

3^{ème} année licence

Option: SCI

Module: CL2

Chapitre 1: introduction a la modélisation -UML & UP-

Dr. Meriem Kermani

Année 2018/2019

Plan

Modélisation

- ❑ Modélisation
- ❑ Cycle de vie d'un logiciel

Langage de modélisation UML

- ▶ UML
- ▶ Différents diagrammes UML

Méthode de Modélisation UP

- ▶ UP
- ▶ Etapes du processus
- ▶ Utilisation des diagrammes

1. MODÉLISATION 1/3

- **Principe** : Un modèle est une abstraction permettant de mieux comprendre un objet complexe (bâtiment, économie, atmosphère, cellule, logiciel, ...).
- **Autre principe** : *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.

1. MODÉLISATION EN INFORMATIQUE 2/3

- La construction d'un système d'information, d'un réseau, d'un logiciel **complexe**, de taille importante et par de nombreuses personnes oblige à modéliser.
- 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

1. MODÉLISATION 3/3

- **Langage de modélisation** : Une syntaxe commune, graphique, pour modéliser (OMT, UML,...).
- **Méthode de modélisation** : procédé permettant de construire un modèle aussi correct que possible et aussi efficacement que possible (MERISE, UP, ...).

2. CYCLE DE VIE DU LOGICIEL (1/2)

- ❑ **Analyse** des besoins et des risques
- ❑ **Spécifications** (ou conception générale) : l'architecture générale
- ❑ **Conception** détaillée : l'architecture détaillée et la description de tous les éléments
- ❑ **Codage** (ou implémentation) : traduction du modèle dans un langage de programmation
- ❑ **Tests** unitaires : vérification de chaque élément du logiciel par rapport aux spécifications

2. CYCLE DE VIE DU LOGICIEL (2/2)

- ❑ **Intégration** : vérification de l'interfaçage des différents éléments du logiciel.
- ❑ **Qualification** (ou recette) : vérification de la conformité du logiciel aux spécifications initiales
- ❑ Documentation
- ❑ Mise en production (**déploiement**)
- ❑ Maintenance corrective et évolutive

Langage de modélisation UML



3. UML (*UNIFIED MODELING LANGUAGE*)

LANGAGE UNIFIÉ DE MODÉLISATION

- **UML** langage unifié de modélisation
- Est le langage standard pour la modélisation objet. Dernière version 2.5 (mars 2015).
- Est un langage graphique destiné à la modélisation de systèmes et de processus.
- Est un langage basé sur l'approche par objets
- UML est édité par l'**OMG**
(**O**bject **M**anagment **G**roup)
responsable de la normalisation
des technologies objet.



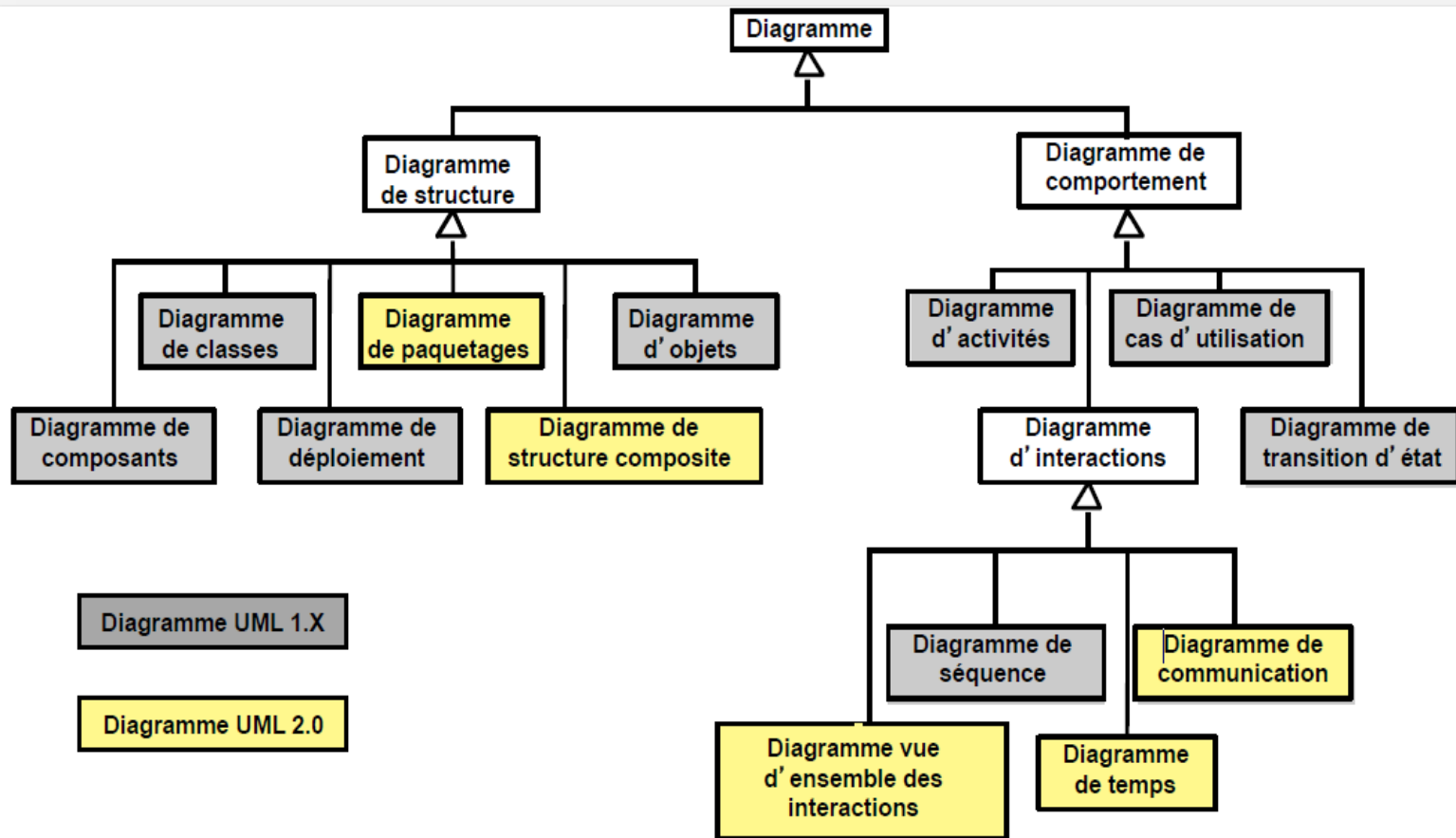
OBJECT MANAGEMENT GROUP

3 . UML (*UNIFIED MODELING LANGUAGE*)

