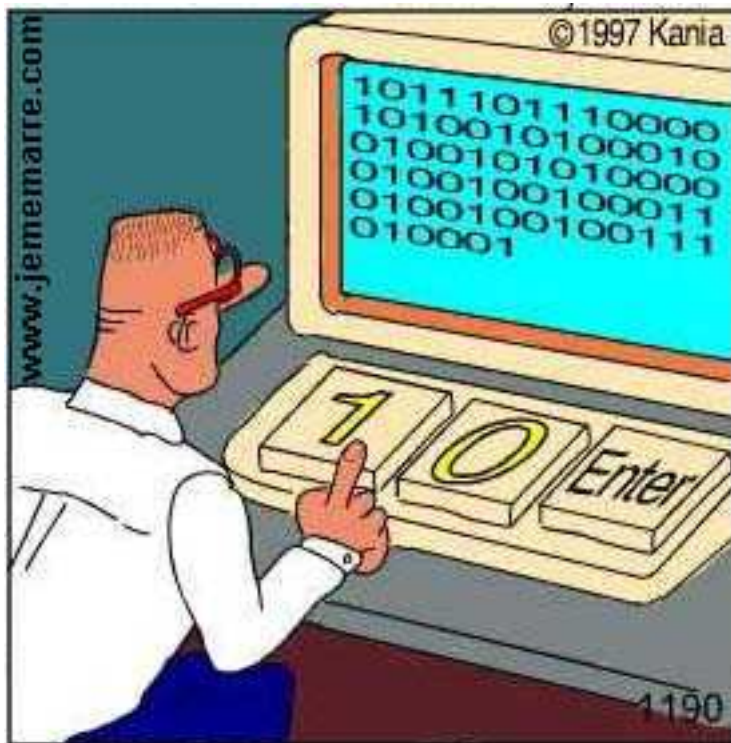

Introduction

Modélisation avec UML

Evolution des langages de programmation



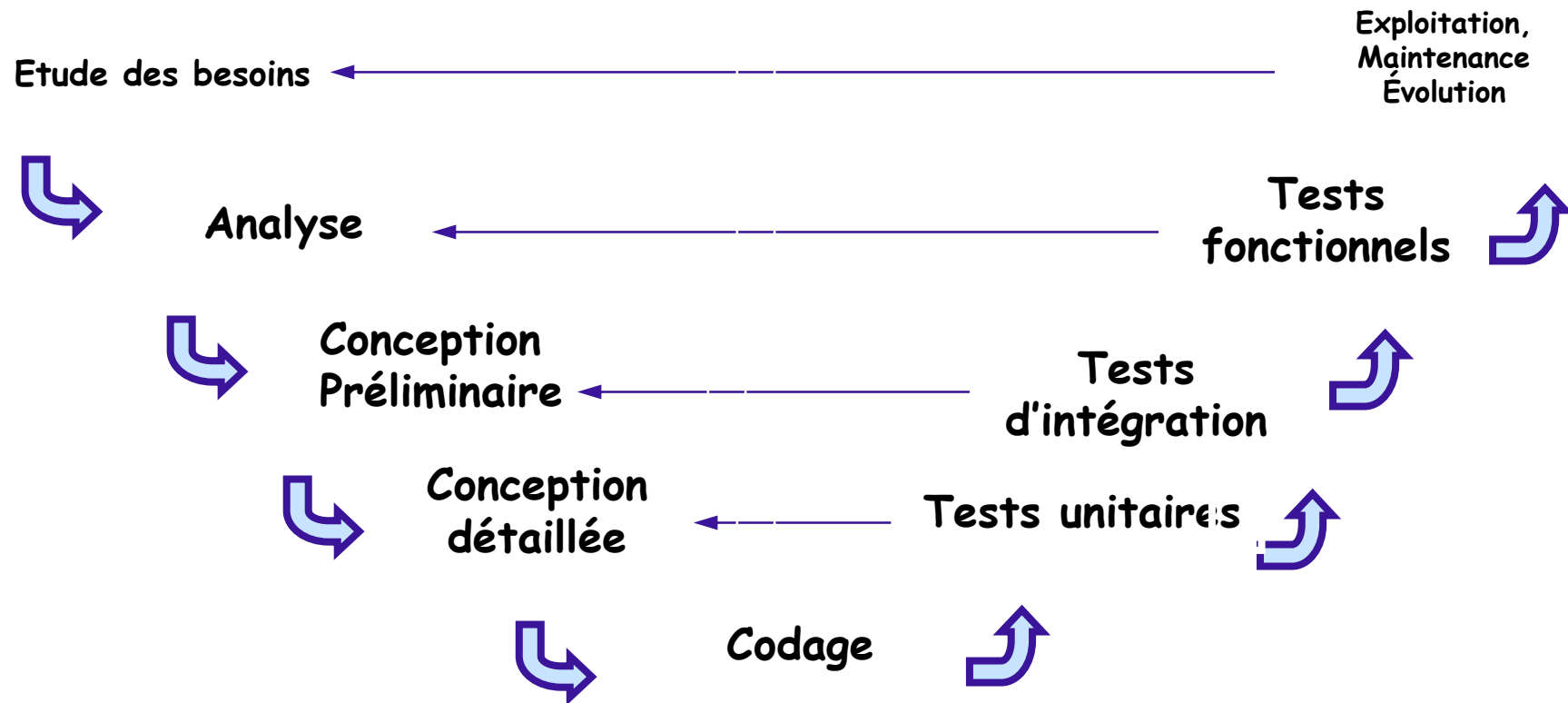
Le vrai programmeur ...

