Architecture logicielle : une introduction

GLO-3001
Architecture logicielle

Luc Lamontagne
Hiver 2010

Plan

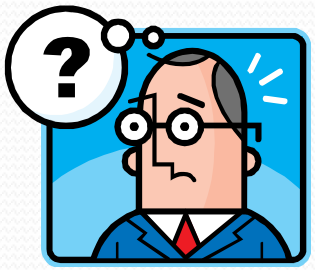
- Définitions
- Architecte logiciel
- Architecture logicielle
- Une bonne architecture logicielle ?
- Architecture vs. conception
- Styles, patrons et idiomes
- Conception logicielle vs. architecture
- Historique

Définitions

- Génie logiciel (*Software engineering*)

  L'application systématique et disciplinée
  d'approches quantifiables
  de développement et de maintenance de logiciels,
  et l'étude de ces approches.

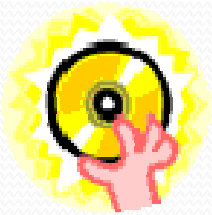
  (IEEE 610.12)



Besoins

Gestion de projet

Processus & outils



Produit logiciel

Définitions

- Architecture logicielle (*Software architecture*)

La structure ou les structures du système, comprenant:
les composantes logicielles,
les propriétés exposées de ces composantes, et
les interactions entre celles-ci.

(*Bass, Clements et Kazman 2003*)

Définitions

- L'architecture logicielle = abstraction d'un logiciel
  - Collection de composantes
    - Client, serveur, filtre, couche, contrôleur, agent
  - Topologie de composantes (relations)
    - Agrégation, classification, généralisation, association
  - Interactions entre les composantes
    - Connecteurs
      - Propriétés des composantes
    - Mode d'interaction
      - Par ex. RPC, Partage de mémoire, synchronisation

Définitions

- Composantes
  - Orientée objet
    - Un module (*package*, librairie)
    - Un groupe de classes
    - Encapsulation des méthodes
  - Procédurale
    - Un groupe de fonctions et de structures de données
- Connecteurs
  - La signature des composantes
    - Interfaces et services offerts par une composante
    - Fonctions et propriétés des composantes.

Pourquoi un architecte? (maison)

Pourquoi un architecte? (maison)

- Satisfaire les besoins du propriétaire
- Assurer la longévité de la maison
  - revente, charpente, catastrophe
- Planifier pour permettre les extensions possibles
- Faciliter l'entretien et le confort
- Contenir les coûts
- Guider les travailleurs qui vont la construire

Pourquoi un architecte? (maison)

- Besoins … — Description du mandat
- Longévité de la maison … — Robustesse
- Extensions possibles … — Modifiabilité
- Entretien … — Maintenabilité
- Confort… — Utilisabilité
- Contenir les coûts .. — Gestion du projet
- Guider les travailleurs… — Devis/plan du projet

Pourquoi un architecte logiciel?