LANGAGE UNIFIÉ DE MODÉLISATION

- UML est unifié car il provient de plusieurs notations qui l'ont précédé
- UML est un langage de modélisation très répandu, notamment grâce à sa **richesse sémantique** qui le rend abstrait de nombreux aspects techniques.

4. DIFFÉRENTS DIAGRAMME UML 1/4



4. DIFFÉRENTS DIAGRAMME UML 2/4

Diagrammes Structurels ou Diagrammes statiques (Structure Diagram)

- **Diagramme de classes (Class diagram):** il représente les classes intervenant dans le système
- **Diagramme d'objets (Object diagram):** il sert à représenter les instances de classes (objets) utilisées dans le système
- **Diagramme de composants (Component diagram):** il permet de montrer les composants du système d'un point de vue physique, tels qu'ils sont mis en oeuvre (fichiers, bibliothèques, bases de données...)
- **Diagramme de déploiement:** il sert à représenter les éléments matériels (ordinateurs, périphériques, réseaux, systèmes de stockage...) et la manière dont les composants du système sont répartis sur ces éléments matériels et interagissent avec eux

4. DIFFÉRENTS DIAGRAMME UML 3/4

Diagrammes Comportementaux ou Diagrammes dynamiques(Behavior Diagram)

- **Diagramme des cas d'utilisation (Use case diagram):** il décrit les possibilités d'interaction entre le système et les acteurs, c'est-à-dire toutes les fonctionnalités que doit fournir le système
- **Diagramme états-transitions (State Machine Diagram):** il ontre la manière dont l'état du système (ou de sous-parties) est modifié en fonction des événements du système
- **Diagramme d'activité (Activity Diagram):** variante du diagramme d'états-transitions, il permet de représenter le déclenchement d'événements en fonction des états du système et de modéliser des comportements parallélisables

4.DIFFÉRENTS DIAGRAMME UML 4/4

Diagrammes Comportementaux ou Diagrammes dynamiques(Behavior Diagram)

Diagramme d'interactions (Interaction Diagram):

- **Diagramme de séquence (Sequence Diagram):** la représentation séquentielle du déroulement des traitements et des interactions entre les éléments du système et/ou des acteurs
- **Diagramme de communication (Communication Diagram):** la représentation simplifiée d'un diagramme de séquence se concentrant sur les échanges de messages entre les objets
- **Diagramme global d'interaction (Interaction Overview Diagram):** variante du diagramme d'activité où les nœuds sont des interactions, permet d'associer les notations du diagramme de séquence à celle du diagramme d'activité, ce qui permet de décrire une méthode complexe

Méthode de modélisation

UP (*Unified Process*)

