

Qualité et productivité des outils logiciels

Groupe 40

Mercredi, de 18h00 à 21h00 PK-4665 (cours)

Responsable(s) du cours

Nom du coordonnateur : MOHA, Naouel**Nom de l'enseignant :** MOHA, Naouel**Local :** PK-4425**Téléphone :** (514) 987-3000 #1660**Disponibilité :****Courriel :** moha.naouel@uqam.ca**Site Web :** <http://www.naouelmoha.net/>

Description du cours

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 du cours

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.

Maîtriser l'utilisation de dépôts de code permettant le développement collaboratif et la préservation de l'historique de développement du code.

Apprécier 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.

Contenu du cours

- Programmation orientée-aspects
- Langages dynamiques
- Domain-Specific Languages
- Annotations
- Écosystèmes de développement
- Méta-programmation
- REST

- WebServices
- Business rules engines
- Déclaratif vs Procédural
- Ligne de commande
- CMMI
- Personnel 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 reviser avec chaque équipe le déroulement des travaux entrepris dans le but d'éviter des dérives éventuelles.

Modalités d'évaluation

Description sommaire	Date	Pondération
Première présentation/démonstration		20 %
Deuxième présentation/démonstration		20 %
Troisième présentation/démonstration		20 %
Examen final		30 %
Participation		10 %

Les présentations/démonstrations et l'examen final sont sujets à la politique d'absence aux examens du département d'informatique. <http://info.uqam.ca/politiques/>

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, consultez le site suivant : <http://www.sciences.uqam.ca/etudiants/integrite-academique.html>

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

Politique d'absence aux examens

L'autorisation de reprendre un examen en cas d'absence est de caractère exceptionnel. Pour obtenir un tel privilège, l'étudiant-e doit avoir des motifs sérieux et bien justifiés.

Il est de la responsabilité de l'étudiant-e de ne pas s'inscrire à des cours qui sont en conflit d'horaire, tant en ce qui concerne les séances de cours ou d'exercices que les examens. **De tels conflits d'horaire ne constituent pas un motif justifiant une demande d'examen de reprise.**

Dans le cas d'une absence pour raison médicale, l'étudiant-e doit joindre un certificat médical original et signé par le médecin décrivant la raison de l'absence à l'examen. Les dates d'invalidité doivent être clairement indiquées sur le certificat. Une vérification de la validité du certificat pourrait être faite. Dans le cas d'une absence pour une raison non médicale, l'étudiant-e doit fournir les documents originaux

expliquant et justifiant l'absence à l'examen – par exemple, lettre de la Cour en cas de participation à un jury, copie du certificat de décès en cas de décès d'un proche, etc. Toute demande incomplète sera refusée. Si la direction du programme d'études de l'étudiant-e constate qu'un étudiant a un comportement récurrent d'absence aux examens, l'étudiant-e peut se voir refuser une reprise d'examen.

L'étudiant-e absent-e lors d'un examen doit, dans les cinq (5) jours ouvrables suivant la date de l'examen, présenter une demande de reprise en utilisant le formulaire prévu, disponible sur le site Web du département à l'adresse suivante : <http://info.uqam.ca/politiques/>

L'étudiant-e doit déposer le formulaire dûment complété au secrétariat de la direction de son programme d'études : PK-3150 pour les programmes de premier cycle, PK-4150 pour les programmes de cycles supérieurs. Pour plus de détails sur la politique d'absence aux examens du Département d'informatique, consultez le site web suivant : <http://info.uqam.ca/politiques>

Intégrité académique

PLAGIAT Règlement no 18 sur les infractions de nature académique. (extraits)

Tout acte de plagiat, fraude, copiage, tricherie ou falsification de document commis par une étudiante, un étudiant, de même que toute participation à ces actes ou tentative de les commettre, à l'occasion d'un examen ou d'un travail faisant l'objet d'une évaluation ou dans toute autre circonstance, constituent une infraction au sens de ce règlement.

La liste non limitative des infractions est définie comme suit :

- la substitution de personnes;
- l'utilisation totale ou partielle du texte d'autrui en la faisant passer pour sien ou sans indication de référence;
- la transmission d'un travail pour fins d'évaluation alors qu'il constitue essentiellement un travail qui a déjà été transmis pour fins d'évaluation académique à l'Université ou dans une autre institution d'enseignement, sauf avec l'accord préalable de l'enseignante, l'enseignant;
- l'obtention par vol, manoeuvre ou corruption de questions ou de réponses d'examen ou de tout autre document ou matériel non autorisés, ou encore d'une évaluation non méritée;
- la possession ou l'utilisation, avant ou pendant un examen, de tout document non autorisé;
- l'utilisation pendant un examen de la copie d'examen d'une autre personne;
- l'obtention de toute aide non autorisée, qu'elle soit collective ou individuelle;
- la falsification d'un document, notamment d'un document transmis par l'Université ou d'un document de l'Université transmis ou non à une tierce personne, quelles que soient les circonstances;
- la falsification de données de recherche dans un travail, notamment une thèse, un mémoire, un mémoire-crédation, un rapport de stage ou un rapport de recherche;
- Les sanctions reliées à ces infractions sont précisées à l'article 3 du Règlement no 18.

Les règlements concernant le plagiat seront strictement appliqués. Pour plus de renseignements, veuillez consulter les sites suivants : <http://www.sciences.uqam.ca/etudiants/integrite-academique.html> et <http://www.bibliotheques.uqam.ca/recherche/plagiat/index.html>

Médiagraphie

VO -- Hunt, A. and Thomas, D. – *The Pragmatic Programmer: From Journeyman to Master* – Addison-Wesley Professional, 1999

VO -- Hunt, A. and Thomas, D. – *Pragmatic Unit Testing in Java with JUnit* – *The Pragmatic Programmers*, 2003

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 -- 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 -- Ford, N. – *The Productive Programmer* – O'Reilly, 2008
 UO -- *Divers liens vers des documents en ligne seront fournis tout au long du cours.*

A : article - C : comptes rendus - L : logiciel
 S: Standard - U : uri - V : volume

C : complémentaire - O : Obligatoire - R : recommandé