

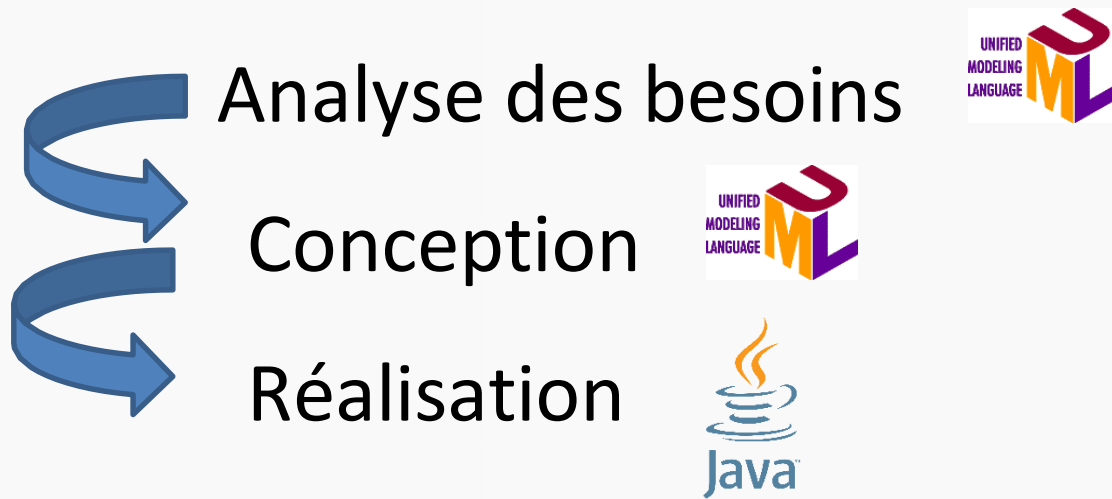
R2-01b

Bases de la conception orientée objet

Partie 6

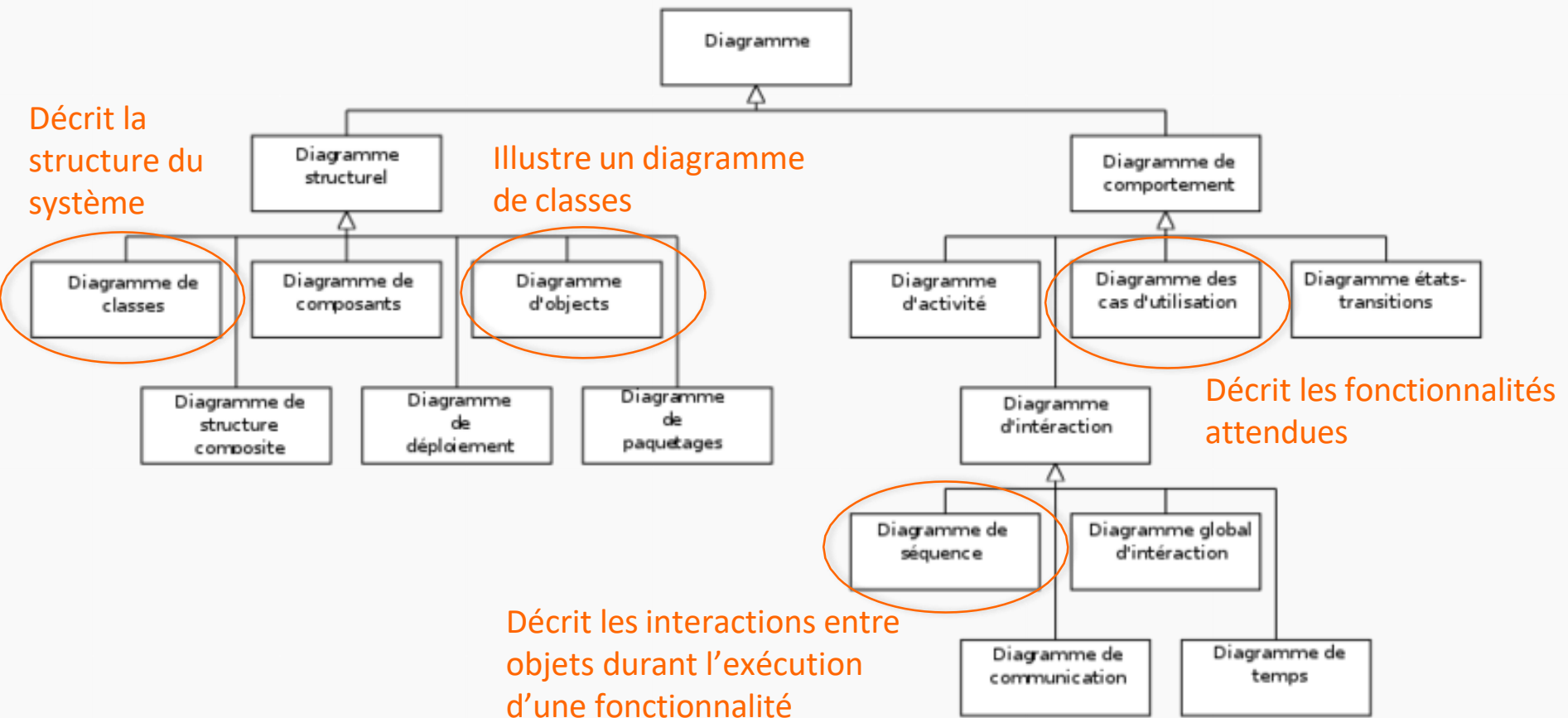
Démarche de développement

Première approche de démarche de développement logiciel



Unified Modeling Language (UML)

Les diagrammes indispensables



Pour chaque diagramme, on choisit le niveau de détail en fonction de l'étape du processus de développement

Les multi-vues dans UML

UML propose un ensemble de diagrammes pour représenter un système :

- **Vue fonctionnelle**

Besoins fonctionnels des clients (**diagrammes de cas d'utilisation**)

- **Vue structurelle**

Aspects statiques, structure du système (**diagramme de classes**)

- **Vue comportementale**

Aspects dynamiques (**diagrammes de séquences**)

Ces multi-vues favorisent une réflexion pertinente car il est possible, dans le cadre d'une vue, de penser à des détails non facilement perceptibles dans une autre vue.

→ **cohérence** des différentes vues nécessaire pour compléter la réflexion.

