Programmation orienté objet 2/Ingénierie logicielle

Composante
IAE Savoie
Mont Blanc

En bref

- **Langues d'enseignement:** Français
- **Ouvert aux étudiants en échange:** Oui

Présentation

Description

L'objectif est de connaître les différentes étapes et méthodes à mobiliser dans chacune de ces étapes, d'un projet de développement d'une application logicielle. Cette matière aborde les notions de base en ingénierie logicielle en favorisant la mise en pratique. Les méthodes et outils sont évoqués avec un objectif de "terrain" et de façon concrète.

Important: les séances se feront avec l'ouvrage de référence en mains: *"Software Engineering"*, I. Sommerville, Pearson

Objectifs

Il s'agit de développer des connaissances et compétences en ingénierie logicielle dans le cadre d'une approche ***"par la pratique"***. Ce cours pourra apporter des notions dans la conception et la modélisation de systèmes logiciels, des compléments en programmation orientée-objet en *Java* (le langage *Java* étant celui qui sera principalement utilisé pour la mise en oeuvre des notions et éléments abordés).

Heures d'enseignement

CM	Cours Magistral	21h
TD	Travaux Dirigés	15h

Pré-requis obligatoires

Savoir programmer une application logicielle avec un langage orienté objet (de préférence). Connaître les méthodes agiles (Scrum).

Plan du cours

Cet enseignement abordera principalement:

* la notion de génie logiciel
* les principaux cycles de vie, processus logiciels, méthodes agiles et lean
* l'ingénierie des besoins et la notion de cahier des charges
* la réalisation d'un benchmark, d'une étude comparative des solutions existantes
* la notion d'architecture logicielle et architecture dirigée par les modèles
* les tests et la conception dirigée par les tests (Test Driven Development)
* les patrons de conception (Design Patterns)
* la loi de Demeter, la refactorisation de code (code refactoring)
* la notion d'intégration continue et quelques outils logiciels assistant le concepteur/développeur
* la notion de DevOps
* les notions de base en qualité logicielle, maintenance, évolution et quelques liens avec les approches et techniques en Intelligence Artificielle

Des compléments en langages de programmation pourront être abordés :

* le traitement des exceptions en Java
* le modèle à évènements dans le cadre de développement d'interfaces graphiques
* l'utilisation de bases de données en Java
* ...

Compétences visées

* Connaître les différentes facettes et activités de l'ingénierie logicielle
* Savoir conduire une étude comparative et multi-critère de solutions existantes
* Savoir définir une architecture logicielle
* Savoir utiliser les bonnes pratiques dans le cadre d'un projet d'ingénierie logicielle
* Savoir mobiliser un ensemble d'outils pour la mise en oeuvre d'un projet d'ingénierie logicielle

Bibliographie

* I. Sommerville, Software Engineering, Pearson
* J. Appelo, Management 3.0, Addison-Wesley Professional.
* J. Appelo, #Workout
* J. Appelo, How to CHange The World, Jojo Ventures.
* M. and T. Poppendieck, Implementing Lean Software Development, Addison-Wesley Professional.
* E. Freeman, E. Freeman, B. Bates, K. Sierra, Head First Designa Patterns O'Reilly.
* M. Fowler Refactoring – Improving the Design of Existing Code Addison-Wesley.

* S. McConnell, Code Complete 2, Microsoft Press.
* A. Oram, G. Wilson, Beautiful Code, O'Reilly.
* Plusieurs ouvrages très intéressants dans la série *The Pragmatic Programmers*: Prgamatic Version Control, Pragmatic Unit Testing, Practices Of An Agile Developer, Lean from the Trenches
* M. Grand, Patterns in Java, Wiley.
* S. Vance, Quality Code, Wiley.
* D. Asteels, test-driven development - A practical guide, The code series.
* B. Meyer, Conception et programmation orientées objet, Eyrolles.
* P-.A. Muller, N. Gaertner, Modélisation objet avec UML, Eyrolles.
* D. Pilone, R. Miles, Head First Software Development O'Reilly.
* K. Sierra, B. Bates Head First Java O'Reilly
* Gosling, Joy, Steele, Bracha The Java Language Specification, Third Edition Addison Wesley.
* P. Roques UML 2 par la pratique, Eyrolles.
* E. Gamma, R. Helm, R. Johnson, J. Vlissides Elements of Reusable Object-Oriented Software Addison-Wesley.
* K. Beck Test Driven Development: By Example Addison-Wesley.
* K. Beck Smalltalk Best Practice Patterns Prentice-Hall.
* Git http://git-scm.com/ book/fr

Infos pratiques

Contacts

Responsable du cours

Herve Verjus
+33 4 50 09 65 94
Herve.Verjus@univ-savoie.fr

Lieux

Annecy-le-Vieux (74)

Campus

Annecy / campus d'Annecy-le-Vieux