- Langage assembleur
= abstraction de la machine
- Langage impératifs (C, Fortran, ...)
= abstraction du langage assembleur
- Langage objets

Pourquoi l'objet?

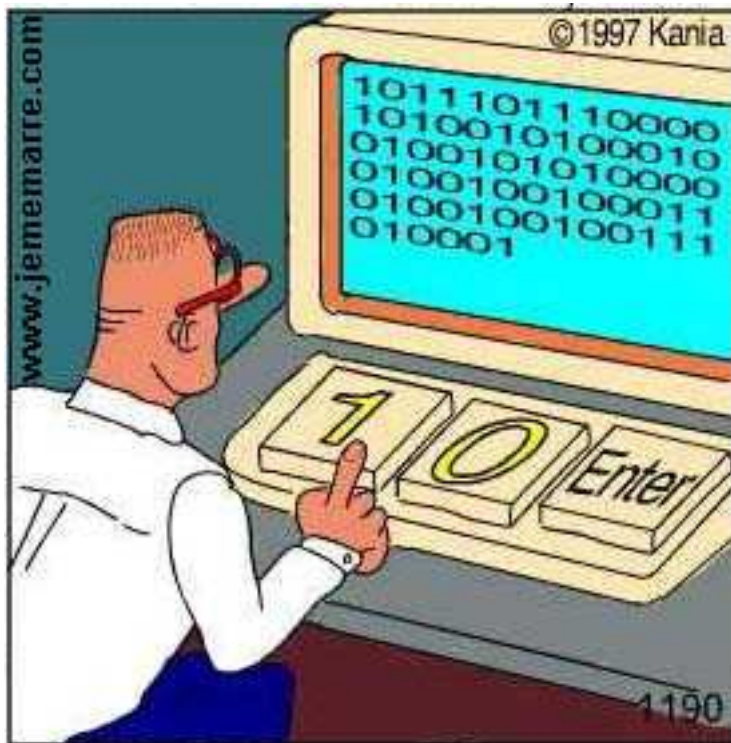
- Réduire le « fossé » entre le monde réel et le monde informatique.
- Construire des applications de plus en plus larges
- Réduire les coûts de maintenance
- Faciliter la réutilisation

Exemple de Cycle de développement du logiciel : le cycle en V



De la POO à l'AOO

Programmation vs Abstraction



Le vrai programmeur ...

- Programmation Orientée Objet (POO)
Smalltalk (70), C++ (85), Java (95)
- Conception Orientée Objet (COO)
Frameworks, Design patterns (95)
- Analyse Orientée Objet (AOO)

Pourquoi modéliser?