Une idée de la démarche

1) Partir des fonctionnalités **attendues** du système (expression des besoins) :

- ce que le système doit faire du point de vue des utilisateurs

2) Puis "affiner " pas à pas, par étapes :

- Analyse (formalisation des besoins)
- Conception (description du "comment")
- Réalisation (traduction dans langage de programmation)

Partir des besoins ...

Une application de gestion de groupes d'étudiants.

- **Besoins (fonctionnalités attendues) :**

- Créer un nouveau groupe
- Inscrire un nouvel étudiant en l'affectant à un groupe
- Changer un étudiant de groupe
- Connaître le groupe d'un étudiant
- Connaître tous les étudiants d'un groupe

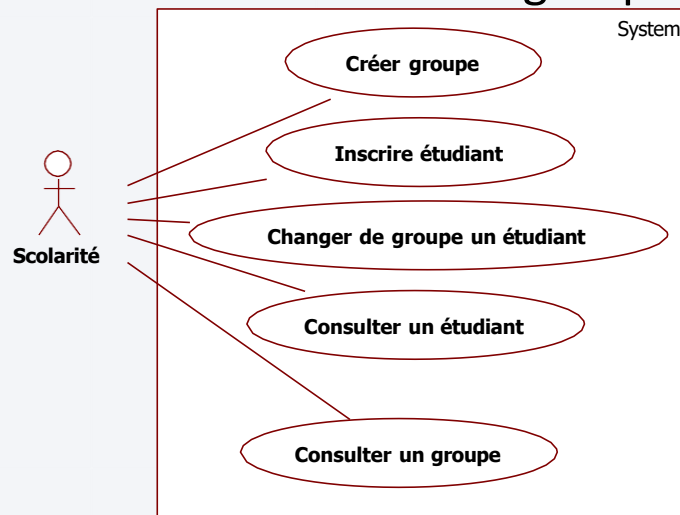
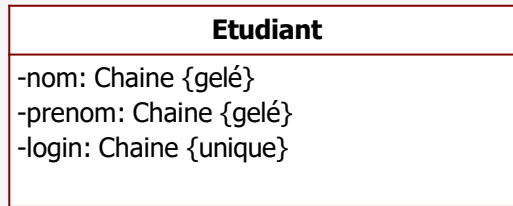


Diagramme des cas d'utilisation

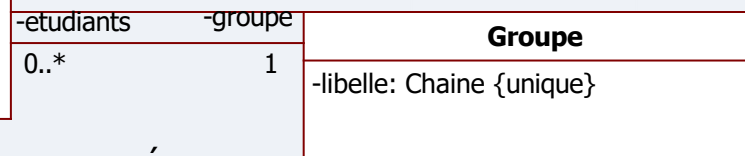
ETAPE 1

Diagramme des cas d'utilisation
+ Pour chaque cas d'utilisation :
préconditions, résumé,
postconditions

Partir des besoins ...



