



Plan de cours

IFT 719 – Processus du génie logiciel (hiver 2012)

Enseignant

Luc LAVOIE

Courriel : Luc.Lavoie@USherbrooke.ca

Bureau : D4-1010-12

Téléphone : (819) 821-8000 poste 62015

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

Disponibilité : sur rendez-vous.

Horaire

Lundi 08:30 à 10:20 D3-2031

Vendredi 08:30 à 10:20 D3-2031

Version et statut

1.0.0 - en vigueur (2012-01-16)

1	Introduction	2
1.1	Objet et portée du document	2
1.2	Définitions	2
1.3	Références	2
1.4	Références	2
2	Présentation	6
2.1	Mise en contexte	6
2.2	Fiche signalétique	7
2.3	Objectifs spécifiques	7
3	Contenu	8
4	Organisation	8
4.1	Modalités d'enseignement	8
4.2	Modalités d'évaluation	8
4.3	Dispositions relatives au plagiat	10
4.4	Calendrier	11

1 Introduction

1.1 Objet et portée du document

Le document décrit l'activité IFT 719 « **Processus du génie logiciel** » offerte au trimestre d'hiver 2012 par le Département d'informatique de la Faculté des sciences. 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>
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

1.4 Références

1.4.1 Références essentielles

[IGL 301]

COLLECTIF GL.

IGL 301 – Spécification et validation des exigences (notes complémentaires et synthétiques).

<http://pages.usherbrooke.ca/llavoie/enseignement/IGL301>

Département d'informatique, Faculté des sciences, Université de Sherbrooke, Sherbrooke, Canada, janvier 2011.

[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, janvier 2011.

1.4.2 Références importantes

[Bray2002]

K. BRAY.

An Introduction to requirements engineering.

Addison-Wesley, 2002.

ISBN 0-201-76792-9 ; UdeS QA 76.758 B744 2002.

[Elmasri2007]

ELMASRI, Ramez ; NAVATHE, Shamkant B.
Fundamentals of database systems.
Fifth Edition, Pearson Addison Wesley, 2007.
ISBN 0-321-36957-2.

[Leffingwell2003]

D. LEFFINGWELL, D. WIDRIG.
Managing software requirements – A use case approach.
2nd edition, Addison-Wesley, 2003.
ISBN 0-321-12247-X; UdeS QA 76.76 D47L44 2003.

[WOL2004]

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
(version en date du 17 décembre 2004 consultée le 11 mai 2007, disponible maintenant sous
<http://pages.usherbrooke.ca/llavoie/projets/GLOGUS/wall-on-line.pdf>)

1.4.3 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]

[Braude2011]

Eric J. BRAUDE, Michael E. BERNSTEIN.
Software engineering: modern approaches.
John Wiley & sons, 2011.
ISBN 979-0-471-69208-9.

[Davis2007]

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

[GDT]

Grand dictionnaire terminologique.
Office québécois de la langue française.
<http://www.granddictionnaire.com>
(consulté le 2011-12-15).

[Hull2004]

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

[IEEE1233]

IEEE Guide for Developing System Requirements Specifications;
IEEE Std 1233-1998, IEEE, New York, 1998;
[QA 76.76 S73I438 1998 – disponible au comptoir de la bibliothèque de Sciences et Génie]

[IEEE830]

IEEE Recommended Practice for Software Requirements Specifications;
IEEE Std 830-1998, IEEE, New York, 1998;
[QA 76.76 S73I44 1998 – disponible au comptoir de la bibliothèque de Sciences et Génie]

[IEEE12207]

Industry Implementation of International Standard ISO/IEC 12207-1995;
IEEE Std 12207.0-1996, IEEE, New York, 1998.
[QA 76.76 S73I44 1998 – disponible au comptoir de la bibliothèque de Sciences et Génie]

[Jackson1995]

Michael JACKSON;
Software Requirements & Specifications;
Addison Wesley, 1995; ISBN 0-201-87712-0.

[Jackson2001]

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

[Jacobson1994]

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

[Jacobson1999] (traduit en français, voir [Jacobson2000])

Ivar JACOBSON, Grady BOOCH, James RUMBAUGH;
The unified software development process;
Addison-Wesley, 1999; ISBN 0-201-57169-2.

[Jacobson2000] (traduction de [Jacobson1999])

Ivar JACOBSON, Grady BOOCH, James RUMBAUGH;
Le processus unifié de développement logiciel;
Eyrolles, 2000 ; ISBN 2-212-09142-7 ; [UdeS 76.76 D47]3514 2000].

[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].

[Kulak2002]

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].

[Pfleeger2010]

PFLEEGER, Shari Lawrence, Joanne M. ATLEE.
Software Engineering – Theory and Practice.
Fourth Edition, Prentice Hall, 2005; ISBN 978-0-13-606169-4.

[Pressman2005]

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

[Sommerville2007]

SOMMERVILLE, Ian
Software Engineering.
Height Edition, Addison-Wesley, 2007; ISBN 978-0-321-31379-9.

[VanVliet2008]

VAN VLIET, Hans
Software Engineering - Principles and Praticce.
Third Edition, Wiley, 2008; ISBN 978-0-470-03146-9.

2 Présentation

2.1 Mise en contexte

Le génie logiciel traite de la configuration d'une machine universelle (ordinateur) dans le but d'atteindre un objectif spécifique. Le logiciel de configuration peut lui aussi être vu comme une machine, mais il diffère des autres machines en ce sens qu'il est intangible. Le génie logiciel doit son nom et sa constitution comme un domaine de connaissance propre à la tenue d'un séminaire organisé par l'OTAN à Garmisch-Partenkirchen en Autriche en 1968. Böhm et Bauer en sont probablement les parrains.

