

Plan de cours IFT719 – hiver 2008

Processus de génie logiciel

Enseignant

Luc LAVOIE

Courriel : Luc.Lavoie@USherbrooke.ca

Bureau : D4-1010-12

Téléphone : (819) 821-8000 poste 62015

Site : <http://pages.usherbrooke.ca/llavoie/>

Disponibilité : sur rendez-vous.

Horaire

Lundi 13 h 30 à 15 h 20 D7-2009

Mercredi 08 h 30 à 10 h 20 D7-2009

Version et statut

1.0.0 - En vigueur en date du 2008-01-07

1	Introduction	2
1.1	Objet et portée du document.....	2
1.2	Définitions.....	2
1.3	Références	2
2	Présentation.....	5
2.1	Mise en contexte.....	5
2.2	Fiche signalétique	5
2.3	Objectifs spécifiques	5
3	Contenu.....	6
4	Organisation.....	6
4.1	Modalités d'enseignement.....	6
4.2	Modalités d'évaluation.....	6
4.3	Calendrier	7

1 Introduction

1.1 Objet et portée du document

Le document s'adresse aux personnes inscrites au programme d'informatique de gestion délocalisé au Maroc. Il décrit l'activité IFT719 « **Processus de génie logiciel** » offerte au trimestre d'hiver 2008. On y présente les objectifs, le contenu, l'organisation et les modalités d'évaluation du cours.

1.2 Définitions

IEEE	<i>The Institute of Electrical and Electronics Engineers, inc.</i>
PGC	plan de gestion de configuration (IEEE SCMP <i>software configuration management plan</i>).
PGP	plan de gestion de projet (IEEE SPMP <i>software project management plan</i>).
SAS	spécification d'architecture du système.
SCL	spécification de conception du logiciel (IEEE SDD <i>software design document</i>).
SES	spécification des exigences du système (IEEE SRS <i>software requirement specification</i>).
UML	<i>Unified Modeling Language</i> .

1.3 Références

1.3.1 Références essentielles

[IGL301]

COLLECTIF GL ;

IGL301 – Spécification et validation des exigences

Notes complémentaires et synthétiques, Département d'informatique, Faculté des sciences, Université de Sherbrooke, Sherbrooke, Canada, avril 2007.

[GLOGUS]

LAVOIE, Luc ;

GLOGUS – recueil de modèles de documents pour le développement logiciel.

<http://pages.usherbrooke.ca/llavoie/glogus.php>

Département d'informatique, Faculté des sciences, Université de Sherbrooke, Sherbrooke, Canada, avril 2007.

1.3.2 Références importantes

[Bray]

K. BRAY;

An Introduction to requirements engineering;

Addison-Wesley, 2003.

[Leffingwell]

D. LEFFINGWELL, D. WIDRIG;
Managing software requirements – A use case approach ;
2nd edition, Addison-Wesley, 2003.

[WOL]

Wall-On-Line : l'e-gouvernement wallon,
La boîte à outils : 15 méthodes d'implication des utilisateurs,
http://egov.wallonie.be/boite_outils_methodes/index.htm
(consulté le 11 mai 2007, version en date du 17 décembre 2004)

1.3.3 Références utiles

[Braude2001]

Eric J. BRAUDE;
Software engineering: an object-oriented perspective;
John Wiley & sons, 2001;
ISBN 0-471-32208-3 [QA 76.758 B74 2000]

[Davis2007]

M. Davis ;
Requirements Bibliography,
<http://web.uccs.edu/adavis/UCCS/reqbib.htm>
(consulté le 2007-03-15)

[Hull2004]

E. HULL, K. JACKSON, J. DICK;
Requirements engineering;
2/E, Springer, 2004;
[TA 168 H85 2005]

[IEEE 1233]

IEEE Guide for Developing System Requirements Specifications;
IEEE Std 1233-1998, IEEE, New York, 1998;
[QA 76.76 S73I438 1998]

[IEEE 830]

IEEE Recommended Practice for Software Requirements Specifications;
IEEE Std 830-1998, IEEE, New York, 1998;
[QA 76.76 S73I44 1998]

[IEEE 12207]

Industry Implementation of International Standard ISO/IEC 12207-1995;
IEEE 12207, IEEE, New York, 1995.

[ISO 12207]

ISO/IEC 12207 - Information Technology—Software Life-Cycle Processes;
1995.

[Jackson2001]

Michael JACKSON;
Problem frames;
ACM Press Book, Addison Wesley, 2001;
ISBN 0-201-59627-X.

[Jackson1995]

Michael JACKSON;
Software Requirements & Specifications;
Addison Wesley, 1995.