Tout étudiant doit être inscrit dans un groupe



On ne doit pas pouvoir changer les nom, prénom et login d'un étudiant, ni le libellé d'un groupe.

Diagramme de classes

ETAPE 2

Diagramme de classes sans opérations
+ Diagramme de séquences haut niveau

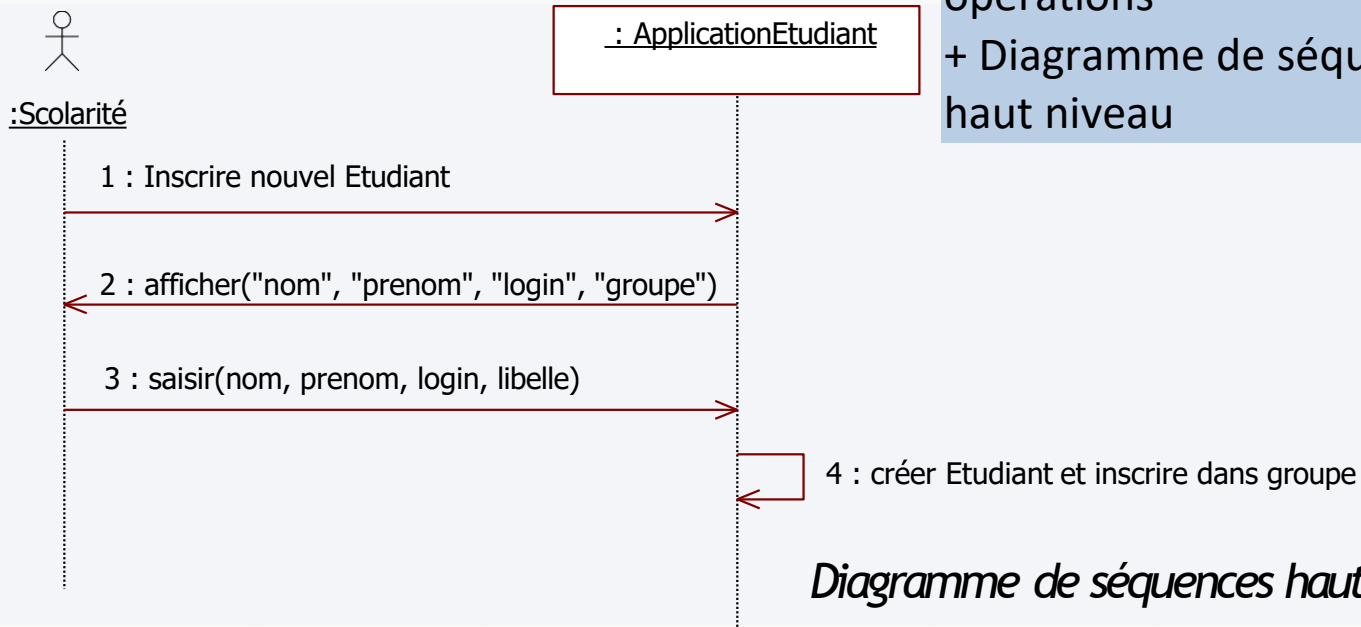


Diagramme de séquences haut niveau

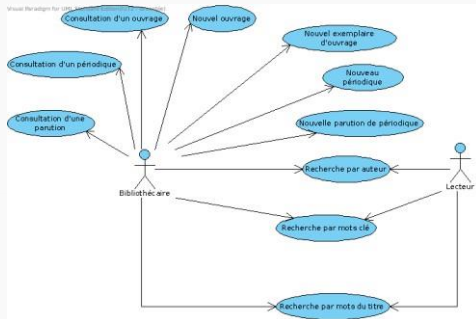
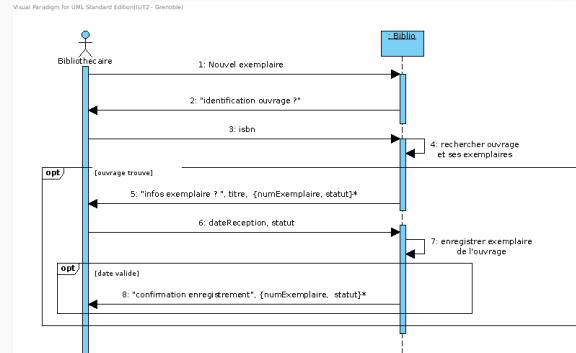
Analyser