- Comprendre le système à informatiser.
« *Un modèle est une simplification de la réalité.* » **Grady BOOCH.**
- Communiquer avec les membres de l'équipe.
- Maîtriser la complexité
- Automatiser la production de logiciel.
 - Documentation.
 - Code.

Unified Modeling Language

UML

- UML est un langage de modélisation (une notation).
- UML n'est pas une méthode
- UML est la notation pour documenter les modèles objets

Processus de développement

Un processus définit **qui fait quoi, quand et comment** dans le but de produire un logiciel de haute qualité en respectant des contraintes de délai, de coûts et de performance.

UML et Processus de développement

- UML ne précise pas le processus de développement.
- Chaque entreprise doit définir son propre processus selon :
 - La taille du système à développer.
 - Les outils disponibles.
 - La culture d'entreprise.
 -

UML et Processus Unifié (UP)

Pour exploiter au mieux UML il faut envisager un processus qui soit :

- Guidé par les cas d'utilisation.
 - Le développement est centré sur les besoins utilisateurs.
- Centré sur l'architecture.
 - L'architecture assure la cohérence du système.
- Itératif et incrémental.
 - Permet de réduire les risques.

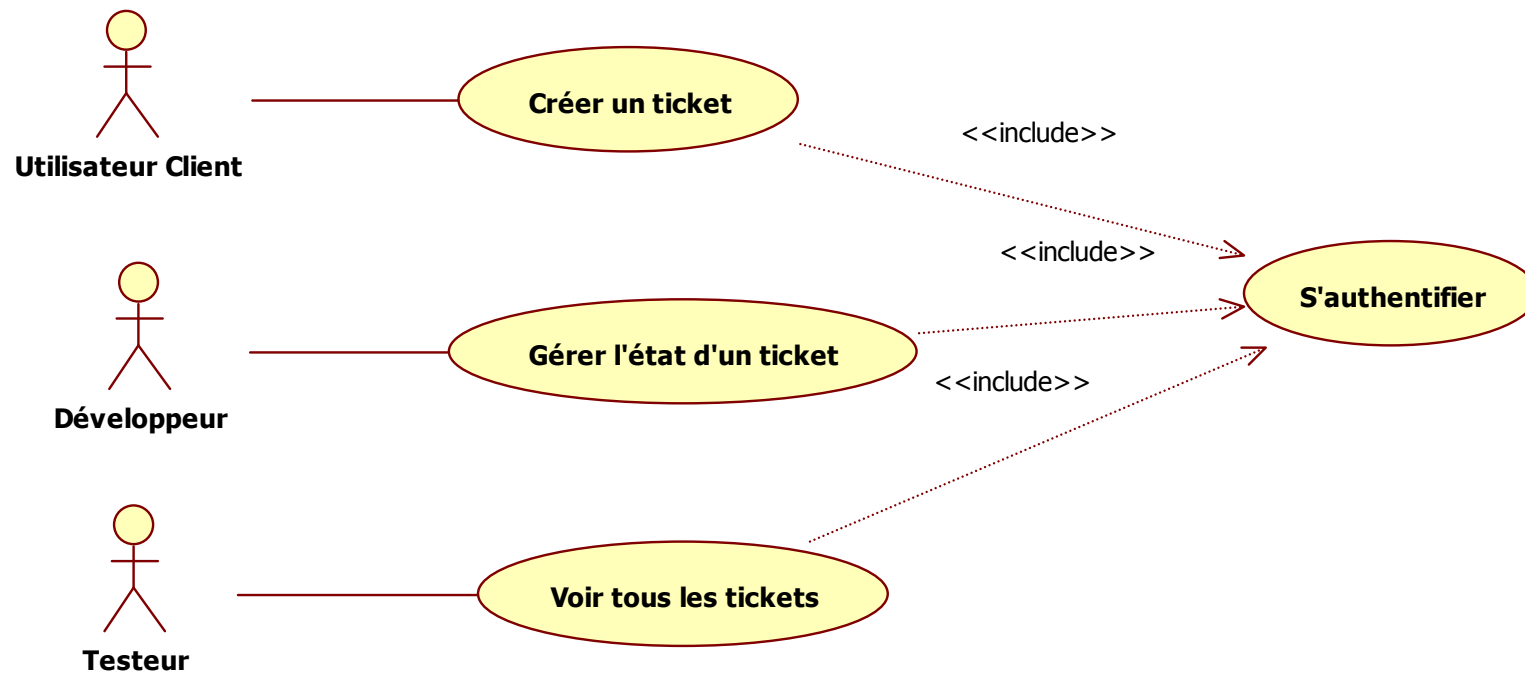
UML et Processus Unifié : RUP

Rational Unified Process

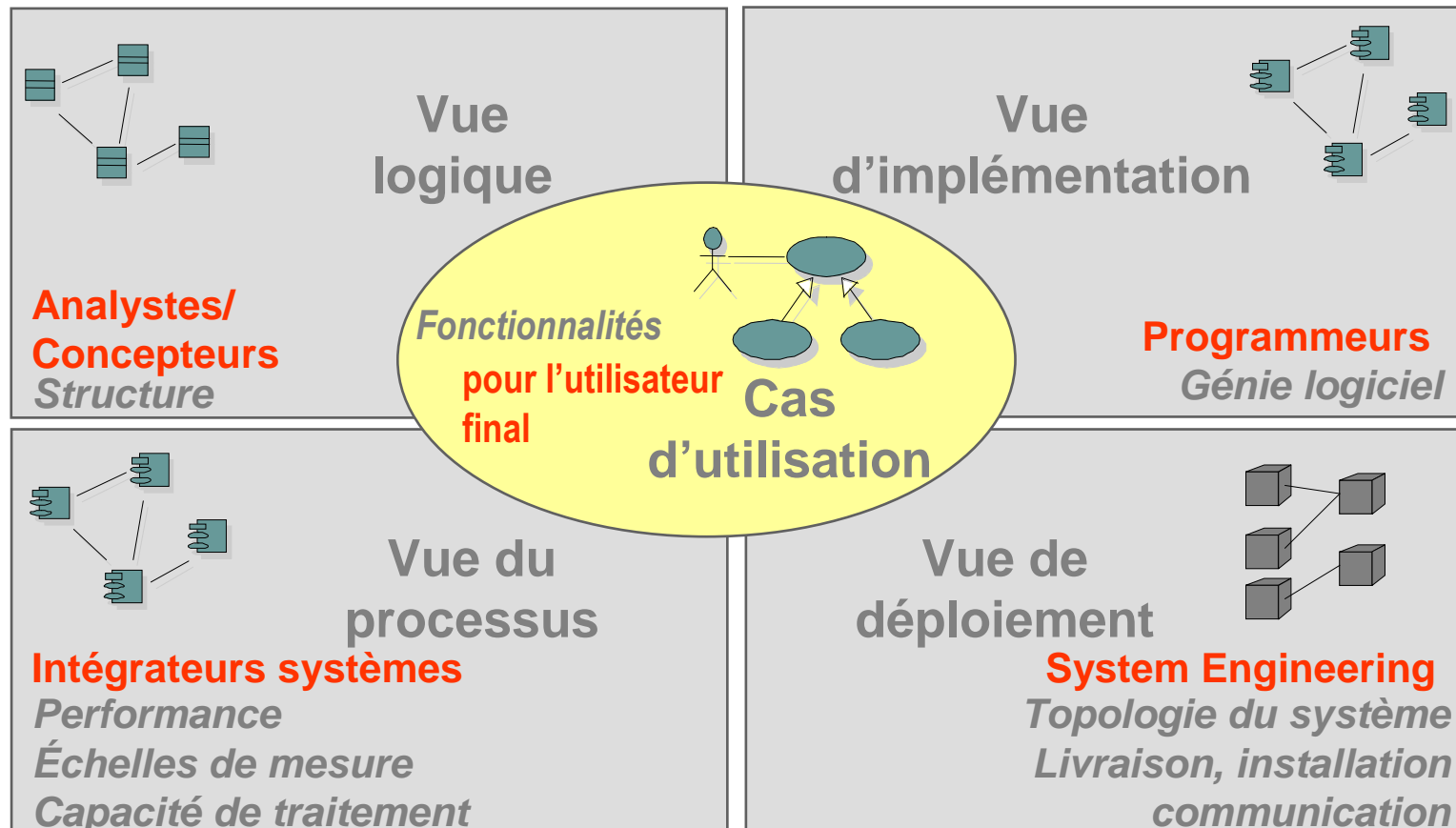
Le Processus Unifié de Rational est :

- Guidé par les cas d'utilisation (CU).
 - Le développement est centré sur les besoins utilisateurs.
- Centré sur l'architecture.
 - L'architecture assure la cohérence du système.
- Itératif et incrémental.
 - Permet de réduire les risques.