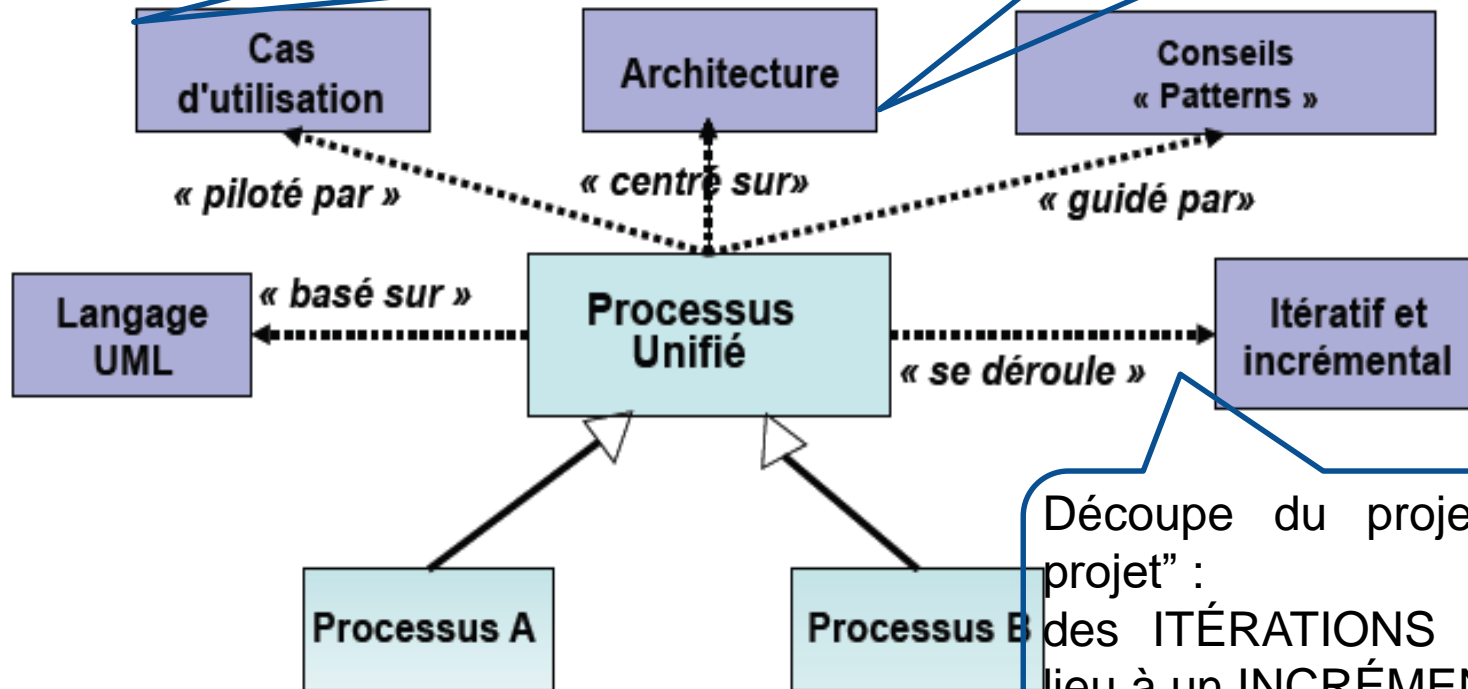
5. PROCESSUS UP

- Le processus unifié est un processus de développement logiciel : il regroupe les activités à mener pour transformer les besoins d'un utilisateur en système logiciel.

5.PROCESSUS UP 2/2

A partir des cas d'utilisation, les développeurs créent une série de modèles UML.

L'architecture regroupe les différentes vues du système qui doit être construit.



Découpe du projet en “mini-projet” :
des ITÉRATIONS qui donnent
lieu à un INCRÉMENT

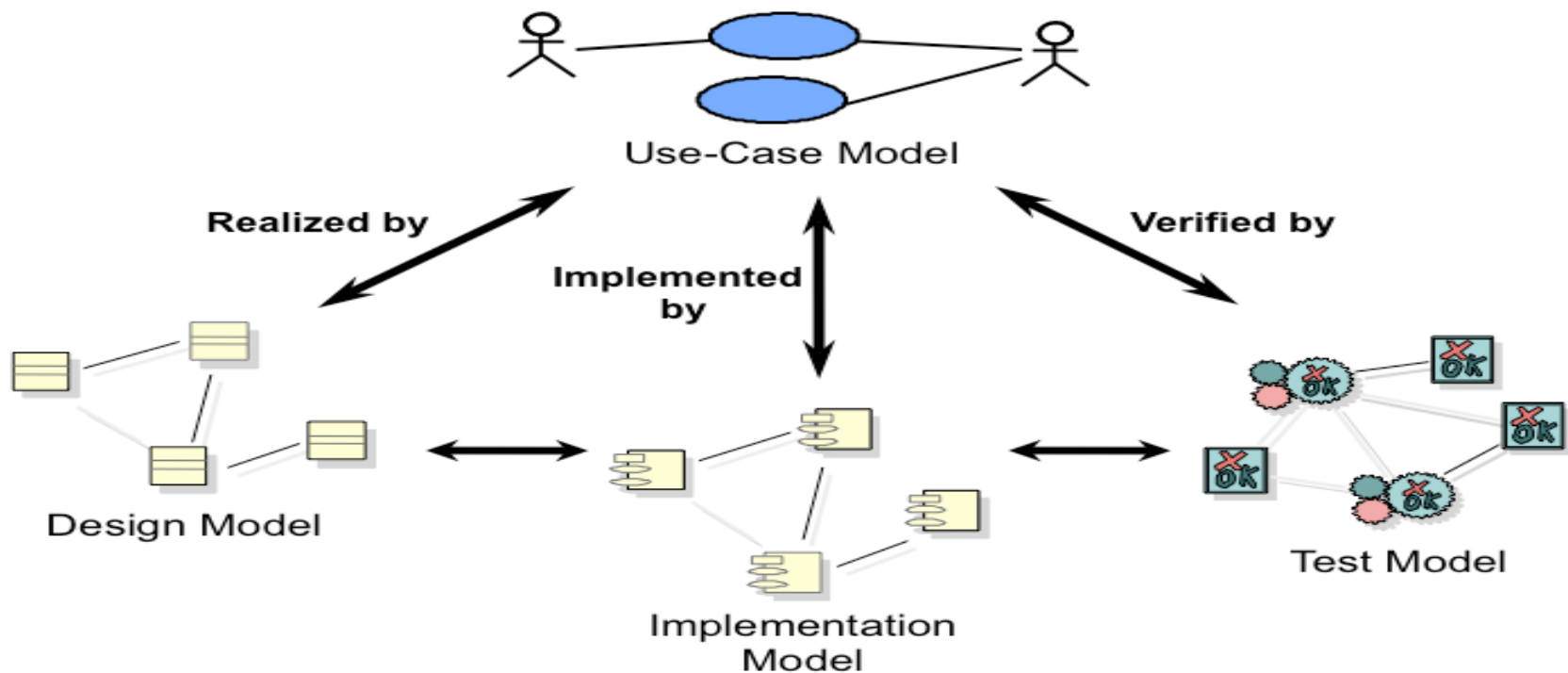
UP est **itérative, incrémentale**, pilotée par les **cas d'utilisations** (les besoins) et centrée sur l'**architecture** du logiciel.

5.PROCESSUS UP

CENTRÉ SUR L'UTILISATEUR

- Le processus de développement est centré sur l'utilisateur

Use cases drive software development



5.PROCESSUS UP

CENTRÉ SUR L'ARCHITECTURE

- Au sens de UP, une architecture :
 - Sert à comprendre le système lorsqu'il est complexe
 - Pilote le projet en découpant les tâches
 - Favorise la réutilisation
- On pourra parler
 - d'architecture logicielle (ou architecture logique) :
 - organisation à grande échelle des classes logicielles en
 - packages, sous-systèmes et couches
 - d'architecture de déploiement : décision de déploiement
 - des différents éléments
 - Notion de patterns architecturaux
 - □ ex. : Couches, MVC...