[Jacobson1994]

Ivar JACOBSON;
Object-Oriented Software Engineering;
ACM Press Book, Addison Wesley, 1994;
ISBN 0-201-54435-0.

[Kotonya1998]

G. KOTONYA and I. SOMMERVILLE;
Requirements engineering: processes and techniques;
John Wiley, 1998;
[QA 76.758 K67 1998]

[Kovitz1998]

B. L. KOVITZ;
Practical Software Requirements: A Manual of Content and Style;
Manning Publications Company, 1998;
[QA 76.76 D47K68 1999]

[Kulak02]

D. Kulak, E. Guiney;
Use Cases: Requirements in Context;
2/E, Addison Wesley Professional, 2004.

[Larman2005]

Craig LARMAN;
Applying UML and patterns - an introduction to object-oriented analysis and design and iterative development;
3/E, Prentice-Hall, Upper Sadel River (NJ), 2005;
ISBN 0-13-148906-2.

[Lauesen2002]

S. Lauesen;
Software Requirements: Styles and Techniques;
Addison Wesley Professional, 2002
[QA 76.754 L38 2002].

[Pressman2005]

PRESSMAN, Roger S.
Software Engineering - A practioner's Approach.
Sixth Edition, McGraw-Hill, 2005.
ISBN 0-07-301933-X.

2 Présentation

2.1 Mise en contexte

-- à venir --

2.2 Fiche signalétique

Objectif

Effectuer l'analyse du processus même de développement des logiciels; utiliser et appliquer les techniques de réingénierie et de réutilisation..

Contenu

Bref aperçu des approches et des normes du développement de logiciels. Étude de quelques cycles de base de développement de logiciels par le paradigme de décision/justification. Illustration sur des exemples. Approches de réingénierie et de rétro-ingénierie des logiciels: limites et perspectives. Techniques de réutilisation des logiciels. Environnements et ateliers de développement assisté des logiciels. Études de cas.

Crédits

3

Organisation

Cours : 3 heures par semaine
Travail personnel : 6 heures par semaine

Référence

<http://www.usherbrooke.ca/programmes/cours/IFT/ift719.htm>

2.3 Objectifs spécifiques

Au terme de cette activité pédagogique, la personne l'ayant réussie sera capable de :

- ◇ planifier et diriger le processus de spécification des exigences ;
- ◇ optimiser les relations entre le processus logiciel et la spécification des exigences ;
- ◇ vérifier et valider les exigences et les documents de spécification ;
- ◇ maîtriser les techniques d'élicitation des exigences ;

- ◇ maîtriser les techniques de spécification des exigences ;
- ◇ générer des scénarios de test fonctionnel ;
- ◇ déduire une architecture technologique.

3 Contenu

Approfondissement de tous les thèmes exposés dans le cadre du cours IGL301.

Approches de réingénierie et de rétro-ingénierie des logiciels: limites et perspectives.

Techniques de réutilisation des logiciels.

Environnements et ateliers de développement assisté des logiciels.

4 Organisation

4.1 Modalités d'enseignement

Les participants sont tenus d'assister aux cours de l'activité IGL301 et de produire sur une base hebdomadaire un rapport synthèse de chacun des modules, complété par un approfondissement fondé sur une recherche personnelle à partir de travaux scientifiques récents.

4.2 Modalités d'évaluation

En plus des deux examens individuels, l'évaluation porte sur :

- ◇ Douze rapports de synthèse ;
- ◇ un séminaire sur un sujet de recherche déterminé conjointement avec l'enseignant.

Le correcteur ou la correctrice peut soustraire jusqu'à 5% de chaque évaluation pour la qualité du français. Des consignes supplémentaires ou des modifications pourront être communiquées au cours du trimestre.

La durée de l'examen période est de 110 minutes et celle de l'examen final est de trois heures – aucune documentation n'est permise et l'usage de la calculatrice est interdit.

Tableau 1 – Sommaire des évaluations

Évaluation	Valeur	Commentaire
examen de mi-session	16 %	Individuel
examen final	40 %	Individuel et récapitulatif
Rapports de synthèse (12 à 2% chacun)	24 %	Individuel
Séminaire	20 %	en équipe de deux ou trois
Total	100 %	

Toute situation éventuelle de plagiat se verrait traitée selon les normes du « Règlement des études » de l'Université de Sherbrooke, disponible à cette adresse :

<http://www.usherbrooke.ca/programmes/etude/>

4.3 Calendrier

Se reporter au calendrier de l'activité IGL301 pour les cours et les examens.

Le rapport de synthèse est à remettre chaque vendredi à 08h30.

Le séminaire doit être présenté au cours de la semaine du 14 mars à un moment à être convenu avec l'enseignant.