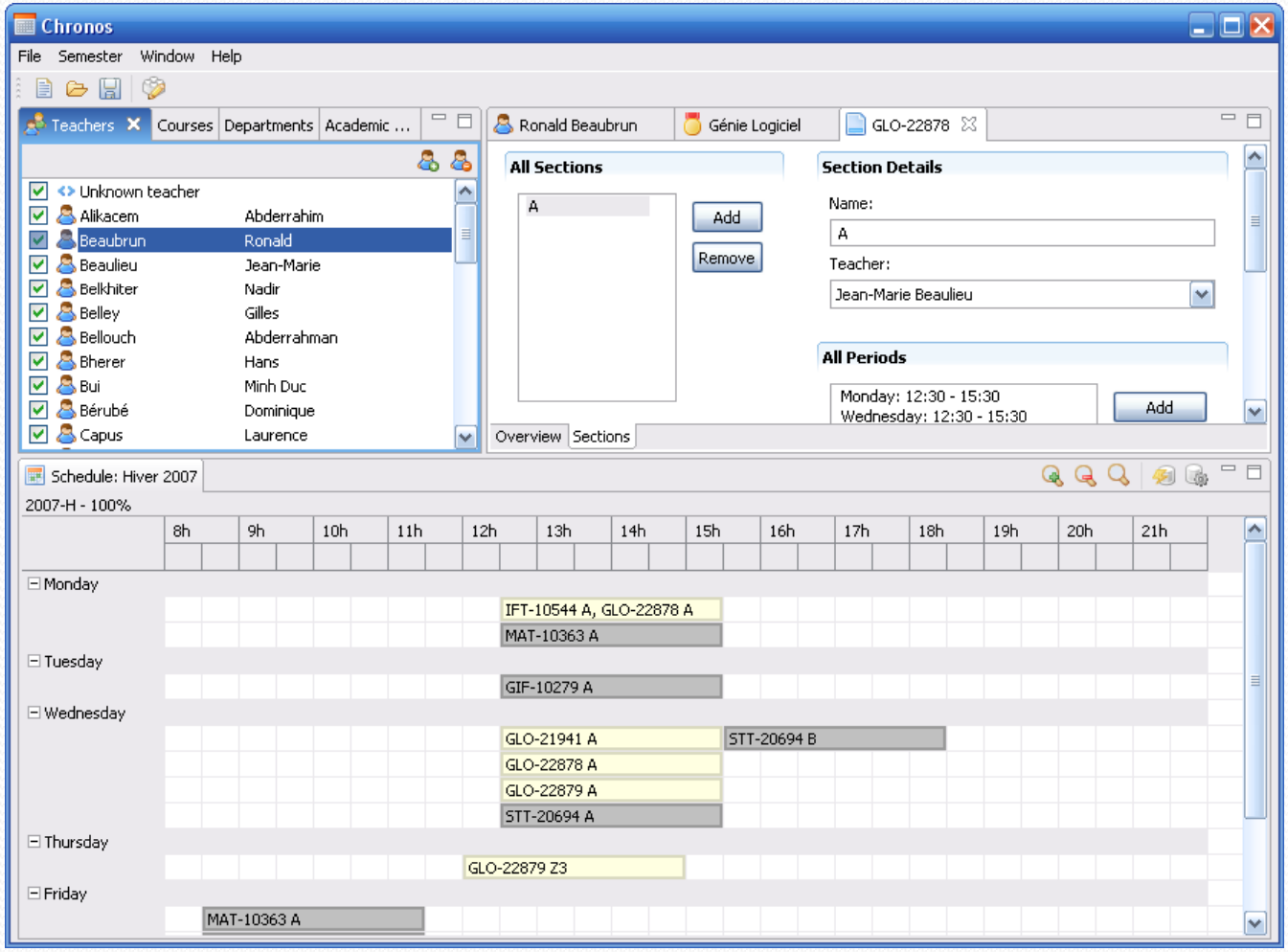
- Assurer la stabilité et la robustesse du système
- Permettre une maintenance relativement facile
- Permettre les évolutions futures
- Contenir les coûts
- Minimiser les complications
- Guider les programmeurs dans leurs efforts de développement

Une bonne architecture ?