5.PROCESSUS UP

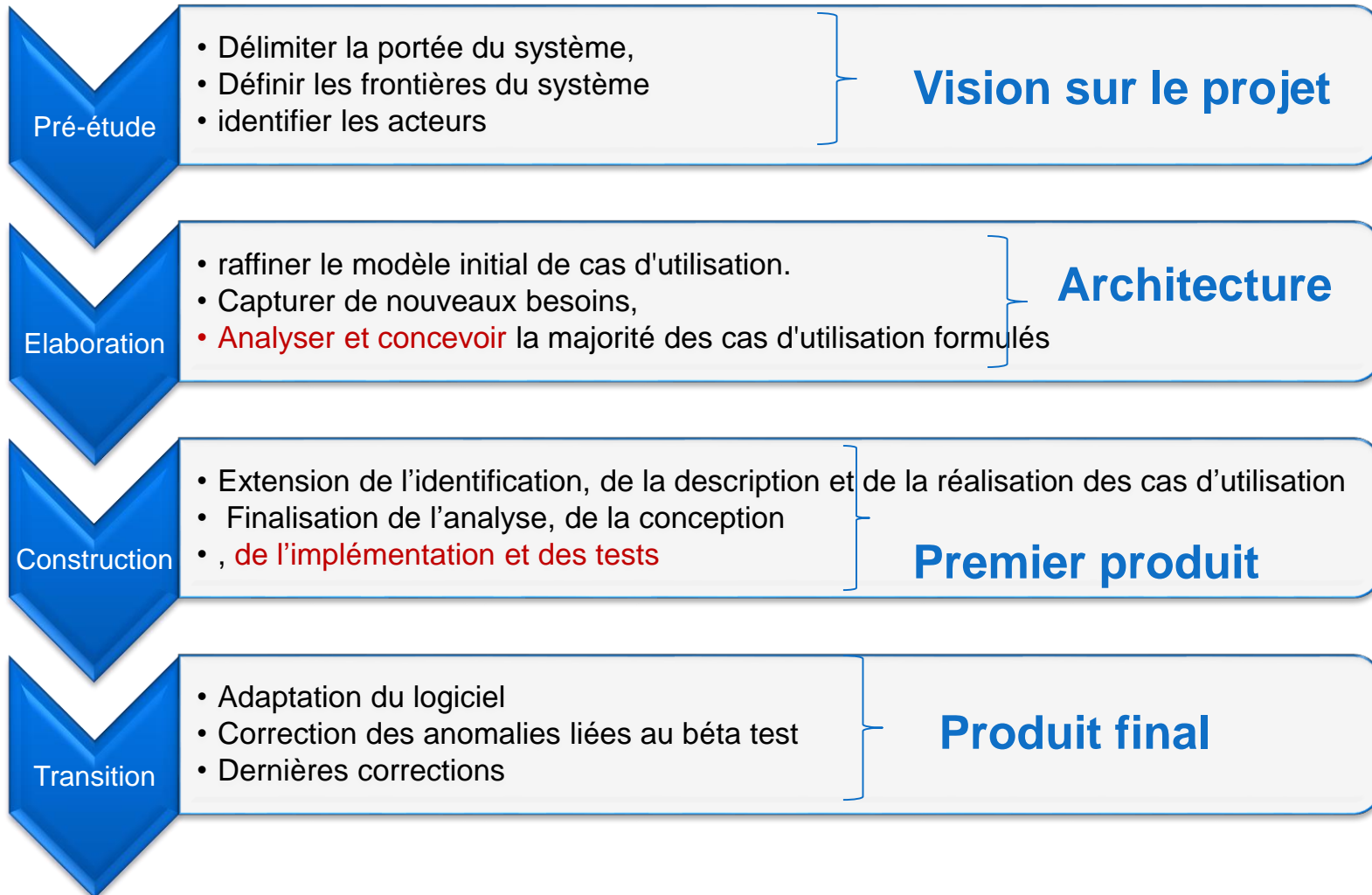
CENTRÉ SUR L'ARCHITECTURE

- Les cas d'utilisation ne sont pas suffisants comme lien pour l'ensemble des membres du projet
- L'architecture joue également ce rôle, en insistant sur la réalisation concrète de prototypes incrémentaux qui «démontrent» les décisions prises
- D'autre part
 - plus le projet avance, plus l'architecture est difficile à modifier
 - les risques liés à l'architecture sont très élevés, car très coûteux
- Objectifs pour le projet
 - établir dès la phase d'élaboration des fondations solides et évolutives pour le système à développer, en favorisant la réutilisation
 - l'architecture s'impose à tous, contrôle les développements ultérieurs, permet de comprendre le système et d'en gérer la complexité

