

MICRO-371

**Architecture software**

Ingram Sandy, Silva Eric

Cursus	Sem.	Type
Microtechnique	BA6, MA2, MA4	Opt.
Robotique	MA2, MA4	Opt.

Langue d'enseignement	français
Crédits	3
Session	Eté
Semestre	Printemps
Examen	Ecrit
Charge	90h
Semaines	14
<b>Heures</b>	<b>3 hebdo</b>
Cours	2 hebdo
Exercices	1 hebdo
<b>Nombre de places</b>	

**Résumé**

Ce cours couvre des sujets liés à l'architecture software moderne et industrielle : la gestion agile de projets, la spécification des besoins, le développement d'applications critiques, la programmation robuste, et les aspects devOps (automatisation de tests, pipeline CI/CD)

**Contenu****Architecture Software**

- Patron de conception (Design patterns) et principes SOLID
- Styles d'architecture
- Processus de développement dans les domaines critiques (automobile, aviation...)
- Spécification des besoins et définition des exigences

**Gestion de projet**

- Approche de gestion agile (principes agiles, outils de planification, méthodologie SCRUM)
- Evaluation de l'effort, des coûts et des risques
- Fiabilité et cycle de vie du produit

**Code robuste et industrialisation**

- "Best practices" pour le développement d'applications critiques (choix des outils, bibliothèques, tests)
- Directives "statiques" de codage (MISRA C...) et outils de test (CPPCheck, PC-lint...)
- Automatisation des tests (approches: TDD, BDD / types: unitaire, intégration...)
- Outils de monitoring (couverture des tests, traçage d'erreurs)

**Intégration et déploiement continu (CI/CD)**

- Introduction au DevOps et CI/CD
- Gitflow et Trunk-based development
- "Best practices" pour l'utilisation de GIT

### Acquis de formation

A la fin de ce cours l'étudiant doit être capable de:

- Expliquer les avantages du développement agile.
- Prendre en considération durant la phase de conception et d'implémentation de logiciels que ces derniers sont en constante évolution.
- Expliquer les techniques et outils de développement et de gestion de projet agile.
- Exploiter les styles d'architecture et patrons de conception avancés.
- Explorer les styles d'architecture et patrons de conception pour orienter les décisions de conception.
- Choisir ou sélectionner les techniques de maintenance et d'évolution pour le développement continu de systèmes logiciels évolutifs et pour l'extension de systèmes existants tout en maintenant leur qualité.
- Choisir ou sélectionner une architecture répondant aux besoins du domaine d'application
- Concevoir un code sûr répondant aux exigences métier
- Classer différentes architectures en fonction de leur niveau de fiabilité
- Formuler des spécifications de code permettant un développement fiable et atomique

### Méthode d'évaluation

- Projet de groupe
- Evaluation écrite

### Ressources

#### Liens Moodle

- <https://go.epfl.ch/MICRO-371>