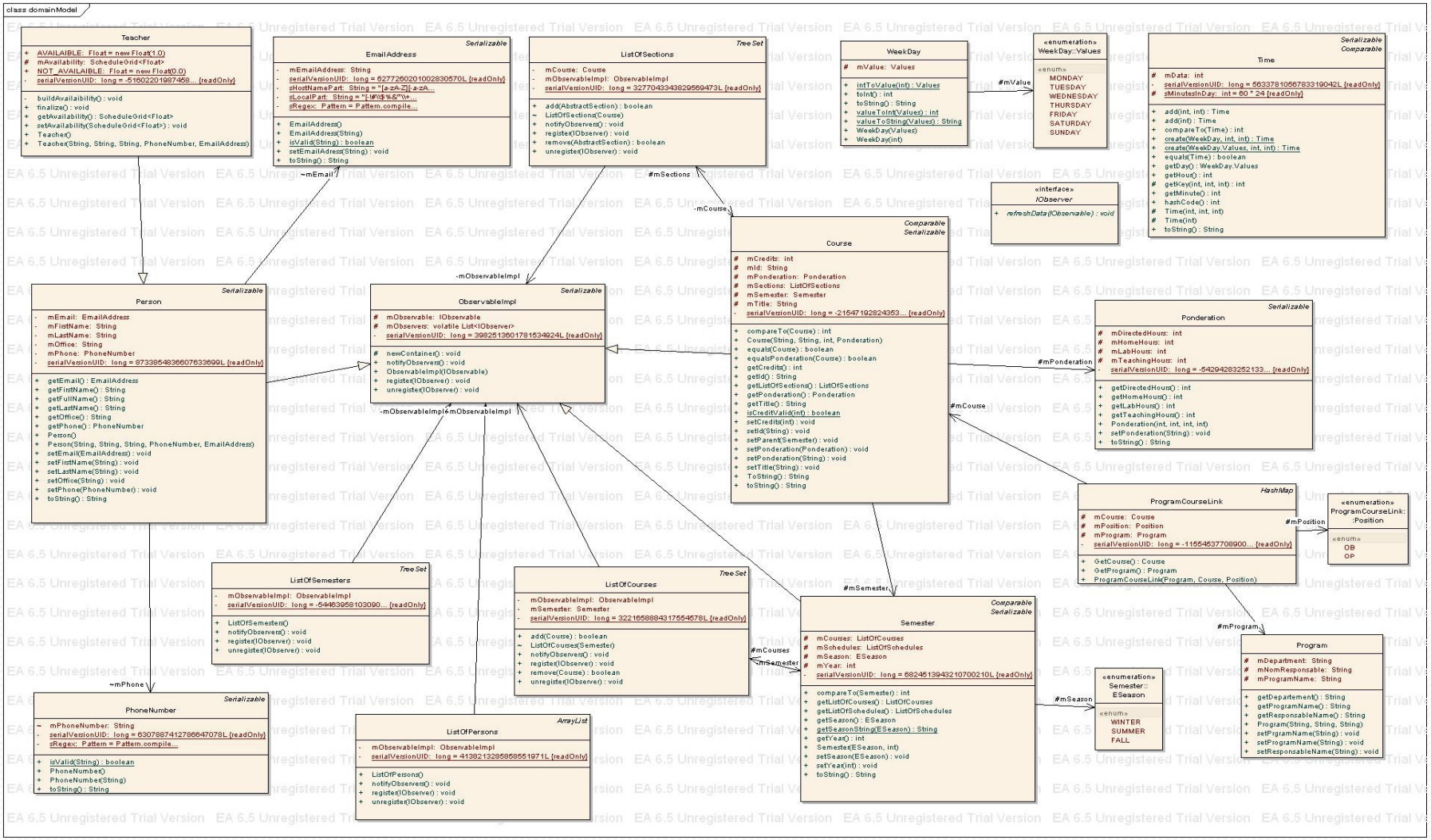
Une bonne architecture

- Partitionnement en sous parties
- Vision globale de la conception
- Vision partagée et comprise des différents intervenants du projet
- Moyens retenus pour atteindre les objectifs de qualité
- Communiquer aux développeurs ce qui doit être concrétisé