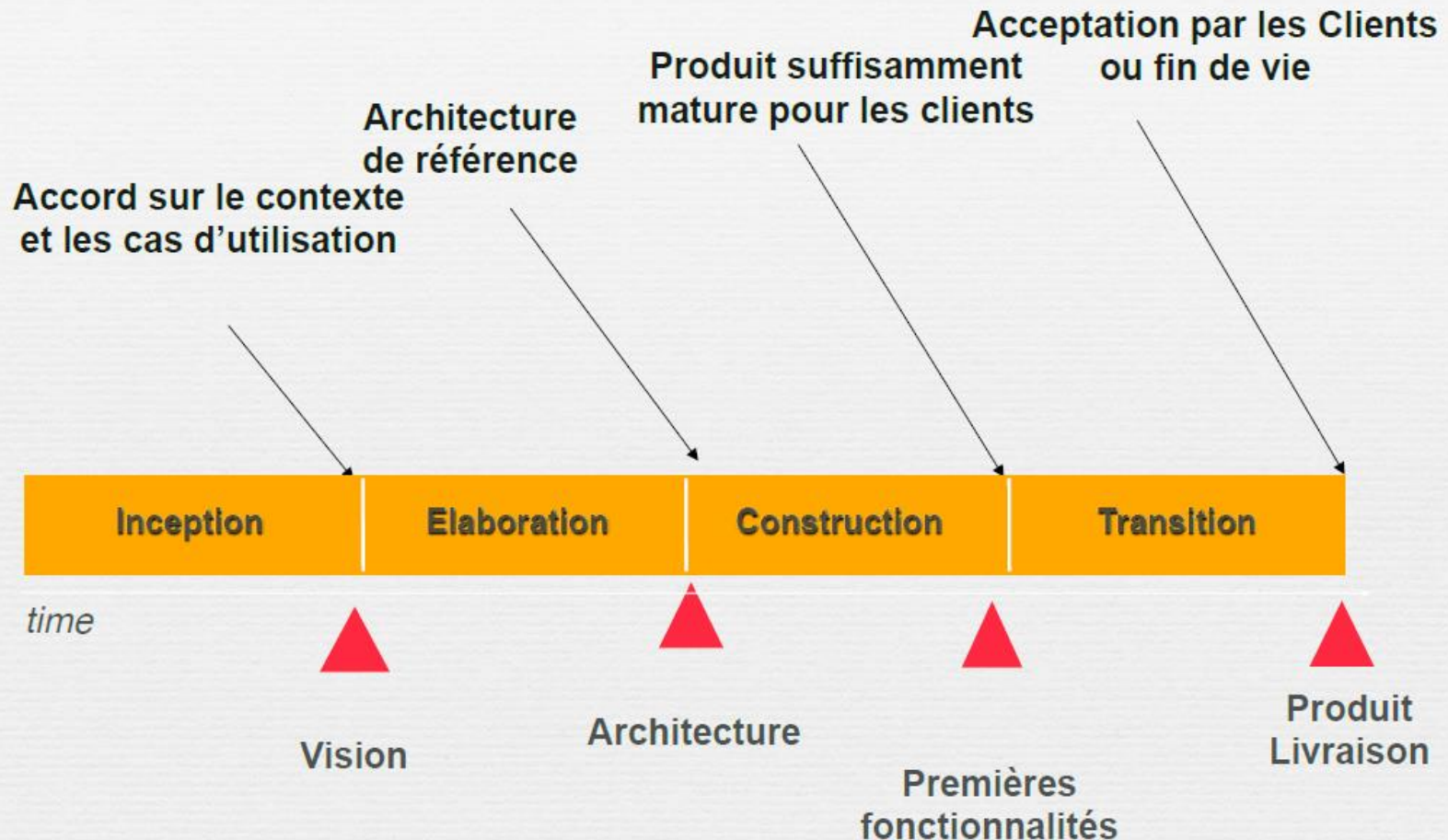
PROCESSUS UNIFIÉ ITÉRATIF ET INCRÉMENTAL

- Ordonnancement des itérations basé sur les priorités entre cas d'utilisation et sur l'étude du **risque**
- Une itération est une séquence **d'activités**
- Une itération se décompose en:
 - Une planification de l'itération
 - Analyse des besoins (raffinement)
 - Analyse et conception
 - Implémentation et tests
 - Évaluation
 - Livraison
- **Un risque est un événement redouté dont l'occurrence est plus ou moins prévisible et provoquant, lorsqu'il se produit, des dommages sur le projet.**

