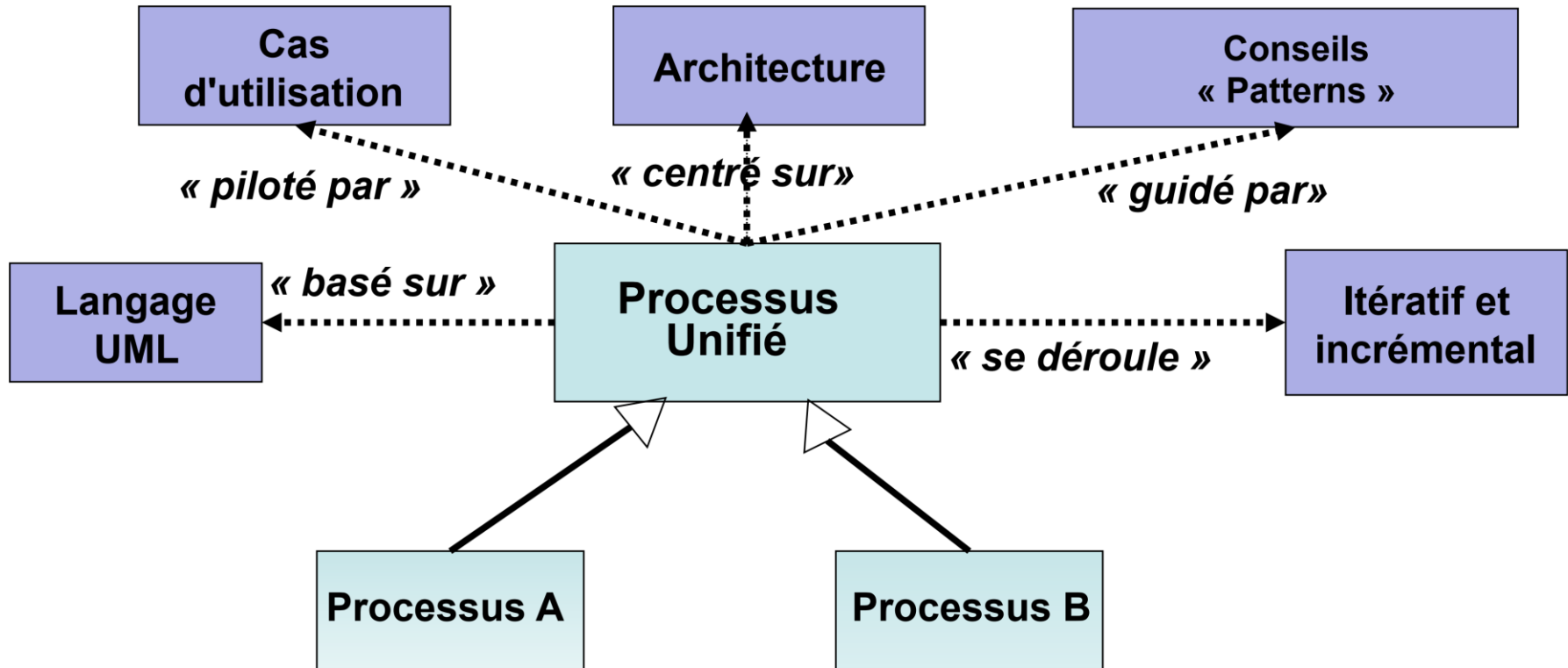
Caractéristiques du processus UP

Itérations et gestion des risques



Conclusion

Conclusion



Conclusion

- UP regroupe de **bonnes pratiques**.
- UP est **non figé et générique** (hautement adaptable : individus, cultures, ...)
- un projet particulier est **une instance** de ce cadre adaptée au **contexte du projet** (taille, personnels, entreprise, compréhension du processus, etc.)
- Les **adaptations** d'UP les plus connues sont :
 - RUP : Rational Unified Process
 - XP : eXtreme Programming
 - 2TUP : Two Tracks Unified Process