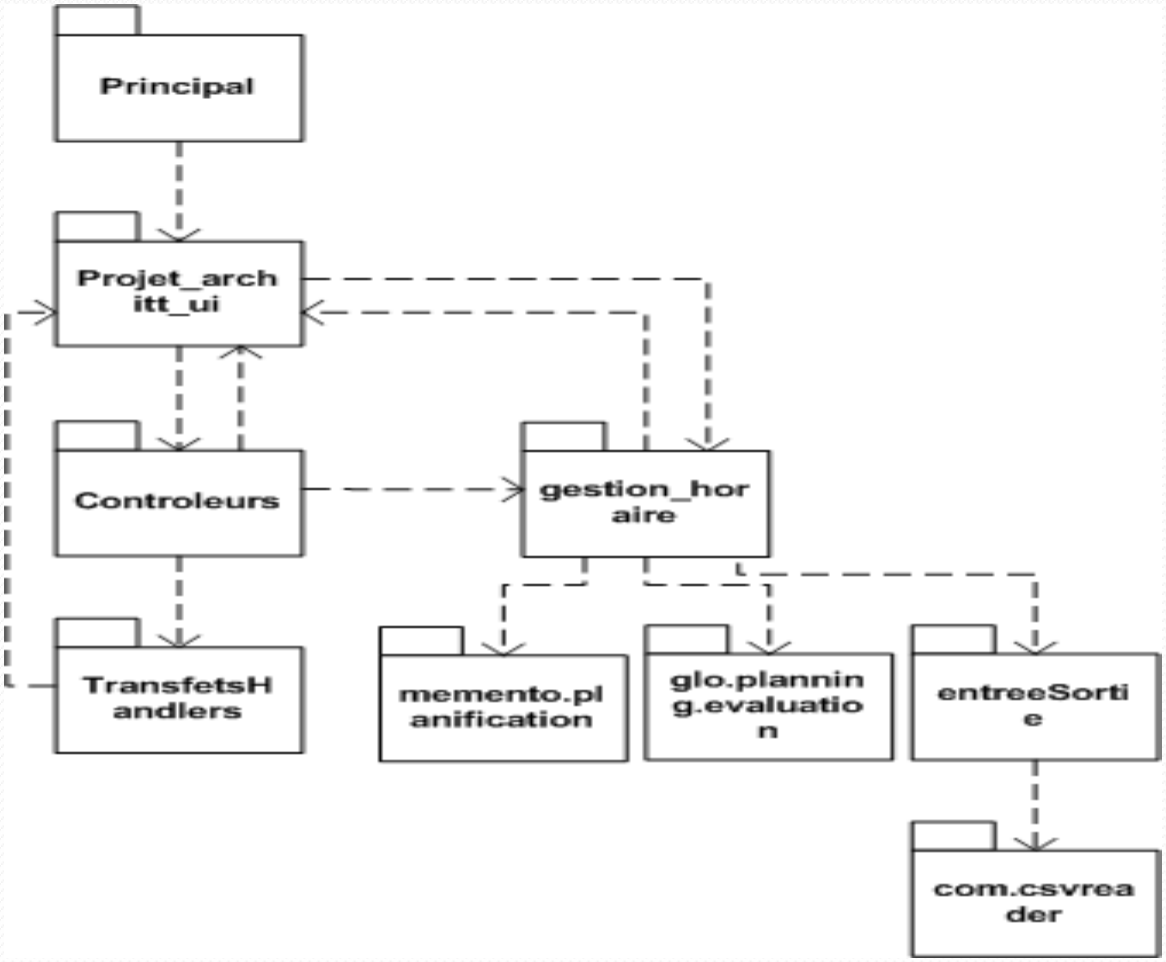
Exemple – Construction d'horaire



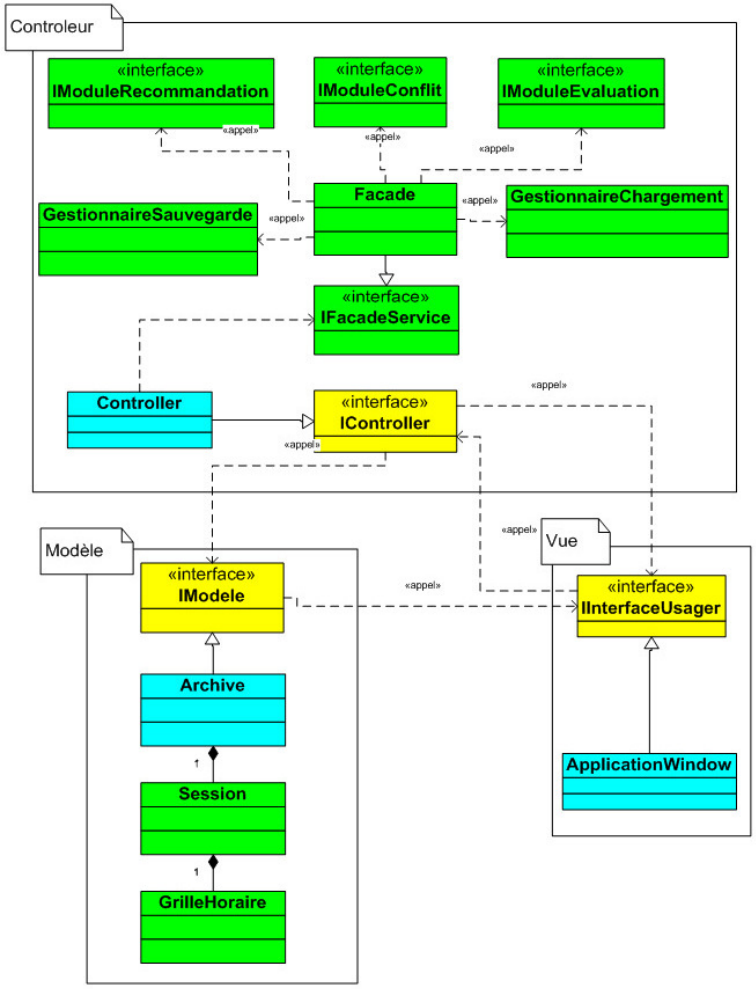
