



# Projet long JAVA

## UML 7

[REDACTED], E. Lumet, [REDACTED]



# Sommaire

## Projet UML 7

- Démonstration
- SCRUM
  - Vision produit
  - Objectifs
  - DoD
  - Vélocité et valeur métier
  - Points positifs
- Conception et réalisation
  - Architecture
  - Choix de conception
  - Réalisation

# Démonstration UML 7

- Démonstration
- SCRUM
  - Vision produit
  - Objectifs
  - DoD
  - Vitesse et valeur métier
  - Points positifs
- Conception & réalisation
  - Architecture
  - Choix de conception
  - Réalisation

# SCRUM

## *Itération 3*

- Démonstration
- SCRUM
  - Vision produit
  - Objectifs
  - DoD
  - Vitesse et valeur métier
  - Points positifs
- Conception et réalisation
  - Architecture
  - Choix de conception
  - Réalisation



# SCRUM

## Vision produit

- Outils

Notre produit est un outil s'intégrant dans un projet lors de la phase de conception qui permet de modéliser et générer une base de code.

- Utilisateurs

- Développeur
- Concepteur système

- Valeur métier

Gain de temps et automatisation



# SCRUM

## Objectifs

1. Sauvegarder et charger un diagramme UML
2. Créer des éléments via l'interface graphique
3. Traiter les cas d'exception
4. Éditer les éléments UML



# SCRUM

## Definition of Done (DoD)

- Javadoc en anglais (à l'exception des tests unitaires)
- Respect des standards de programmation
- Tests unitaires JUnit 4 (partie non graphique)
- Validation manuelle par deux autres personnes
  - Partie graphique
- Règles GIT respectées (cf. [guide de référence](#))
- Rapports d'itération complets ( cf. livrables [cregut.perso.enseeiht.fr](http://cregut.perso.enseeiht.fr))
- Diagrammes de classe complétés au fur-et-à-mesure
- Validation Sonarqube



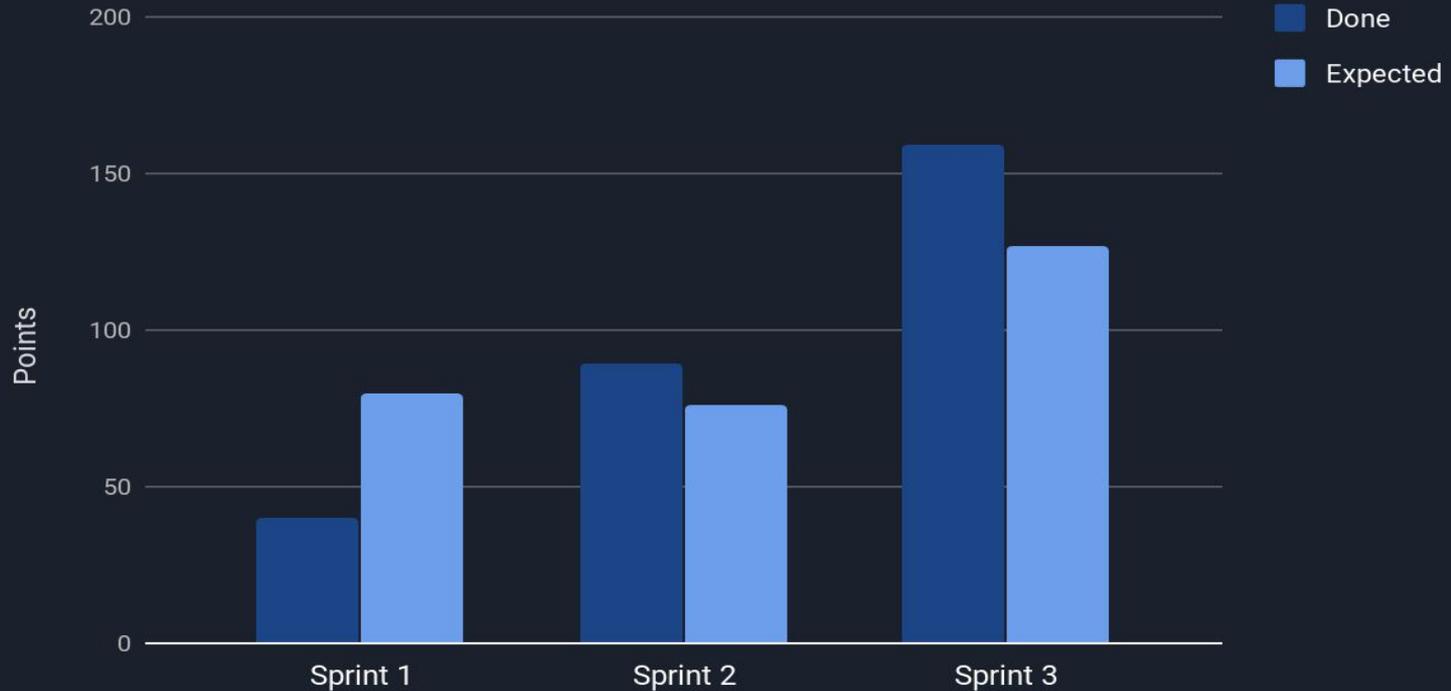
# SCRUM

## Vélocité et valeur métier

- **Vélocité**
  - Nombre d'US effectuées : **24**
    - Critères d'acceptation validés
  - Nombre d'US total dans le sprint : **24**
  - Point d'effort : **159 / 127**
- **Valeur métier : 1800**
  - MUST : 200
  - SHOULD : 150
  - COULD : 100
  - WOULD : 50

