

Projet transversal M1 - 2016

E.Coquery

`emmanuel.coquery@univ-lyon1.fr`

`http://liris.cnrs.fr/~ecoquery`
→ Enseignement → Projet transversal

10/10/2016

Projet transversal (MultiMIF)

Objectif : pratiquer le développement d'un projet

- Se rapprocher d'une expérience de développement professionnel
- Mettre en pratique les techniques d'autres cours de M1
 - Programmation avancée, Génie logiciel, Programmation Web, Gestion de données pour le Web, ...

Groupes de 6 étudiants

Pas le choix de vos partenaires

Nécessite de s'organiser

- Méthodes et outils

Trouver sa place dans le groupe

- quel rôle pour qui ?

Une note de groupe

- modulable par étudiant (plus ou moins 2 points)

MultiMIF 2016 : développement collaboratif en ligne

Objectif : pouvoir développer une application entièrement en ligne :

- Éditer du code (y compris coloration syntaxique, etc)
- Conserver, versioner et partager le code
- Compiler, tester, construire des paquets
- Collaborer (documents, wiki, forums/chats)
- Gérer le suivi des développements et des bugs
- Extraire des informations du code et les utiliser, par exemple
 - commentaires TODO, affichage/liens Javadoc, etc
- Pouvoir importer/exporter le code d'un projet
- Gérer les utilisateurs et leurs droits

Organisation du projet

Méthodologie de projet : SCRUM

- bien définir les différents rôles
- utiliser le système de demandes de la forge pour le suivi (avancement, temps passé)
- planifier le travail du groupe
- faire des comptes rendus de toutes les réunions

Durées des mêlées : 1 semaine (moins d'éventuels cours/TD/TP)

4 semaines pour le projet

Semaine 1 : du 10 au 14 octobre

Semaine 2 : du 17 au 21 octobre

- constitution et organisation des groupes
- configuration des environnements
 - développement, test, démo
- conception
- planification
- 1er jet d'architecture (itération 0)

4 semaines pour le projet

Semaine 3 : du 14 au 18 novembre

- développement

Semaine 4 : du 21 au 25 novembre

- développement
- soutenances

Permanences cette semaine

- Lundi après-midi : réunion de lancement (EC-AT)
- Mardi après-midi : Conception - maquettage
- Mercredi matin : Install (support technique EC - LM)
- Mercredi après-midi : Conception AT
- Jeudi matin : Install (support technique EC - LM)
- Jeudi après-midi : Conception
- Vendredi matin : Bilan conception AT
- Vendredi après-midi : Fin install (support technique EC ? LM ?)

Gestion de la qualité

Gérer la qualité de la production

- Tester le code écrit
- Utiliser des outils d'audit de code (Sonarqube)
 - Règles de codage
 - Quantité de code testé dans les test unitaires
- Faire relire son code par les autres
→ *code review* dans la forge

Livrables de type rapport

Documents, comptes-rendus :

- Définir un modèle de document commun à tous les rapports du groupe
 - Titre, date de dernière modification, auteurs
 - Tableau des modifications et relectures :

Date	Personne	Action
2016-10-10	L. Médini	Validation
2016-10-10	A. Tabard	Relecture, corrections mineures
2016-10-09	E. Coquery	Première version complète
2016-10-07	E. Coquery	Plan

- Les comptes rendus de réunion doivent lister présents, absents excusés et absents (uniquement pour les membres du projets sensés participer à la réunion)

Ces livrables sont à déposer **sur la forge Redmine** dans le projet du groupe, **dans la section "Fichiers"**

Document de fin d'études préliminaires

- VOTRE cahier des charges
- Cas d'utilisation principaux
- Risques identifiés
- Pistes d'architecture

A rendre pour le 16 octobre 2016

Document d'architecture logicielle

- Résumé de l'architecture (1 diagramme + explications)
- Ensemble des diagrammes UML
 - Description du matériel et du logiciel
 - Vues statiques et dynamiques
- Mémos techniques (choix techniques et justifications), en particulier :
 - Représentation des ressources en REST
 - Solution(s) de persistance des données

A rendre le jour de la soutenance

Rendu du code

- E. Coquery, A. Tabard et L. Médini devront avoir le rôle *reviewer* dans le projet Redmine du groupe.
- Ajouter un tag « FINAL » à la version du dépôt correspondant au rendu
 - Ce tag doit dater au plus tard du **jour de la soutenance**
- Ce code doit être déployé le jour de la soutenance sur une des VMs de l'équipe à une URL du type :
`http://192.168.xxx.yyy/multimif/demo`
 - Il devra rester déployé jusqu'au jury du premier semestre
 - Un **login** disposant de droits d'**administrateur** devra être préconfiguré sur la démo et accessible aux enseignants. Ce login et le mot de passe correspondant devront être clairement indiqués sur la **première page du wiki** du projet Redmine.

Compte-rendus de gestion de projet

- A la fin de chaque mêlée : document bilan
 - Résumé de l'avancement durant le mêlée
 - Identification des points de blocage techniques et/ou humains rencontrés
 - y compris leur (non) résolution
 - état de la qualité du projet (e.g. screenshots Sonarqube)

Démonstration et soutenance

Démonstration

- Préparer un scénario de démonstration
- Montrer les différentes facettes de l'interface utilisateur et préciser les échanges client-serveur
- Montrer que l'application persiste bien les différentes informations

Soutenance

- Présenter votre mode de fonctionnement collaboratif, toutes les informations indiquant comment vous avez conduit votre projet
- Présenter vos choix d'architecture, quelques diagrammes UML, les patterns utilisés, etc
- Montrer que votre produit répond aux attentes du client

Contraintes techniques

Application Web :

- Serveur Java
- Interactions avec le serveur en suivant le principe REST
 - Sauf pour l'authentification (gestion via `filter`)

Construction, test unitaires, packaging : `maven`

Persistance des données :

- Données applicatives : base de données
 - + framework si adapté
- Code et versions : libre, par exemple
 - gestionnaire de version (e.g. `git` ou `mercurial`)
 - base de données

Contacts

- Gestion de données et UE en général : Emmanuel Coquery
- Programmation Web : Lionel Medini
- Gestion de projet : Aurélien Tarbard

`prenom.nom@univ-lyon1.fr`