Une architecture ?



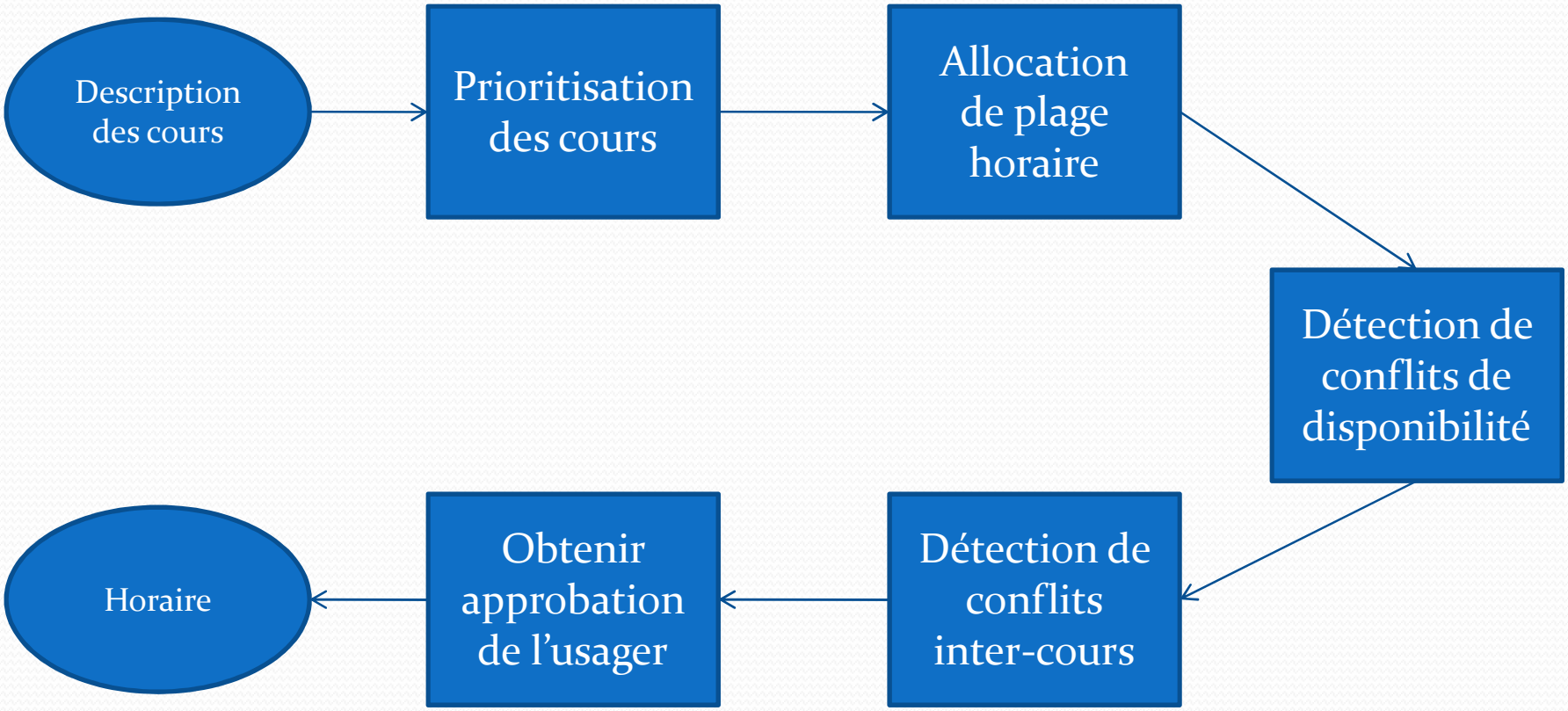
Une architecture ?