# SCRUM

## Graphe vélocité





# SCRUM

## Points positifs

- Prise en compte des *User Stories* techniques
- Plus d'écoute dans le groupe
- Objectifs fixés
- 1 *User Story* = 1 personne
- Séparation en 2 groupes :
  - Groupe "expert" : les DUT Info
  - Groupe "novice" : les DUT R&T
- Tests JUnit 4 effectués

# Conception et réalisation

- Démonstration
- SCRUM
  - Vision produit
  - Objectifs
  - DoD
  - Vitesse et valeur métier
  - Points positifs
- Conception et réalisation
  - Architecture
  - Choix de conception
  - Réalisation



# Conception et réalisation Architecture

- Diagramme modèle
- Diagramme vue
- Diagramme contrôleur



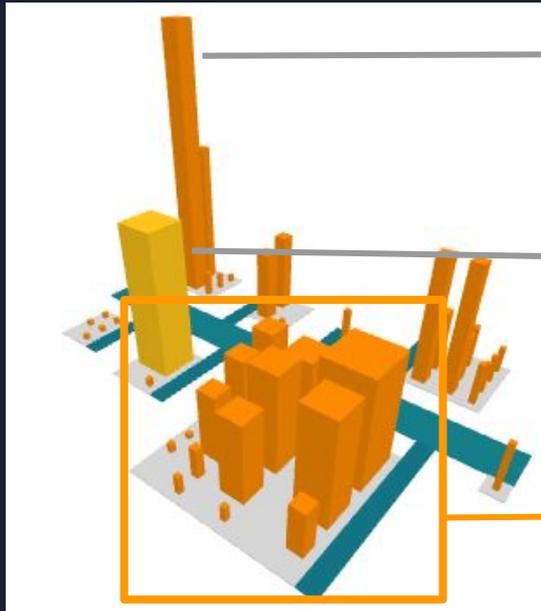
# Conception et réalisation

## Choix de conception

- MVC actif
- Patron visiteur pour la génération de code
- Librairie de fonctions statiques (éviter la duplication de code)
- Utilisation de Swing
  - Vu en cours
  - Connu par les membres du groupe
- Utilisation de JGoodies (pour les formulaires)
- Exceptions contrôlées

# Conception et réalisation

## SonarQube : *Visualisation 3D*



### Contrôleur pour l'édition des méthodes

- Peu complexe
- Le code est très long... (IHM)

### Générateur Java

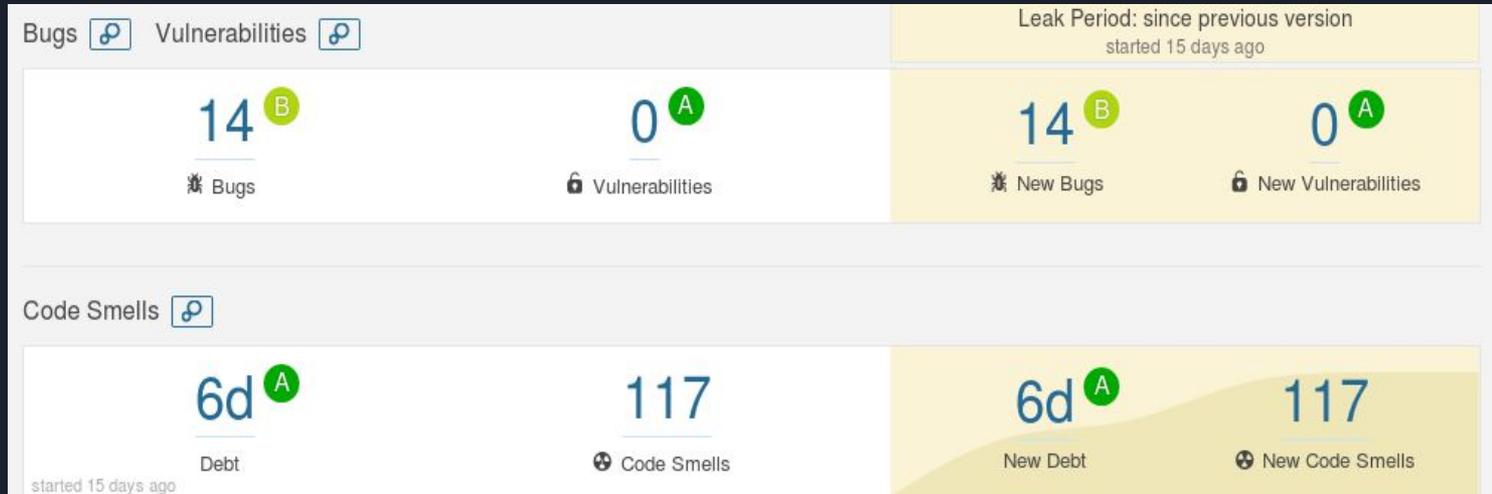
- Complexité moyenne
- Taille en terme de code convenable

### Quartier homogène dans l'ensemble

- Représente le package modèle
- **UmlInterface** est complexe
  - nécessite probablement des améliorations.

# Conception et réalisation

## SonarQube : *Quality Gate*





# Concept et réalisation

## Réalisation : Problèmes et solutions

- Problème de la gestion des relations
  - GridBagLayout (complexe à mettre en oeuvre)
- Estimation de la complexité
- Développement de l'interface utilisateur
  - Usage d'un générateur de code afin d'accélérer le développement de l'interface utilisateur