ETAPE 3

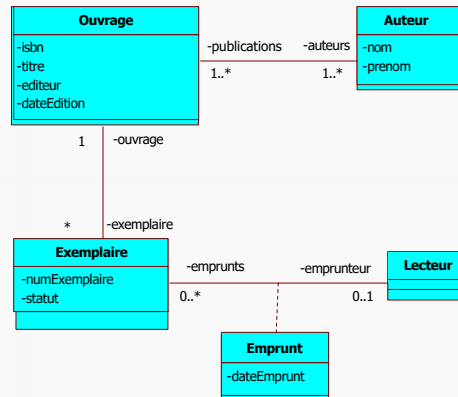
Assurer la cohérence des 3 diagrammes et vérifier que les besoins sont couverts

+ maquette IHM

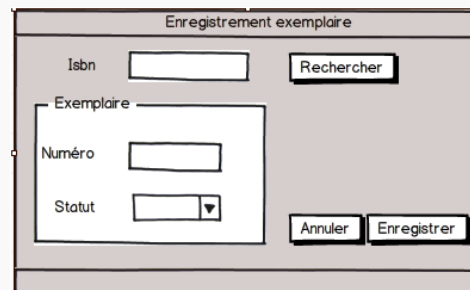
Diagrammes de séquences
description des scénarios sous
forme de diagramme de
séquences système (acteurs /
système cible)



Cas d'utilisation
diagramme et description des
fonctionnalités attendues



Diagrammes de classes capture des
concepts du domaine (les classes,
attributs, associations)



Maquette de l'IHM

Concevoir ...

ETAPE 4

Compléter diagramme de classes avec opérations
+ détailler diagramme de séquences