Une architecture ?



Une architecture ?



Architecture vs. conception

- L'architecture est une forme de conception
- Les conceptions ne sont pas toutes architecturales
  - Plusieurs détails de bas niveau ne sont pas spécifiés dans une architecture
    - Ex. Le choix d'une structure de données
    - Ex. Le choix d'un algorithme
  - L'architecture ne définit pas une implémentation du logiciel
  - Mise plutôt sur la structure et moins sur le domaine d'application
    - Les fonctionnalités sont orthogonales à la structure

Styles, patrons et idiomes

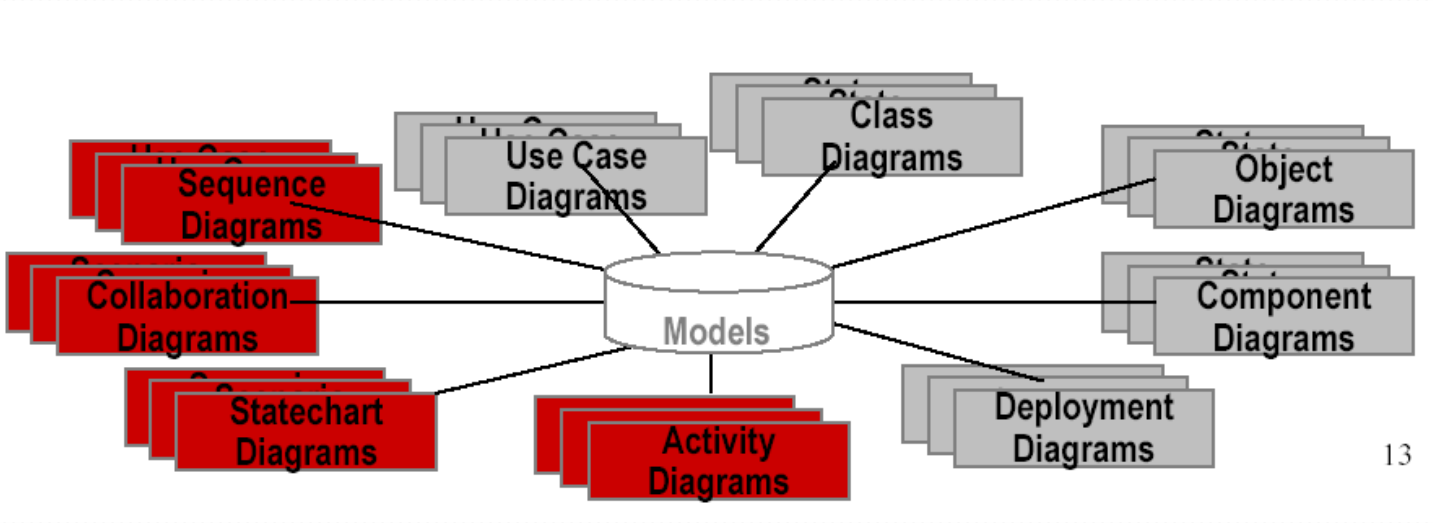
- Termes fréquemment utilisés pour décrire
  - Les structures utilisées pour la conception de logiciel
  - Des recettes éprouvées et reconnues des praticiens
  - Des exemples de bonnes pratiques
    - Par opposition aux mauvaises pratiques (anti-patterns)
  - Des trucs et astuces
  - Des spécificités utiles de certaines technologies