RUP est guidé par les CU



RUP est centré sur l'Architecture



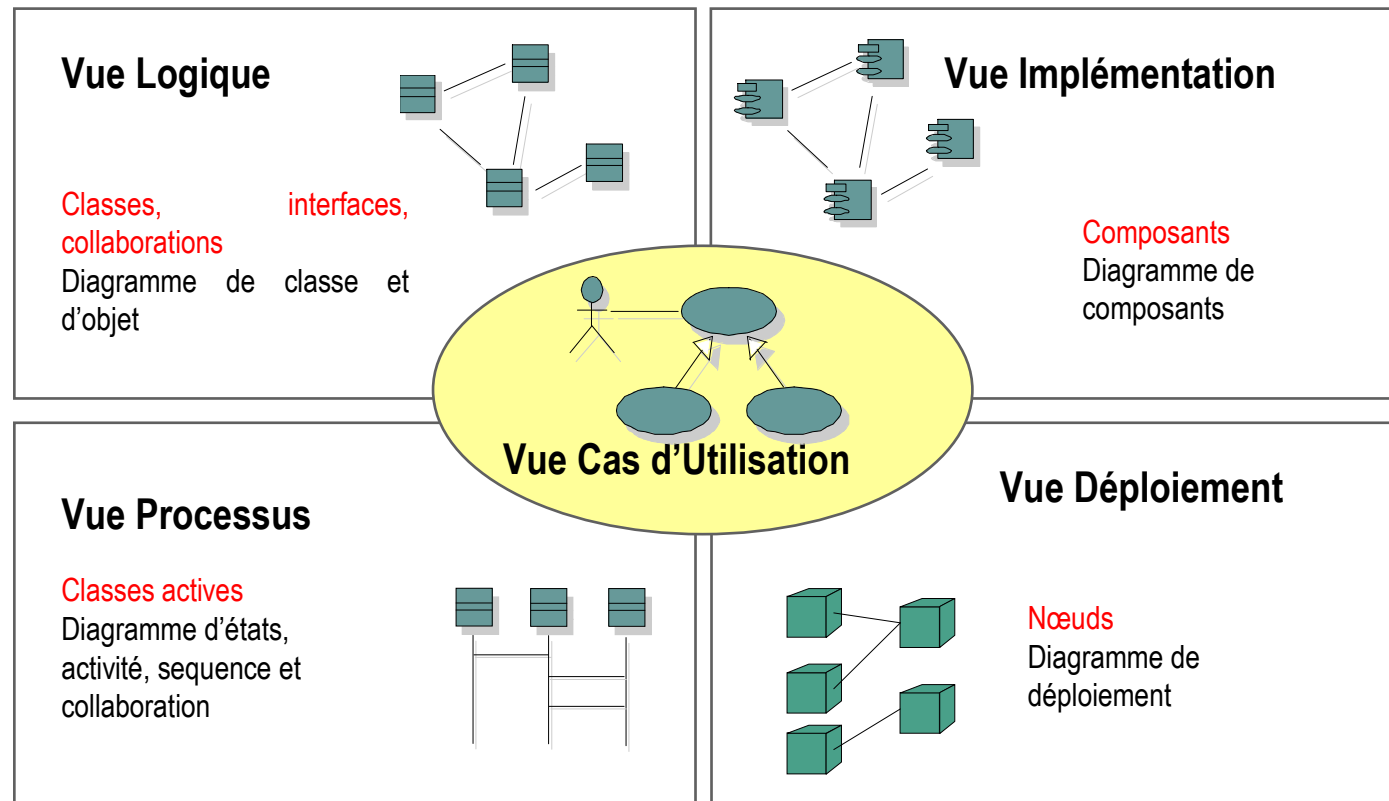
Architecture RUP - Modèle des 4 +1 vues

Une vue de l'architecture est la description d'un système d'un point de vue particulier, couvrant certains points et en omettant certains autres.