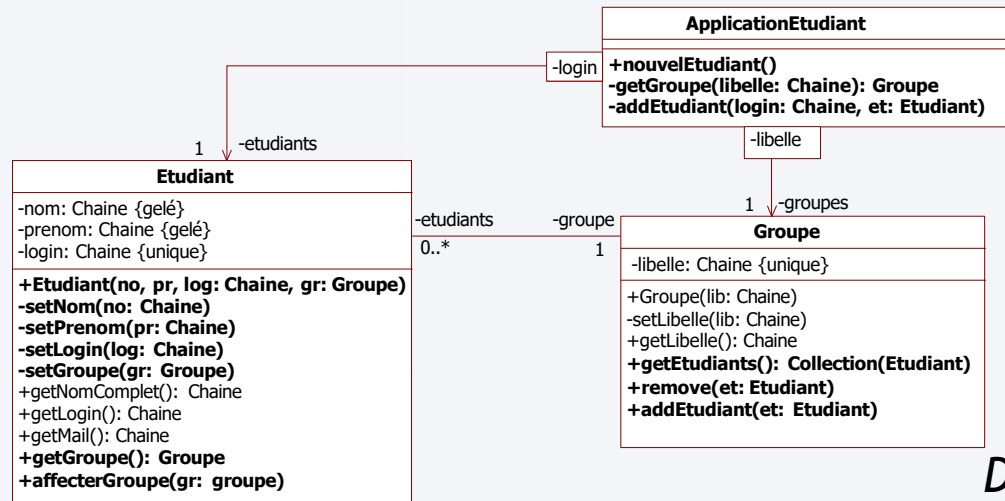


Diagramme de classes

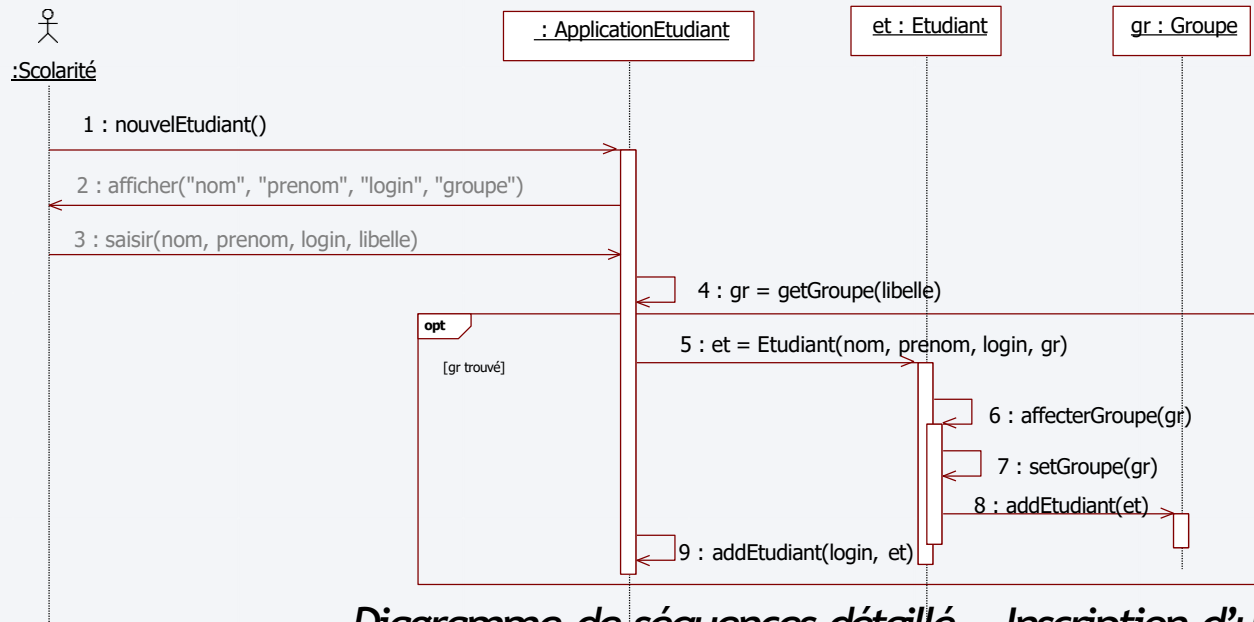


Diagramme de séquences détaillé - Inscription d'un étudiant

Coder ...

```

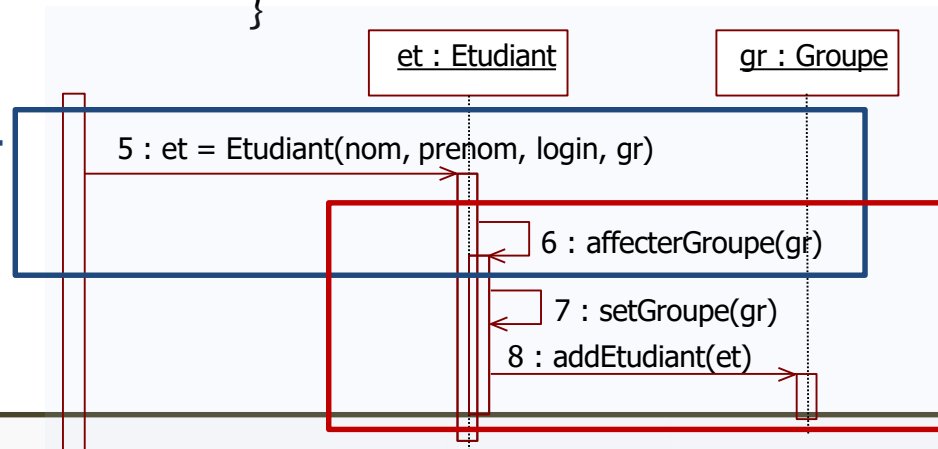
public class Etudiant {
    private String nom ; // prenom, login
    private Groupe groupe ;
    public Etudiant (String no, ..., Groupe gr) {
        this.setNom(no); ... ;
        this.affecterGroupe(gr);
    }
    private void setNom(String no) {
        nom = no;
    }
    ....
    private void setGroupe(Groupe gr) {
        groupe = gr;
    }
    public Groupe getGroupe() {
        return groupe ;
    }
}