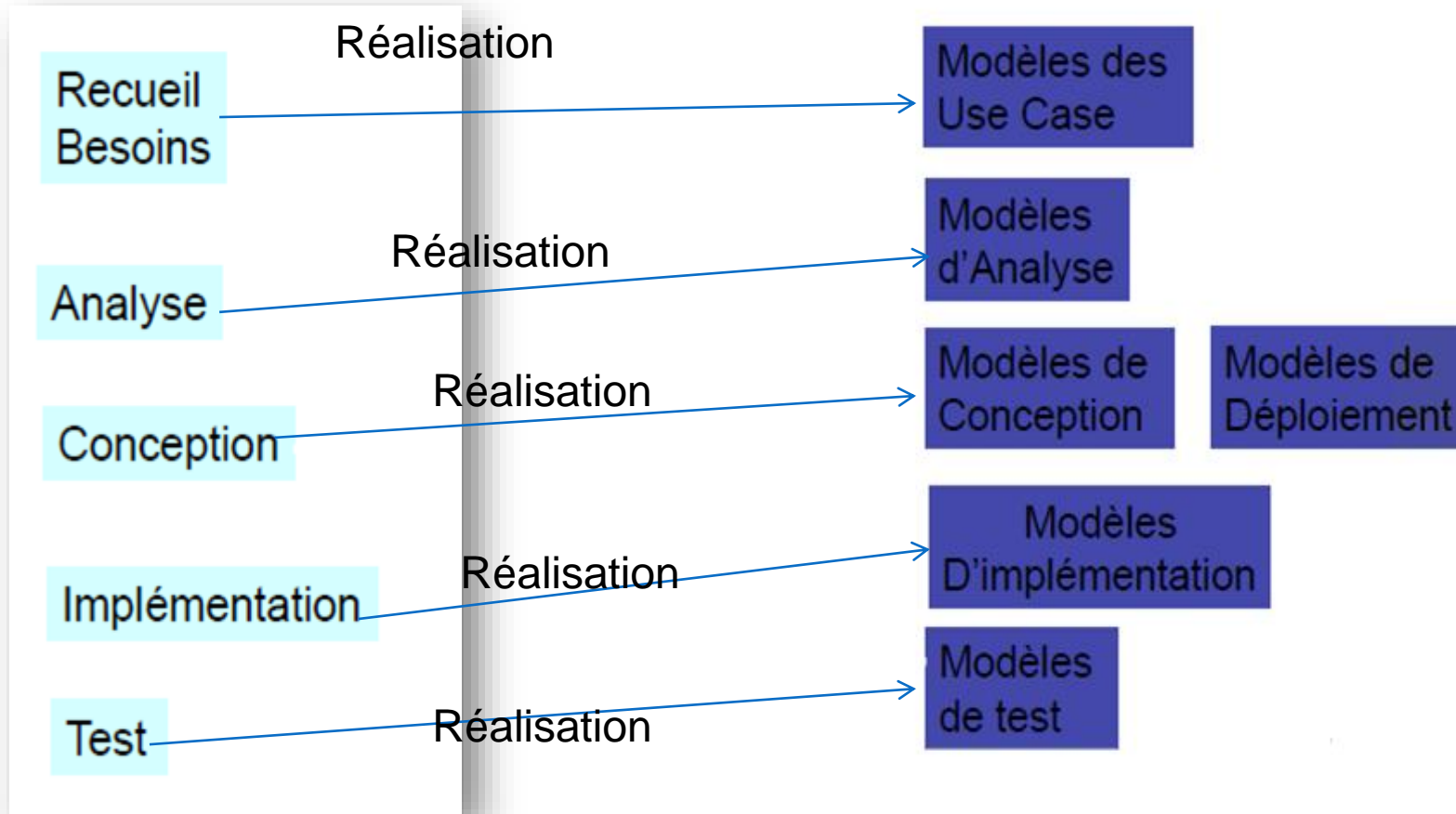
6.PHASES DU PROCESSUS 1/2



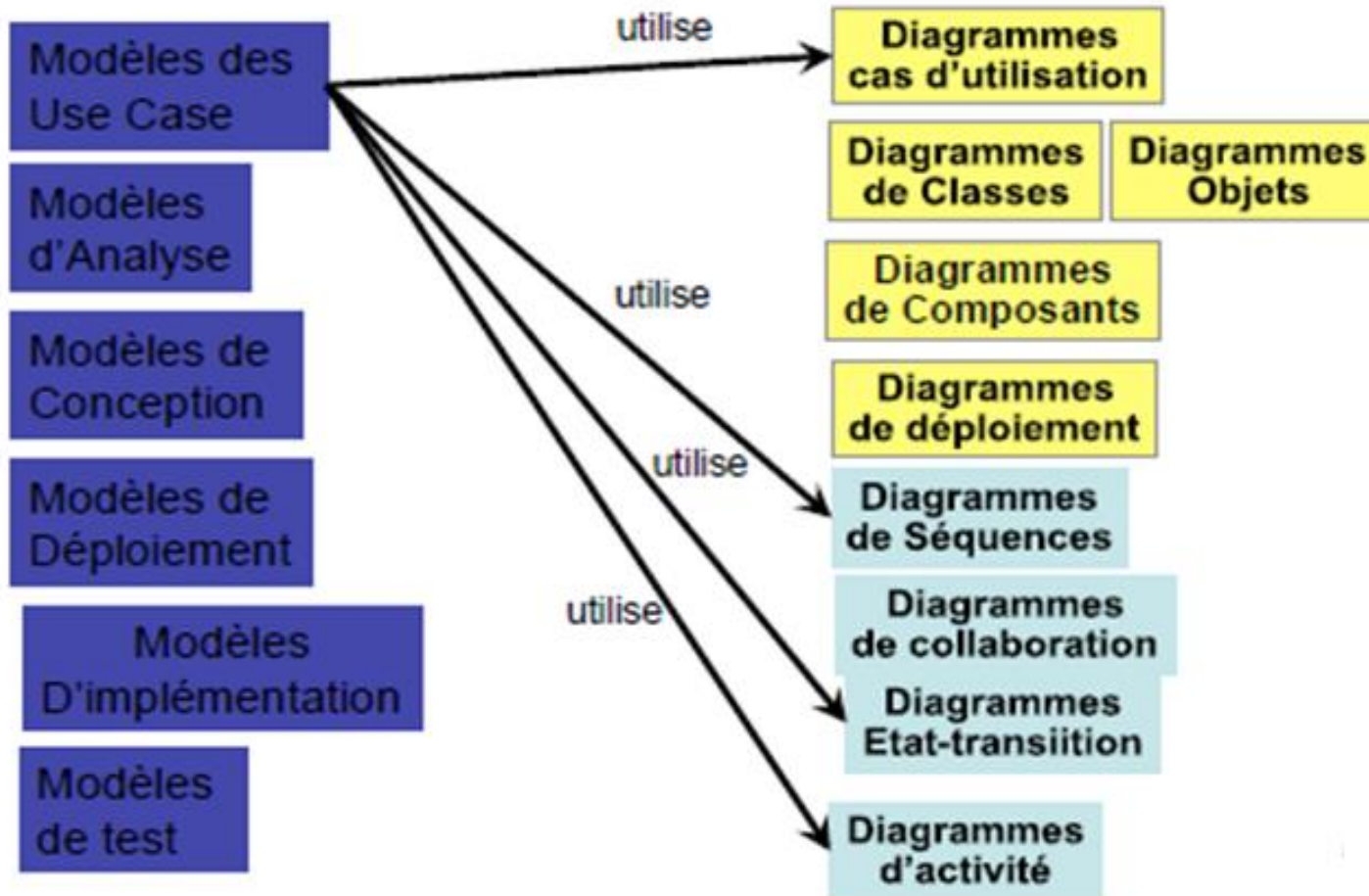
6.PHASES DU PROCESSUS 2/2



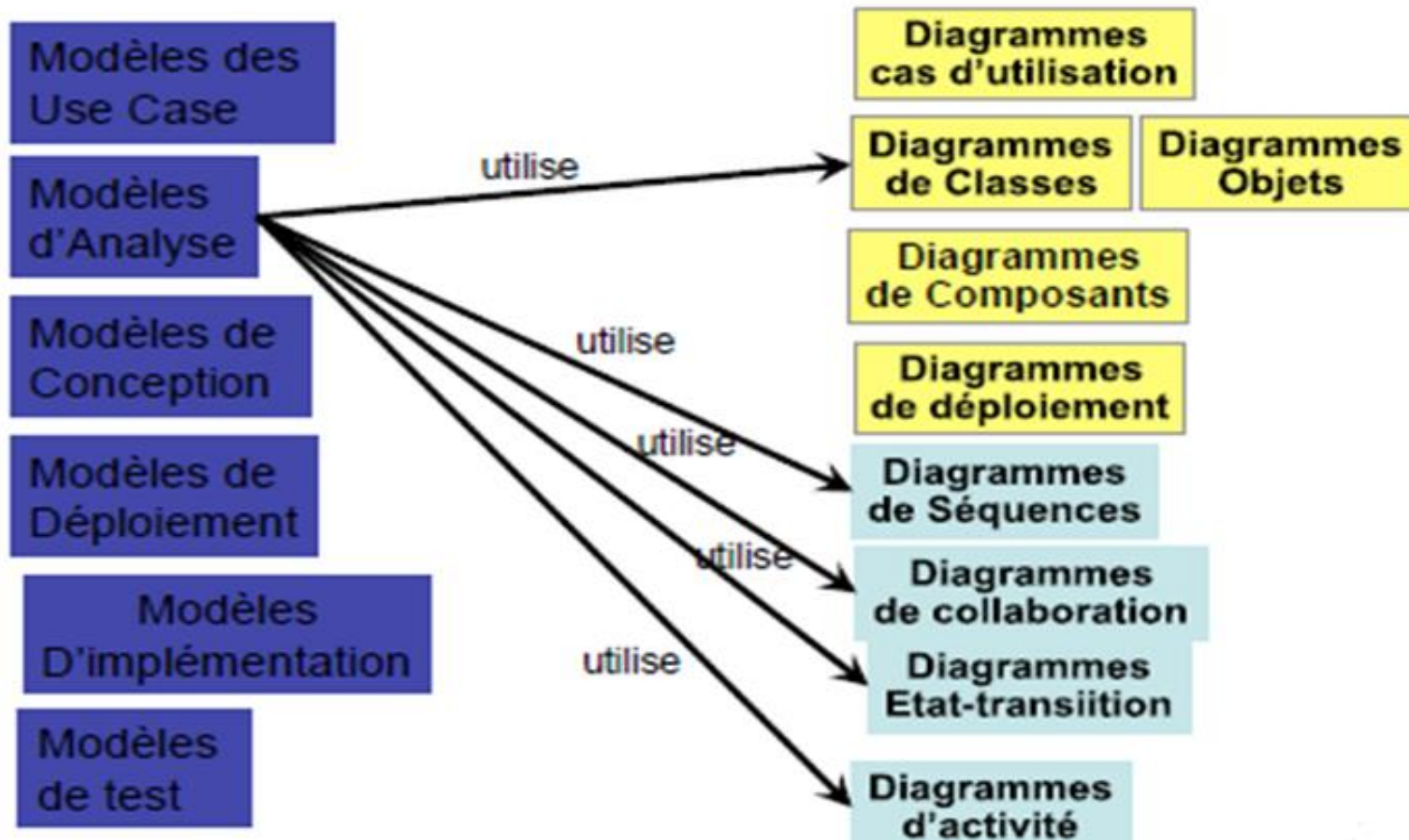
6. PHASES DU PROCESSUS 2/2



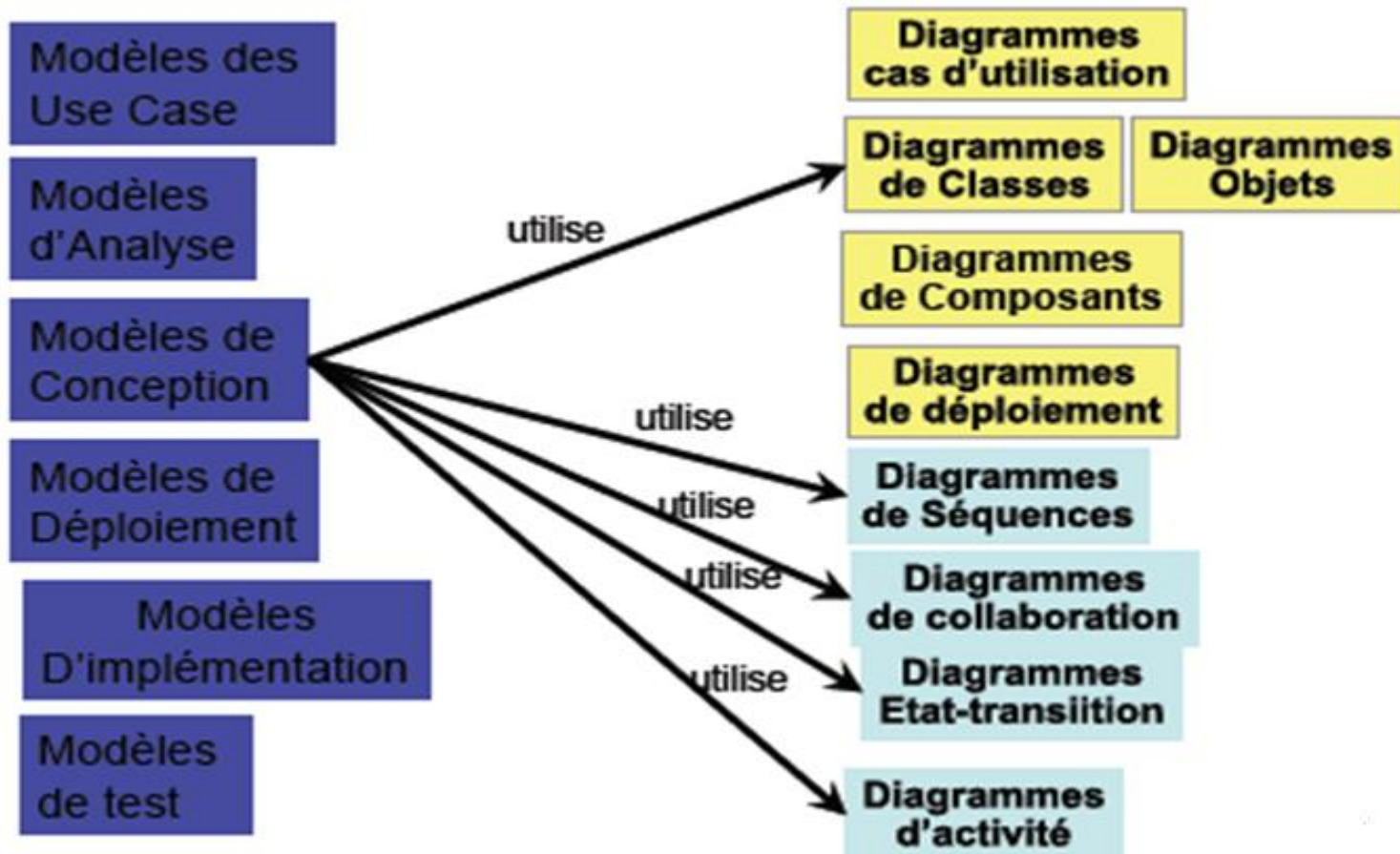
