

GROUPE 40

(514) 987-3000

Jeudi, de 18h00 à 21h00

DESCRIPTION Aperçu des outils pour assister au développement et à l'entretien des logiciels. Plates-formes d'intégration des outils. Environnements de développement. Outils pour la rétro-ingénierie des logiciels. Critères et stratégies d'évaluation des outils.

- OBJECTIFS
- Maîtriser l'utilisation de divers outils d'assistance au développement et à la maintenance de logiciels.
 - Maîtriser l'utilisation de plate-formes de développement libres intégrant divers outils.
 - Être capable d'identifier, d'évaluer et d'apprendre à utiliser de nouveaux outils d'assistance au développement.
 - Faire apprécier aux étudiants l'importance des enjeux reliés à la qualité et la productivité engendré par l'utilisation d'outils logiciels.
 - Comprendre les tendances émergentes en réalisation de logiciels.
 - Comprendre la problématique de la qualité des logiciels.
 - Comprendre la dichotomie déclaratif / impératif.
 - Initier les étudiants à la réalisation de langages spécifiques à un domaine d'affaire (DSL).
 - Initier les étudiants aux langages de programmation dynamiques.
 - Initier les étudiants aux techniques de métaprogrammation.
 - Initier les étudiants aux nouvelles approches dans le domaine.

ÉVALUATION	Description sommaire	Date	Pondération
	Projet de session : premier livrable	11 février 2010	20%
	Projet de session : second livrable	18 mars 2010	20%
	Projet de session : dernier livrable	15 avril 2010	20%
	Examen final	22 avril 2010	30%
	Participation	s. o.	10%

Un travail remis en retard reçoit la note zéro à moins d'avoir fait l'objet d'une entente préalable avec le professeur.

Le détail des conditions de réalisation de chaque travail est précisé avec la description du travail.

La qualité du français fait partie intégrante des critères d'évaluation des travaux et des examens jusqu'à un maximum de 25%.

Les règlements concernant le plagiat seront strictement appliqués. Pour plus de renseignements, veuillez consulter les sites suivants :

<http://www.sciences.uqam.ca/decanat/reglements.php>

<http://www.bibliotheques.uqam.ca/recherche/plagiat/index.html>

La note de passage du cours est de 60% pour l'ensemble de l'évaluation et de 50% pour l'examen final.

- CONTENU
- Programmation orientée-aspects
 - Langages dynamiques
 - em>Domain-Specific Languages
 - Annotations
 - Écosystèmes de développement
 - Méta-programmation
 - REST
 - WebServices
 - em>Business rules engines
 - Déclaratif vs Procédural
 - Ligne de commande
 - CMMI
 - Personal Software Process

- Cloud Computing
- Programmation fonctionnelle
- Estimation
- Métriques
- Approche Top-Down
- Tests Unitaires
- Test-Driven Development
- Techniques de productivité personnelles
- Domain-Driven Design

Approche pédagogique

Le cours se base sur une approche par projet. L'approche par projet s'inscrit dans l'esprit de la formation par compétence. Il permet la mobilisation des ressources de l'étudiant dans la réalisation d'une tâche authentique.

Les étudiants devront travailler en équipe (taille: de 1 à 4 étudiants maximum). Il est fortement conseillé de ne pas effectuer les projets seul. Les critères de correction et les attentes ne seront pas modifiés en fonction du nombre d'étudiants dans l'équipe. Chaque équipe formera un groupe de consultants à qui seront confiés des mandats à réaliser. Chaque projet sera réalisé par l'ensemble des équipes.

Chaque période de cours sera divisée en deux. La première partie abordera des thèmes choisis préalablement par les étudiants pour les aider dans la réalisation de leurs projets. Si aucun thème n'est proposé par les étudiants, un thème contemporain sera présenté par le professeur à son choix. La deuxième partie consistera à réviser avec chaque équipe le déroulement des travaux entrepris dans le but d'éviter des dérives éventuelles.

RÉFÉRENCES

- VC Cockburn, A. – *Agile Software Development* – Addison Wesley, 2002
- VC Thomas, D. et Hansson, D.H. – *Agile Web Development with Rails* – Pragmatic Programmers, 2005
- VC Laddad, R. – *AspectJ in Action: Practical Aspect-Oriented Programming* – Manning, 2003
- VC Martin, R.C. – *Clean Code. A Handbook of Agile Software Craftsmanship* – Prentice Hall, 2009
- VC McConnell, S. – *Code complete* – Microsoft Press, 2nd edition, 2004
- VC Gamma, E. and Helm, R. and Johnson, R. and Vlissides, J. – *Design patterns: elements of reusable object-oriented software* – Addison-Wesley, 1995
- VC Carlson, D. – *Eclipse Distilled* – Addison-Wesley, 2005
- VC Hohpe, G. and Woolf, B. – *Enterprise integration patterns* – Addison-Wesley Boston, 2003
- VC Beck, K. – *Implementation Patterns* – Addison-Wesley Professional, 2008
- VC Fowler, M. – *Patterns of Enterprise Application Architecture* – Addison-Wesley Professional, 2003
- VC DeMarco, T. and Lister T. – *Peopleware, 2nd ed.* – Dorset House Publishing Co, 1999
- VC Tulach, J. – *Practical API Design: Confessions of a Java Framework Architect* – Apress, 2008
- VC Clark, M. – *Pragmatic project automation* – Pragmatic Bookshelf, 2004
- VC Hunt, A. – *Pragmatic Thinking and Learning* – Pragmatic Bookshelf, 2008
- VC Hunt, A. and Thomas, D. – *Pragmatic Unit Testing in Java with JUnit* – The Pragmatic Programmers, 2003
- VC Odersky, M. and Spoon, L. and Venners, B. – *Programming in Scala* – Artima Press, 2008
- VC Thomas, D. and Hunt, A. – *Programming Ruby* – Addison-Wesley, 2001
- VC McConnell, S. – *Rapid Development: Taming Wild Software Schedules* – Microsoft Press, 1996
- VC Fowler, M. – *Refactoring: Improving the Design of Existing Code* – Addison-Wesley Professional, 1999
- VC Richardson, L. and Ruby, S. – *RESTful Web Services* – O'Reilly, 2007
- VC Boudreau, T. and Tulach, J. and Wielenga, G. – *Rich Client Programming: Plugging into the NetBeans (TM) Platform* – Prentice Hall, 2007
- VC McConnell, S. – *Software Estimation: Demystifying the Black Art* – Microsoft Press Redmond, WA, USA, 2006
- VC Beck, K. – *Test-driven development: By example* – Addison-Wesley, 2003
- VC Brooks Jr, F.P. – *The mythical man-month, Anniversary ed.* – Addison-Wesley, 1995

- VC Hunt, A. and Thomas, D. – *The Pragmatic Programmer: From Journeyman to Master* – Addison-Wesley Professional, 1999
- VC Ford, N. – *The Productive Programmer* – O'Reilly, 2008
- VC Weinberg, G.M. – *The Psychology of Computer Programming, Silver Anniversary ed.* – Dorset House, 1998

A : article – C : comptes rendus – L : logiciel – N : notes – R : revue –
S : standard – U : uri – V : volume

C : complémentaire – O : obligatoire – R : recommandé