```

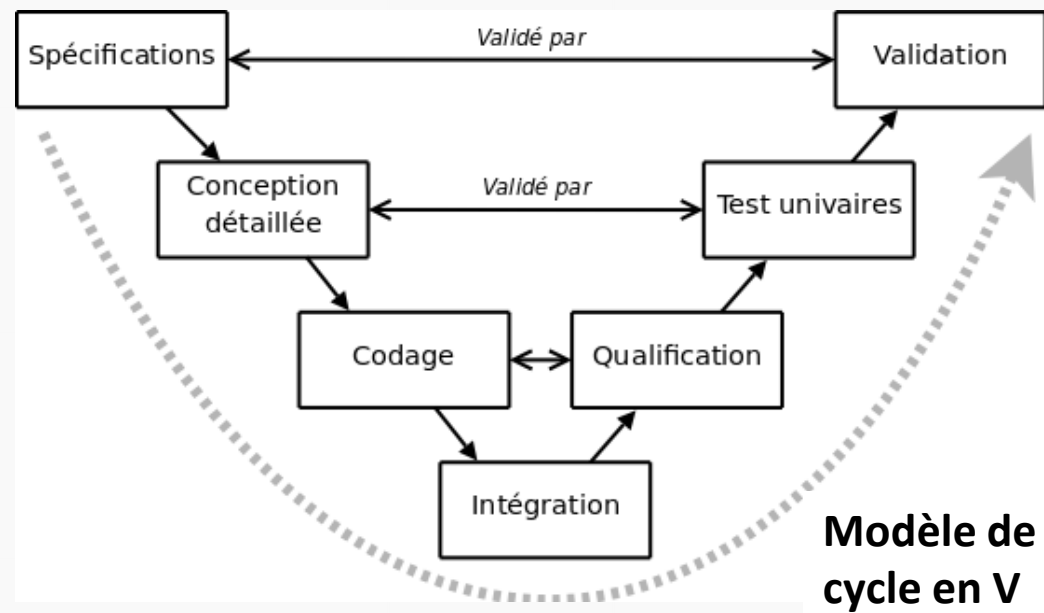
```

private void affecterGroupe (Groupe gr) {
    if (gr != null) {
        if (! gr.getEtudiants().contains(this)) {
            if ( this.getGroupe() != null) {
                this.getGroupe().remove(this);
            }
            this.setGroupe(gr);
            this.getGroupe(). addEtudiant(this);
        }
    }
}

```

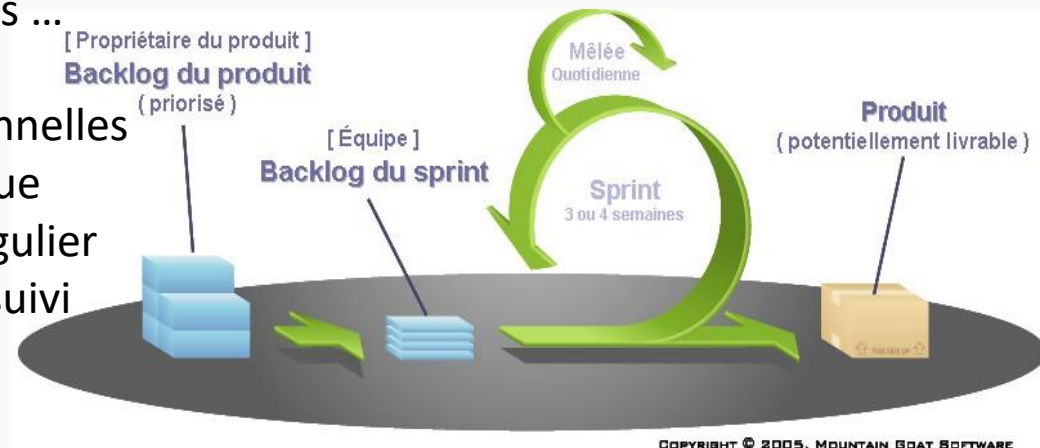


Combiner rigueur et agile



Méthode agile - quatre principes fondateurs

- Priorité aux personnes et aux interactions ... plutôt qu'aux processus et outils
- Des applications fonctionnelles opérationnelles ... plutôt qu'une documentation pléthorique
- Collaboration avec le client, feedback régulier
- Réactivité au changement ... plutôt que suivi d'un plan strict



Un compromis :

→ **Modèle suffisamment précis, cohérent et détaillé MAIS aussi simple que possible**