

Description du module

Génie logiciel: méthodes de développement et architecture

Généralités**Nombres de crédits ECTS**

3

Sigle du module

TSM_SoftwEng.

Version

04 Mars 2013

Responsable du module

Martin Kropp, FHNW

Langue

	Lausanne	Berne	Zurich
Enseignement	<input type="checkbox"/> E <input checked="" type="checkbox"/> F	<input type="checkbox"/> D <input type="checkbox"/> E <input type="checkbox"/> F	<input checked="" type="checkbox"/> D <input type="checkbox"/> E
Documentation	<input checked="" type="checkbox"/> E <input type="checkbox"/> F	<input type="checkbox"/> D <input type="checkbox"/> E <input type="checkbox"/> F	<input type="checkbox"/> D <input checked="" type="checkbox"/> E
Questions d'examen	<input type="checkbox"/> E <input checked="" type="checkbox"/> F	<input type="checkbox"/> D <input type="checkbox"/> E <input type="checkbox"/> F	<input checked="" type="checkbox"/> D <input type="checkbox"/> E

Catégorie du module

- Bases théoriques élargies
- Approfondissement technique et scientifique
- Modules de savoirs contextuels

Périodes

- 2 périodes d'enseignement frontal et une période d'exercice par semaine
- 2 périodes d'enseignement frontal par semaine

Breve description /Explication des objectifs et du contenu du module en quelques phrases

Le module fournit une étude approfondie des sujets sélectionnés dans le domaine du génie logiciel. Ces sujets sont issus des domaines suivants: méthodes de spécification, méthodes de développement agiles, architecture logicielle, évolution du logiciel.

Objectifs, contenu et méthodes**Objectifs d'apprentissage et compétences visées**

- L'étudiant comprend et peut expliquer les avantages d'une méthodologie de développement itérative et incrémentale.
- En fonction des caractéristiques d'un projet donné, l'étudiant est en mesure de sélectionner, d'adapter et d'appliquer la méthodologie de développement appropriée.
- L'étudiant connaît des principes de conception et d'architecture avancés et peut les utiliser pour diriger la phase de conception d'un système.
- L'étudiant est conscient qu'un logiciel est un système complexe en évolution constante.
- L'étudiant sait comment améliorer, étendre et intégrer des logiciels existants, tout en maintenant leur qualité.

Contenu du module avec pondération des contenus d'enseignement**Génie Logiciel**

- Principes et valeurs des méthodes de développement agiles
 - Création de valeur
 - Gestion du risque
 - Culture d'équipe
 - Relations clients
- Bonnes pratiques et techniques de gestion
 - Gestion de la communication entre parties prenantes
 - Revues de projets
 - Assurance qualité
 - Gestion des changements
 - Planification itérative et incrémentale
- Méthodologies de développement modernes

- Vue d'ensemble et comparaison de différentes approches
- Unified Process (UP), eXtreme Programming (XP), Scrum
- Implications sur la gestion de projet

Architecture logicielle

- Le rôle de l'architecture est des architectes logiciels
 - notion d'architecture
 - modèles de référence, architectures de référence
 - structures et vues architecturales
 - architecture et documentation
 - les rôles de l'architecte logiciel
- Principes de conception avancés
 - interface-oriented design
 - couplage et cohésion
 - gestion des dépendances
 - les principes SOLID
 - conception par contrat
- Modèles de conception et d'architecture
 - styles d'architectures
 - modèles pour architectures distribuées
- Sélection, création et évaluation d'architectures logicielles
 - qualités systémiques
 - méthodes d'analyse

Evolution du logiciel

- Principes d'évolution
 - Développement, maintenance, évolution
 - "Software aging"
 - Compréhension du code
- Analyse et qualité du logiciel
 - métriques
 - techniques de visualisation
 - débogage systématique
 - contrôle de qualité continu
 - techniques de contrôle d'architecture
- Evolution de code hérité
 - Re-Technologies: Reverse Engineering, Re-Engineering, Re-Factoring
 - Re-engineering orienté objet
 - Travailler efficacement avec du code hérité
 - Tester des systèmes hérités

Méthodes d'enseignement et d'apprentissage

2 périodes de cours et 1 période d'exercice par semaine

travail personnel, lectures et exercices

Connaissances et compétences prérequis

- Conception et programmation orientées selon l'objet dans plus d'un langage de programmation
- Unified Process (UP)
- Unified Modeling (UML)
- Design Patterns: Elements of Reusable Object-Oriented Software (Gamma, Helm, Johnson, Vlissides; ISBN 0-201-63361-2)
- Modèles/styles/principes architecturaux essentiels, empilages, architecture orientée service, MVC, pipes&filtres, par exemple venant de Pattern-Oriented Software Architecture Volume 1 (Buschmann, Meunier, Rohnert, Sommerlad, Stal; ISBN 0-471-95869-7)
- Notions de gestion des versions et des configurations
- Notions d'intégration continue et de construction automatisée (par ex. : ant, make, cruisecontrol)
- Notions de tests unitaires (par exemple junit ou nunit)

Bibliographie

1	Jim Highsmith: Agile Software Development Ecosystems Mary & Tom Poppendiek: Lean Software Development Kent Beck: eXtreme Programming 2nd Ed.
2	Ken Schwaber et al, Agile Software Development with Scrum, Prentice Hall, 2002
3	Alistair Cockburn: Agile Software Development Alistair Cockburn: Crystal Clear
4	Stephen Palmer, John Felsing: A Practical Guide to the Feature-Driven Development Kent Beck: eXtreme Programming explained, 2nd Ed.
5	Gernot Starke: Effektive Software Architekturen 2. Auflage
6	Ken Pugh, Interface Oriented Design
7	Doug Schmidt et.al.: Pattern-oriented Software Architecture, Vol. 2 Frank Buschmann et al: Pattern-oriented Software Architecture, Vol. 4
8	Robert C. Martin: Agile Software Development
9	Lehmann "Laws of Software Evolution Revisited"
10	Martin Fowler et al, Refactoring Joshua Kerievsky, Refactoring to Patterns
11	Gerard Meszaros: xUnit Test Patterns
12	Michael Feathers, Working Effectively with Legacy Code
13	Andreas Zeller: Why Programs Fail ISBN 1558608664
14	Len Bass, Paul Clements, Rick Kazman: Software Architecture in Practice 2 nd Ed.

Mode d'évaluation**Conditions d'admission aux examens de fin de module (tests exigés)**

Aucun

Examen écrit de fin de module

Durée de l'examen: 120 minutes

Moyens autorisés: Documentation papier