# Module Génie Logiciel / Conception / Méthodes Agiles

## Chapitre 0 – Présentation du Module


**Véronique DESLANDRES** - Licence DEVOPS, IUT de LYON

# Présentation du module

- 28h, une évaluation finale, du CC
- **Courant DevOps / Génie Logiciel** : 2h
  - Cours / TD
- **Introduction aux Méthodes Agiles** : 12h
  - Scrum : Cours / ateliers
  - Quizz Vidéo
- **Conception et Pattern** : 14h
  - **Conception** de code : UML / SOLID / Design Pattern
  - Conception **d'architecture de MS**
  - Cours, TP : UML et Java

# L’évaluation

- **Examen final**
  - 2/3 de la note
  - Documents autorisés : corrigés de TD
- **Contrôle continu**
  - 1/3 de la note
  - Un quizz
  - Participation en cours

# POSITIONNEMENT DU MODULE
# Les Pré requis

- *Théoriquement vues en Bac+2*
- Utiliser UML pour modéliser
  - Interpréter un diagramme UML donné
    - Diag de Classes, des Cas d'Utilisation, Diag d'Activité, Diag de Séquences
  - Concevoir un diagramme UML modélisant un aspect d'un objet d'étude
  - Vérifier la cohérence de différents diagrammes modélisant un même objet d'étude
    - Entre 2 ou 3 diagrammes

# Compétences en Conception de niveau Bac+3

- Concevoir **l'architecture** logicielle
  - Architecture de **microservices**
  - Modéliser **l'objet d'étude** en diagrammes UML
  - Structurer un logiciel en paquetages et classes **faiblement couplés** et à **forte cohésion**
  - Respecter les **principes SOLID**
  - **Design Patterns**

➔ *Dans ce module*

*et dans les modules de Développement du s2*

# Compétences Méthodologies

- Mettre en œuvre une **méthodologie** pour concevoir, réaliser et maintenir des logiciels de qualité
  - **Découvrir Scrum / les méthodes agiles**
  - **Découvrir les principes de l'architecture hexagonale**

➔ Dans *ce module + module Gestion Projet (pour Scrum)*

# Compétences Techniques

- Comment implémenter de « **bons logiciels** » ?
- Mettre en œuvre **à bon escient** les mécanismes offerts par les langages de POO :
  - héritage, généricité, surcharge, polymorphisme, …

  ➔ Pratiques de conception, d'écriture de code, de tests, liées au courant DEVOPS

- *Vues dans les Modules :*
  - *Java, .net et J2EE*
  - *ANSIBLE, Conteneurisation*
  - *ce module (volet Qualité Logicielle)*

# Vous avez dit Pédagogie ?

- **Classe inversée** : plus qu'une inversion d'agenda (travailler le cours à la maison, et faire des activités en cours)
- Il s'agit de **construire collectivement** le cours

- Votre travail :
  - **Lire** les tutoriels / regarder les **vidéos**
  - Transmettre en cours les points qui restent flous ou mal compris
- Mon travail :
  - Vous rendre **acteurs** de votre apprentissage
  - Répondre à **vos** besoins d'apprentissage

# Pédagogie active (2)

- En modélisation, il y a souvent **plusieurs solutions** possibles à un même problème

- Mais certains choix sont **faux** (ou mauvais) : c'est à partir de ceux-là que vous apprendrez le plus
  - Droit à l'erreur

# Supports

- Mes supports sont des présentations de cours, en séance, assez volumineux...
  - Mis à disposition sur ma page web du labo :

    https://go.univ-lyon1.fr/pageevps
  - Ou chercher : DESLANDRES LIRIS → page Enseignement

- Je peux proposer des poly (textes) d'autres auteurs
  - Vous pouvez aussi trouver ceux qui vous conviennent

# Présentation réciproque

- **V. Deslandres**
  - Enseignante-chercheure à l'IUT de Lyon depuis 2003
  - 5 ans chez Renault - Direction de la Recherche
  - Laboratoire LIRIS, équipe IA (modélisation multi-agents)

**Tour de salle**

- Quelle formation antérieure ?
- Connaissance de DevOps ? d'UML ? de LOO ?

# Plan du Chapitre 1

## Introduction

… Vous avez dit *DEVOPS* ?

- Pourquoi une méthode de modélisation
  - Coûts du logiciel
- Quels sont les problèmes liés au développement d'applications ?
  - Qualimétrie
- Qu'est-ce qu'un modèle ? Modèle OO ?
- UML vs. Agilité
- Les AGL

# Bibliographie UML

BU IUT Lyon !

- Pour **débuter** :
  - Introduction à UML, de Sinan Si Alhir, traduit par Alexandre Gachet, 221 p., 1ère édition, sept. 2005, Editions O'Reilly.
- **Référence** : mise en pratique d'UML dans des projets réels
  - UML en Action, Pascal Roques, Franck Vallée, Ed. Eyrolles
  - Modélisation Objet avec UML, Muller, Gaertner, Eyrolles
- Comprendre **UML sur des applications web** :
  - UML2 - Modéliser une application web, Pascal Roques, 2008, Editions Eyrolles


# Rappel : parties prenantes du dév. logiciel

- **Utilisateurs**
  - Ceux qui se servent du logiciel
- **Clients**
  - Ceux qui paient pour le logiciel
- **Développeurs** (les *Dev*)
  - Ceux qui conçoivent et implémentent le logiciel
- **Exploitants** (la Prod, les *Ops* en Anglais)
  - Ceux qui exploitent le logiciel (mise en production, maintenance, évolution)
- **Managers**
  - Ceux qui supervisent la production du logiciel


# Le courant Devops

(26 slides)