

Projet de développement

Philippe Collet

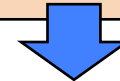
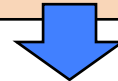
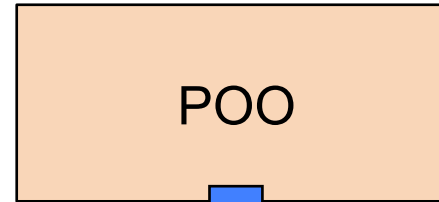
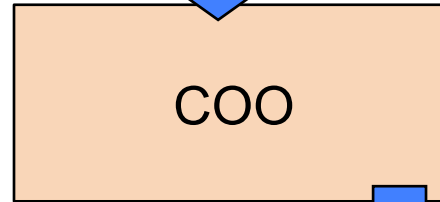
Licence 3 MIAGE – S6

2016-2017

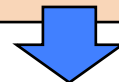
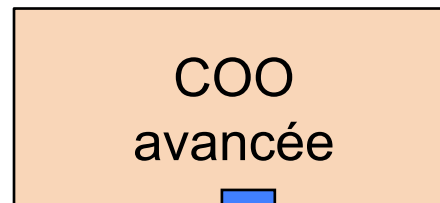
http://miageprojet2.unice.fr/index.php?title=User:PhilippeCollet/Projet_de_d%C3%A9veloppement_2016-2017

Contexte et mise en perspectives

Systèmes modulaires



Systèmes orientés objets



Architectures logicielles à plus grande échelle :

- Approches orientées services et micro-services (modulaires !)
- Couche objet pour connexion automatisée aux BDs
- Déploiement distribué

Premier semestre

Second semestre

Master...

Objectifs

- ❑ **Réalisation, en équipe,**
 - d'un développement OO de taille conséquente
 - à partir d'un cahier des charges et d'une architecture préétablis en Java

- ❑ **Donc : du développement !**
 - **Sans (gros) problème de conception**
 - **Avec des problèmes de**
 - ◆ Communication (équipe de 5)
 - ◆ Techniques de programmation (API, etc.)
 - ◆ Fiabilité (tests unitaires indispensables)
 - **En quasi-autonomie**
 - **Avec professionnalisme**
 - ◆ Bonne réponse au cahier des charges
 - ◆ Efficacité, rapidité, qualité



Objectifs



❑ Travail demandé

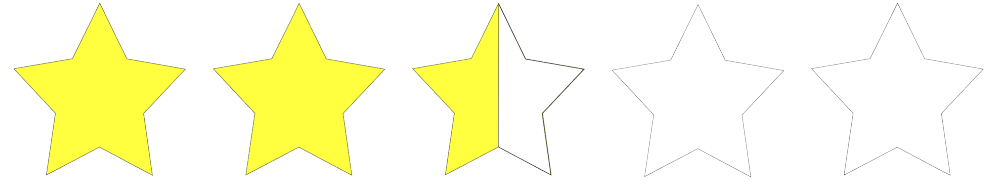
- **Beaucoup, beaucoup (beaucoup !) de travail personnel**
- Surtout par rapport au faible volume des TD
- Pour fournir du code et de la gestion de projet (expliciter ce que vous faites et ce que vous allez faire)

❑ Problématique

- Comment vous organiser en équipe ? Développer en équipe ? Coder/tester ?
Etre efficace ? Communiquer ?
- Passer du cahier des charges à une définition et en suivi :
 - ◆ Des objectifs généraux et du livrable principal (l'application finale)
 - ◆ Des jalons pour y arriver, de comment évaluer qu'on arrive bien à ces jalons
 - ◆ Des livraisons intermédiaires et démos qui ont de la valeur client
 - ◆ Des contraintes

Calendrier

- ☐ Jeudi 19 janvier : cours 1
- ☐ Lundi 23 janvier : **date limite de retour par mail des formations des équipes (5 étudiants par équipe, mêmes équipes en COO avancé et Projet devt)**
- ☐ Jeudi 26 janvier : cours 2
- ☐ Jeudi 2 février : cours 3, **publication du sujet**
- ☐ Jeudi 9 février : cours 4, **dernier cours**
- ☐ Jeudi 16 février : **premier découpage en ligne (milestones = version JIRA + premiers tickets)**
- ☐ Lundi 27 février : TD 1, **premier TD de suivi**
- ☐ Lundi 6 mars : TD 2
- ☐ Lundi 13 mars : TD 3
- ☐ Lundi 20 mars : TD 4
- ☐ Lundi 27 mars : TD 5
- ☐ Lundi 3 avril : TD 6
- ☐ Lundi 10 avril : TD 7, **dernier TD de suivi**
- ☐ **Vendredi 21 avril, 23h59 : arrêt du développement (site de gestion du projet et des sources)**
- ☐ **Lundi 24 avril : soutenance**



- Code livré : architecture, qualité et tests : 40 %**
- Gestion de Projet (ticket, git, organisation) : 25 %**
- Soutenance : 20 %**
- Avancement en TD : 15 %**

Principe de suivi : 7 semaines de TD de suivi

□ Lundi Matin :

■ 8h-12h15 : 2 à 4h en présence de l'enseignant tuteur

- ◆ Questions sur les fonctionnalités à réaliser
- ◆ Propositions / validation sur le découpage du travail
- ◆ Validation sur la conception de l'application
- ◆ Surveillance de l'avancement du projet
- ◆ Suivi et évaluation de chaque membre de l'équipe individuellement
- ◆ Aide technique sur le langage utilisé

■ Tout au long de la matinée :

- ◆ En profiter pour continuer d'avancer sur les points durs
- ◆ Bien valider la répartition du travail, la charge de chacun
- ◆ Mettre en place des tests
- ◆ Finaliser un test d'intégration avec toute l'équipe

Le TD de suivi ne fait que le... suivi

- ❑ Le développement dure 8 à 9 semaines, et non pas 2 ou 3 heures**
 - Il faut organiser l'activité de ces semaines avant et pendant
 - Chaque heure "perdue" compte
 - Il faut gérer l'information (documents, codes, tests...) en continu, particulièrement lorsqu'on travaille à 5...

- ❑ Avant le démarrage des TD de suivi, il vous est demandé :**
 - De former des équipes
 - De vous auto-former (un minimum) aux outils (atlas, git)
 - De formaliser le cahier des charges de ce que vous avez à réaliser sous forme de jalons (objectifs intermédiaires), avec un premier découpage en tâche à réaliser

Soutenance et évaluation

- ❑ **Soutenance : 15 minutes selon un plan préétabli**
 - **Présentation (**technique**) de la réalisation**
 - **Fonctionnalités réalisées (ou pas)**
 - **Grand choix de conception (et justification)**
 - **Organisation des tests**
 - **Points forts et points faibles de votre implémentation**
 - **Gestion du projet, problèmes rencontrés**
 - **Et une mini-démo (une démo se prépare...) de 2 minutes maximum**
 - **Attention, le travail de chacun doit être bien identifié**

- ❑ **Evaluation**
 - **La forme (soutenance, documents,...)**
 - **Le code (qualité, test...)**
 - **Mise en œuvre des outils (atlas, git)**
 - **Gestion du projet : livrables réguliers**

Organisation du cours

- Introduction
- Gestion de versions
- Gestion des tickets / atlassian
- V&V, Tests unitaires en Java - Junit
- Objets Mock – Mockito
- TDD – Principes et application - Kata
- Construction automatique - Maven
- Agilité et Kanban
- Documentation

