

II.2405 – Génie Logiciel

INFORMATIONS GÉNÉRALES

Titre : Génie Logiciel
Identifiant du module : II.2405
Responsable du module : Zakia KAZI-AOUL
ECTS : 5
Quantité de travail moyenne par élève : de 100h à 150h dont 42h encadrées
Travail en équipe : oui
Mots clés : Extraction de besoins, Cycle de développement, Test, Approches agiles, Scrum, Kanban, XP, TDD, Design pattern, UML, Clean code

PRÉSENTATION

Les systèmes informatiques, tels que les réseaux sociaux, les sites marchands ou encore les jeux vidéo, constituent une composante importante de notre quotidien. Le développement de ses systèmes nécessite l'utilisation de méthodes, d'approches, et d'outillages adaptés aux différentes phases de leur cycle de vie. Ceci se traduit par l'application d'une approche systématique, disciplinée et quantifiée pour la spécification, la conception, le développement, le test et le déploiement et la maintenance de ces systèmes.

Dans ce module, vous allez vous familiariser aussi bien aux méthodes classiques qu'aux approches agiles qui permettent de réaliser des produits informatiques rentables et de qualité, répondant au mieux aux exigences du client. Vous allez aussi découvrir les "best practices" en développement logiciel.

OBJECTIFS PÉDAGOGIQUES

- Savoir utiliser les techniques d'extraction des besoins
- Réaliser un cahier des charges à partir de besoins client
- Réaliser un dossier de spécification avec une approche agile
- Conduire un projet à l'aide des approches agiles Scrum et Lean
- Savoir appliquer les best practices dans le développement logiciel

Prérequis

Première expérience en gestion de projet en équipe, notions de base en Java (à défaut Python) et POO.

Contenu/programme

Concepts

- Méthodes et processus (cycles, méthodes classiques, RUP)
- Qualité du logiciel
- Extractions de besoins
- Approches agiles (Kanban, XP, amélioration continue, Scrum)
- Design patterns
- Test
- Clean code (principes SOLID, etc.)

Outils utilisés par l'enseignant/intervenant

- Matériel d'animation de mise en pratique (Post-it, Lego®, tableau de bord, etc.)
- Environnement de développement Java

Outils utilisés par l'apprenant

- Outils de planifications et de conduite de projet (ex : Jira, Scrum Board)
- Quelques design patterns (Factory, Strategy, etc.)
- Artefacts issus des approches agiles (ex : elevator pitch, impact mapping, story map, user story, etc.)
- JUnit pour faire du Test-Driven Development (TDD)

MODALITÉS PÉDAGOGIQUES

Méthodes d'apprentissage

Le module se déroule en alternant théorie (environ 1/3) et pratique (environ 2/3). Les cours peuvent être dispensés en classe entière ou en petits groupes.

Ce module comprend des séances de projets par équipe (4 à 5 élèves par équipe) ainsi que des séances de travaux pratiques. Les concepts théoriques sont souvent illustrés par des cas réels, ou peuvent faire l'objet de mise en pratique immédiate au travers de jeux de rôle ou d'exercices.

Modalités d'évaluation

L'évaluation de ce module repose sur un projet en équipe de 4 à 5 élèves ainsi que sur des contrôles de connaissance (évaluation individuelle).

- Évaluation collective (de 40% à 50%) : Projet de spécification par équipe
- Évaluation individuelle (de 60% à 50%) : 2 à 3 contrôles de connaissance répartis dans le semestre

Langue de travail

Les supports sont présents en anglais. Le cours quant à lui est assuré en français.

Les livrables du projet peuvent être rendus en français ou en anglais. Les différents contrôles de connaissance sont proposés en français et en anglais.

BIBLIOGRAPHIE, WEBOGRAPHIE, AUTRES SOURCES

- **Open class room :**
 - <https://openclassrooms.com/courses/initiez-vous-a-la-gestion-de-projet-agile>
 - <https://openclassrooms.com/courses/gerez-un-projet-digital-avec-une-methodologie-en-cascade>
- **Lean Startup :**
 - <http://theleanstartup.com>
 - <http://theleanstartup.com/book>
- **Story mapping :**
 - <https://www.dunod.com/sciences-techniques/story-mapping-visualisez-vos-user-stories-pour-developper-bon-produit>
 - <https://www.featuremap.co/fr>
- **Kanban :**
 - <https://www.dunod.com/sciences-techniques/kanban-pour-it-une-nouvelle-methode-pour-ameliorer-processus-developpement>
- **Scrum :**
 - <https://www.dunod.com/sciences-techniques/scrum-guide-pratique-methode-agile-plus-populaire>
 - <https://openclassrooms.com/courses/gerez-votre-projet-avec-une-equipe-scrum>
- **User-stories :**

- <https://www.mountaingoatsoftware.com/books/user-stories-applied>
- **Conférences agiles**
 - Agile France : conf.agile-france.org/
 - Agile Tour : www.agiletour.org
 - Lean Kanban France : <http://leankanban.fr>
- **Clean code**
 - Clean Code: A Handbook of Agile Software Craftsmanship, Robert C. Martin
- Head First Design Patterns, Freeman, Eric T, Elisabeth Robson, Bert Bates, Kathy Sierra, O'Reilly Media, 2004
- <http://www.sei.cmu.edu/publications/documents/02.reports/02tr011.html>
- www.swebok.org
- Software Engineering, Fourth Edition, Shari Lawrence Pfleeger, Joanne M. Atlee, International Edition, 2010
- Software Engineering: A Practitioner's Approach, Roger S. Pressman. McGraw Hill Higher Education, Avril 2009
- Software Requirements, Second Edition (Pro-Best Practices), Karl E. Wiegers, Microsoft Press, Mars 2003