Styles, patrons et idiomes

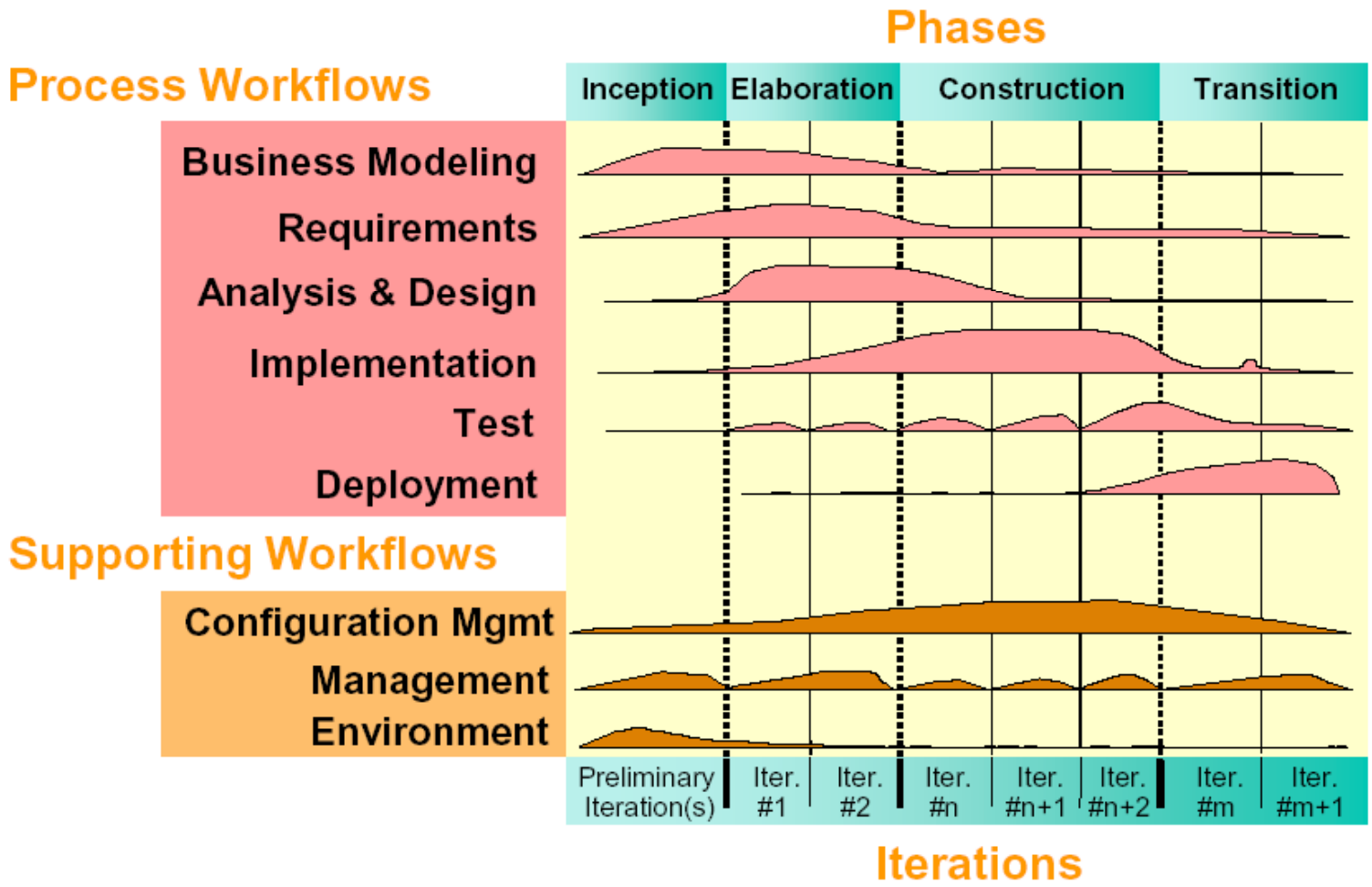
- Styles (*macro patterns*)
  - Structures de haut niveau pour décrire un logiciel d'un point de vue global
  - Ex. Modèle par couche (layered), MVC, …
- Patrons (*micro patterns*)
  - Solutions abstraites applicables à différents contextes mais qui offrent les mêmes avantages chaque fois
  - Correspond aux *Design Patterns*
- Idiomes
  - Décrit de bonnes pratiques liées
    - à un formalisme de programmation
    - à un langage en particulier