Le travail d'un architecte est de prendre tous ces différents points de vue et de réaliser l'architecture adéquate satisfaisant tous les intervenants (clients, analystes, concepteurs, développeurs, intégrateurs, chefs de projets ...).

Chacune de ces cinq vues est une description simplifiée d'un système à partir d'un angle différent.

4+1 vues et diagrammes UML



Modèle des 4 +1 vues

La **vue des cas d'utilisation** contient les scénarios principaux qui sont utilisés pour faire fonctionner l'architecture et pour la valider.

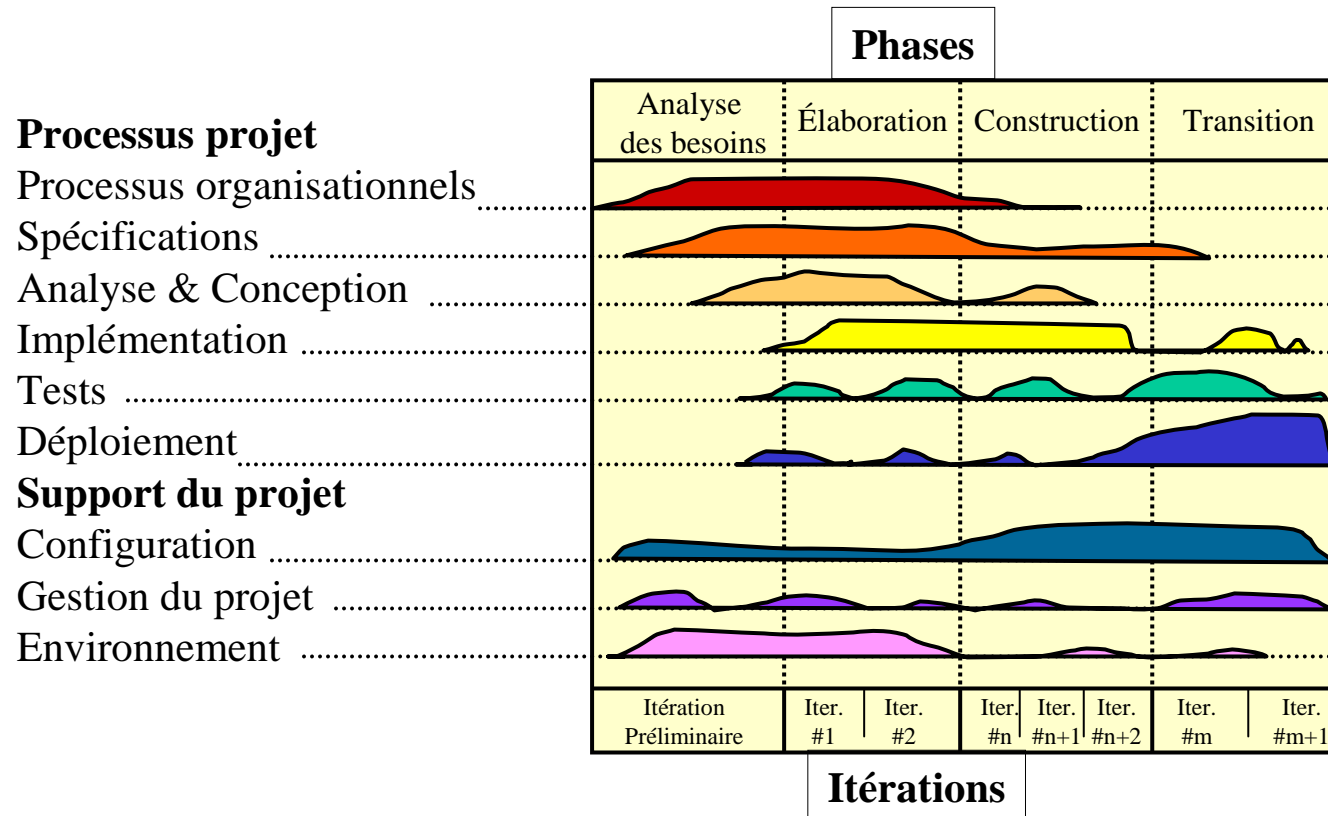
La **vue logique** identifie la plupart des paquetages, sous-systèmes et classes.