7. UTILISATION DES DIAGRAMMES 1/4



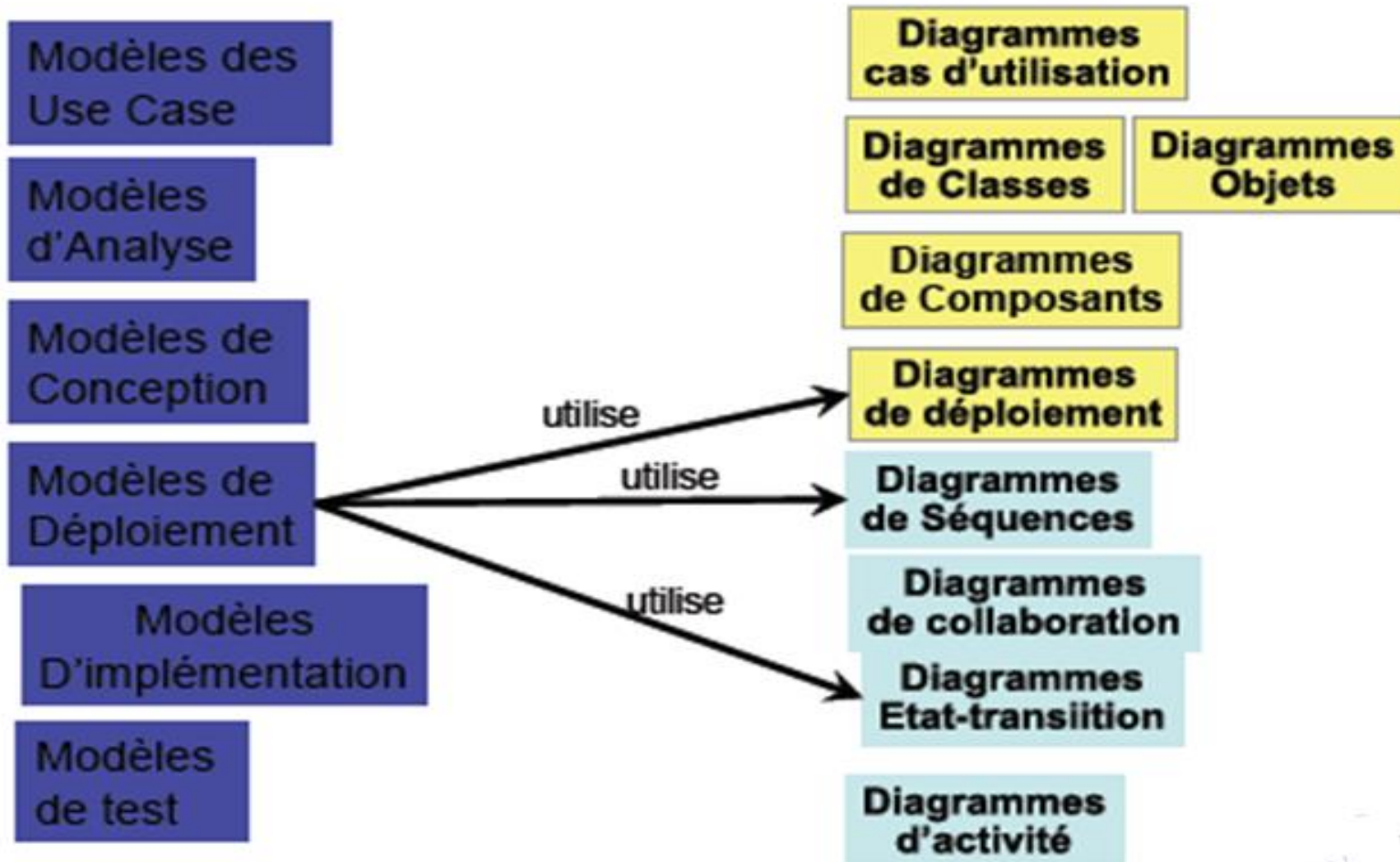
7. UTILISATION DES DIAGRAMMES 2/4



7. UTILISATION DES DIAGRAMMES 3/4



7. UTILISATION DES DIAGRAMMES 4/4



CONCLUSION

- Variantes UP
- Il existe plusieurs variantes:
 - RUP: Rational Unified Process (IBM)
Version industrielle avec une panoplie d'outils
 - UPEDU:
Version allégée pour des environnements académiques
- Toutes ces variantes:
 - élaborent des modèles qui seront interreliés
 - ont des mécanismes d'évaluation et d'adaptation du processus