

GROUPE	40 BERGER, Jacques Jeudi, de 18h00 à 21h00	berger.jacques@uqam.ca	(514) 987-3000 3699	PK-4115
--------	---	------------------------	---------------------	---------

DESCRIPTION	Rôle de la réalisation et de la maintenance dans le cycle de vie du logiciel. Évolution et maintenance du logiciel. Méthodes propres à étendre la durée de vie. Sélection de la méthode appropriée de réalisation. Prototypage. Mise au point. Gestion de la maintenance. Réutilisation et rétro-ingénierie des logiciels. L'interaction entre réalisation et maintenance sera traitée tout au long du cours.
-------------	---

OBJECTIFS	<ul style="list-style-type: none"> • Comprendre la problématique de la réalisation et de la maintenance de logiciels. • Situer les activités de réalisation et de maintenance dans le cycle de vie du logiciel. • Connaître les principales normes concernant la réalisation. • Initier les étudiants aux approches contemporaines concernant la réalisation et la maintenance. • Initier les étudiants aux environnements intégrés de développement. • Comprendre les tendances émergentes au niveau de la réalisation. • Familiariser les étudiants aux approches de «refactoring». • Initier les étudiants aux principaux outils de test. • Faire apprécier aux étudiants l'importance des enjeux reliés à l'assemblage et au déploiement des logiciels. • Familiariser les étudiants aux outils de «build». • Sensibiliser les étudiants aux activités postérieures à la réalisation. • Connaître certains environnements de développements. • Utiliser les outils de gestion de configuration. • Initier les étudiants aux nouvelles approches dans le domaine.
-----------	--

ÉVALUATION	Description sommaire	Date	Pondération
	Projet 1 – Mise en place d'un IDE de base et d'un gestionnaire de version		20%
	Projet 2 – Mise en place des outils de type "build"		20%
	Projet 3 – Mise en place des outils de type "écosystème"		20%
	Présence et participation active en classe	Continu	10%
	Examen final		30%

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.

Au plus tard le mercredi de chaque semaine, les étudiants devront avoir transmis par courriel au professeur une feuille de temps décrivant les activités accomplies lors de la semaine précédente (la date et l'heure de réception du courriel seront utilisées pour contrôler le respect de la date de tombée). La forme et le contenu de cette feuille de temps seront précisés en classe. Pour tout retard, 1% sera retranché à la note finale de l'étudiant.

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

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

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

www.integrite.uqam.ca

<http://www.bibliotheques.uqam.ca/plagiat>

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

CONTENU	<input type="checkbox"/> Introduction <input type="checkbox"/> Injection de dépendance <input type="checkbox"/> Programmation par aspect <input type="checkbox"/> Moteur de règles
---------	---

- ❑ Langages de scripts
- ❑ Qualité et métrique de la réalisation

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. Cependant, chaque équipe aura une approche à promouvoir. L'attribution des approches sera effectuée par le professeur.

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 CARLSON, David – *Eclipse Distilled* – 2005, Addison Wesley – ISBN: 0321288157
- VC CLARK, Mike – *Pragmatic Project Automation* – 2004, The Pragmatic Bookshelf – ISBN: 0974514039
- VC D'ANJOU, Jim; FAIRBROTHER, Scott; KEHN, Dan; KELLERMAN, John; MCCARTHY, Path – *The Java Developer's Guide to Eclipse (2ième édition)* – 2005, Addison Wesley – ISBN: 0321305027
- VC Ed. BOURQUE, Pierre; DUPUIS, Robert – *Guide to the Software Engineering Body of Knowledge* – 2005, IEEE Computer Society – ISBN: 0769523307
Ce livre est disponible gratuitement en format PDF sur le site de SWEBOK.
- VC HUNT, Andy & THOMAS, Dave – *Pragmatic Unit Testing – In Java with JUnit* – 2003, The Pragmatic Bookshelf – ISBN: 0974514012
- VC LADDAD, Ramnivas – *AspectJ in Action – Practical Aspect-Oriented Programming* – 2003, Manning – ISBN: 1930110936
- VC MASON, Mike – *Pragmatic Version Control – Using Subversion (2ième édition)* – 2006, The Pragmatic Bookshelf – ISBN: 0977616657
- VO Mc CONNELL, Steve – *Code Complete 2 (2ième édition)* – 2004, MicrosoftPress – ISBN: 0735619670
Ce livre est obligatoire pour le cours. Une version française est également disponible.
- VC THOMAS, Dave & HUNT, Andy – *Pragmatic Version Control – Using CVS* – 2003, The Pragmatic Bookshelf – ISBN: 0974514004
- SC ESA PSS-05-05 - *Guide to the software detailed design and production phase* – <http://martin.louis.uqam.ca/domaines/normes/esa/pss0505.pdf>
- SC ESA PSS-05-09 - *Guide to software configuration management* – <http://martin.louis.uqam.ca/domaines/normes/esa/pss0509.pdf>
- SC IEEE Std 1074-1997 IEEE - *Standard for Developing Software Life Cycle Processes*
- SC IEEE Std 1219-1998 IEEE - *Standard for Software Maintenance*
- SC IEEE Std 828-1998 IEEE - *Standard for Software Configuration Management Plans*
- SC IEEE Std 829-1998 IEEE - *Standard for Software Test Documentation*
- SC IEEE/EIA 12207.0-1996 - *Guide for Information Technology – Software life cycle processes*
- SC IEEE/EIA 12207.1-1997 - *Guide for Information Technology – Software life cycle processes – Life cycle data*
- SC IEEE/EIA 12207.2-1997 - *Guide for Information Technology – Software life cycle processes – Implementation considerations*
- SC ISO/IEC 12207:1995 - *Information technology – Software life cycle processes*
- UC Ant – <http://ant.apache.org/>
Page d'accueil du projet
- UC Apache Tomcat Servlet Container – <http://tomcat.apache.org/>
Page d'accueil
- UC CruiseControl – <http://cruisecontrol.sourceforge.net/>

- Page d'accueil du projet
- UC CVS – <http://www.nongnu.org/cvs/>
Concurrent Versions System
 - UC CVSNT Open Source – <http://www.cvsnt.org/>
Page d'accueil
 - UC Eclipse – <http://www.eclipse.org/>
Page d'accueil
 - UC Groovy – <http://groovy.codehaus.org/>
Page d'accueil
 - UC JUnit.org – <http://www.junit.org/>
Page d'accueil
 - UC Louis Martin – <http://martin.louis.uqam.ca/> Louis Martin
Portail
 - UC Maven – <http://maven.apache.org/>
Page d'accueil du projet
 - UC Sun – <http://java.sun.com/>
Site principal de Sun pour Java
 - UC Sun – <http://java.sun.com/docs/codeconv/index.html>
Conventions de codage de Java
 - UC SWEBOK – <http://www.swebok.org/>
Guide to the Software Engineering Body of Knowledge
 - UC Tigris.org - Open Source Software Engineering Tools – <http://subversion.tigris.org/>
Subversion version control system

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

C : complémentaire – O : obligatoire – R : recommandé