La **vue d'implémentation** décrit l'organisation des modules du logiciel.

La **vue des processus** concerne les aspects concurrents du système à l'exécution: tâches, threads et leur interaction.

La **vue de déploiement** décrit les ressources matérielles et la répartition du logiciel dans ces ressources.

RUP est itératif et incrémental



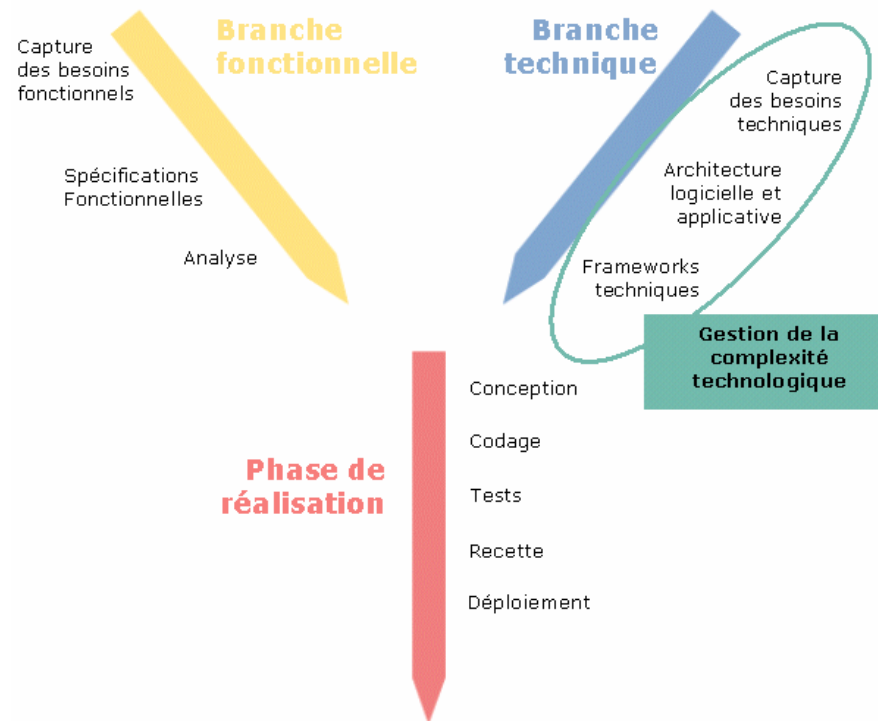
Processus Unifié

RUP

- Le RUP est à la fois une méthodologie et un outil prêt à l'emploi (référentiel Web)
- Cible des projets de plus de 10 personnes
- Points forts
 - Spécifie le dialogue entre les différents intervenants du projet : les livrables, les plannings, les prototypes...
 - Propose des modèles de documents, et des canevas pour des projets types
- Points faibles
 - Coûteux à personnaliser : batterie de consultants
 - Très axé processus, au détriment du développement

Processus Unifié : 2TUP

2 Track Unified Process



Processus Unifié

2TUP

- Détaillé dans «UML en action», Pascal Roques (Eyrolles)
- Cible des projets de toutes tailles
- Points forts
 - Fait une large place à la technologie et à la gestion du risque
 - Définit les profils des intervenants, les livrables, les plannings, les prototypes
- Points faibles
 - Ne propose pas de documents types du développement

Processus Unifié

Agile Unified Process?

Version simplifiée du RUP!

It describes a simple, easy to understand approach to developing business application software using agile techniques and concepts yet still remaining true to the RUP

Processus Unifié

Arrington

- Processus Unifié simplifié décrit dans «Enterprise Java with UML», C.T. Arrington (OMG Press).
- Documente les phases suivantes :
 - Modélisation des besoins (Gathering requirements)
 - Analyse orientée objet
 - Choix de technologie
 - Architecture
 - Conception