Documentation - UML

- Notation préconisée en orienté objet
- Utilise plusieurs diagrammes pour décrire une composante ou un système
- Ces diagrammes permettent de répondre rapidement et de façon standardisée à plusieurs questions



Architecture vs. approche OO



Historique

- Avant 1990 - Les intuitions
  - Terme « Software Engineering »
    - NATO Software Engineering Conference (1968)
  - Travaux de Parnas et Brooks (1972-1975)
    - Développement modulaire: cohésion forte & Couplage faible
    - Logiciel peut être décrit par plusieurs structures
      - Pas uniquement une seule.
  - Travaux de Christopher Alexander (1977)
    - Patrons de conception en architecture
  - Travaux de Beck et Cunningham (1987)
    - Application de patrons en programmation

Historique

- Première moitié 1990 – Les fondements du domaine
  - Travaux sur les *Design Patterns* par le *Gang of Four* (1994)
  - Travaux de Kruchten (1995)
    - 4+1 *Views*
      - *Logical (Functionnal)*
      - *Process (Concurrency)*
      - *Deployment (Physical)*
      - *Data*
      - *Use case (Scenario-based)* (+1)

Historique

- Première moitié 1990 – Les fondements du domaine
  - Travaux de Shaw et Garlan (1996)
    - Identification et catégorisation des styles architecturaux
    - Langage de description de l'architecture (ADL)
- Fin des années 90
  - Plusieurs livres portant principalement les patrons
  - Recommandations par le IEEE
    - Méthode de conception basée sur les attributs de qualité
    - Vues architecturales + Documentation
  - Implication importante du *Software Engineering Institute* (SEI)

Historique (suite)

- Actuellement
  - Reconnaissance du rôle d'architecte logiciel
  - Prolifération de livres (*Software Architecture*)
    - Attention le terme semble parfois galvaudé !
  - Conférences scientifiques
    - WIKSA (IEEE)
    - European Workshop on Software Architecture (EWSA)
    - QoSA
  - Méthodes formelles
  - Langage de description architecturale (ADL)
  - *Reverse engineering*
  - Programmation orientée aspect (AOP)
  - *Service oriented architecture* (SOA)
  - *Model driven architecture…* (MDA)

Conclusion

- L'architecture est une conception de haut niveau d'un logiciel
- Beaucoup de recettes sont déjà disponibles
  - Ne pas réinventer la roue!
- La conception architecturale est un élément important de l'approche orientée objet
- L'architecture repose moins sur les fonctions que sur la qualité du logiciel

La suite…

- Préparation pour le projet de session
- Étude des *designs patterns*
- Points importants
  - Programmation OO et Java
    - Maîtrise des principales notions de Java
  - Familiarité avec UML
    - Utilisé pour décrire les structures
      - classe, séquence, composante, déploiement
  - Éléments de contrôle de qualité
    - Tests unitaires - JUnit