Le logiciel de configuration d'une machine universelle est désigné sous plusieurs appellations différentes, selon la caractéristique mise de l'avant : logiciel (intangibilité), programme (déterminisme), système (complexité).

Puisqu'on construit généralement un système pour atteindre un but donné, il est préférable de déterminer et de détailler d'abord quel est ce but. Ce qui nous amène à l'ingénierie des exigences, la partie du génie logiciel qui permet de déterminer quel système sera développé.

L'élaboration d'une architecture logicielle découle de cette première étape. Elle a pour but de délimiter les unités organiques qui permettront de réaliser les fonctions du logiciel de façon efficace, traçable et modifiable (pour ne nommer que quelques-unes des propriétés recherchées).

Même si dans certains cas il est possible d'induire l'architecture de la spécification formelle, la vérifiabilité et la validité du produit résultant nécessitent un examen indépendant. Voilà une troisième étape.

Les composants organiques doivent communiquer entre eux et les humains interagir avec certains d'entre eux. L'ingénierie des interfaces machine-machine et des interfaces personne-machine intervient alors dans une quatrième étape.

La conception globale et la conception détaillée, souvent regroupées sous la désignation de programmation, visent à construire effectivement chacun des composants organiques et des composants d'interface. La conception globale s'intéresse aux algorithmes, aux structures de données et à la complexité algorithmique des solutions proposées pour incarner les composants logiciels. La conception détaillée s'intéresse à l'expression des solutions élaborées lors de la conception globale. Parfois considérées comme deux étapes distinctes, parfois comme une seule étape, la conception globale et la conception détaillée permettent la concrétisation du logiciel effectif qui devient alors produit.

Les frontières entre chacune de ces étapes ne sont pas arbitraires, mais ne sont pas toujours tranchées. L'ordre dans lequel elles sont réalisées peut varier considérablement selon qu'il s'agit d'un projet prédictif ou expérimental. L'agencement des activités et des tâches en phases, en vagues, en cascades, en itérations ou en spirales nécessite de nombreuses adaptations. Plusieurs procédés ont donc vu le jour qui, lorsqu'ils sont contextualisés, déterminent les processus de développement effectivement utilisés.

Ces processus de développement sont le plus souvent réalisés dans le contexte d'un projet dont la gestion nécessite elle-même plusieurs processus (gestion de portée, de durée, de coût, des ressources humaines, des risques, etc.). L'interaction entre tous ces processus est alors déterminante pour le succès du projet.

Le cours s'intéressera cependant principalement aux processus techniques, liés au développement proprement dit du logiciel.

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/fiches-cours/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 développement et, plus particulièrement celui d'ingénierie des exigences ;
- ◇ optimiser les relations entre les différents processus logiciels ;
- ◇ contrôler, superviser et maîtriser les processus de développement logiciels.

3 Contenu

1. Revue des principaux procédés de développement logiciel
2. Procédés et processus d'ingénierie des exigences

3. Documentation des processus de développement logiciel
4. Revue des techniques d'exploration
5. Méthodes d'analyse, de spécification et de suivi des exigences
6. Techniques d'analyse, de spécification et de suivi des exigences
7. De la spécification à l'architecture
8. De la spécification aux essais

4 Organisation

4.1 Modalités d'enseignement

Les participants sont invités à assister aux cours de l'activité IGL 301. Les périodes de cours visent à expliquer la matière contenue dans les manuels de référence. L'étudiante, l'étudiant, est responsable d'effectuer préalablement les lectures correspondant au sujet de la semaine.

Les travaux dirigés présentent des exercices individuels ou en groupe selon les exigences du programme et les besoins des étudiantes et des étudiants.

Les travaux pratiques consistent en des prestations nécessitant l'utilisation de concepts, de méthodes et de techniques présentées en cours. Ces travaux ne comprennent pas de programmation.

4.2 Modalités d'évaluation

En plus de l'examen de mi-parcours et de l'examen final, l'évaluation comprend des travaux pratiques accomplis dans le cadre du projet de session. Ces travaux sont réalisés en équipe de deux à trois personnes. Les équipes doivent rester stables pour tous les travaux pratiques. Un bilan de fin de projet, individuel, complète l'évaluation des travaux pratiques. Un rapport de synthèse d'un article de recherche récent complète l'évaluation.

Tableau 1 – Sommaire des évaluations

Évaluation	Valeur	Modalité
Examen de mi-parcours	30 %	Individuel
Examen récapitulatif	40 %	Individuel
TP1 – Démarrage du projet	3 %	En équipe de deux ou trois
TP2 – Exploration	3 %	En équipe de deux ou trois
TP3 – Analyse A	3 %	En équipe de deux ou trois
TP4 – Analyse B	3 %	En équipe de deux ou trois
TP5 – Spécification	3 %	En équipe de deux ou trois
TP6 – Bilan de fin de projet	3 %	Individuel
Rapport de synthèse	12 %	Individuel
Total	100 %	

Tout étudiant, toute étudiante, qui omet de remettre un travail au moment prescrit par l'échéancier doit rencontrer l'enseignant afin de déterminer une nouvelle date de remise. Dans tous les cas, une pénalité de 10 % par jour de retard est imposée.

L'évaluation est faite en tenant compte de la clarté des documents. Conformément aux articles 36, 37 et 38 du règlement facultaire d'évaluation des apprentissages¹, l'enseignant peut retourner à l'étudiante ou à l'étudiant tout travail non conforme aux exigences quant à la qualité de la langue et aux normes de présentation.

En cas de circonstances extraordinaires au-delà du contrôle de l'Université de Sherbrooke et sur décision de celle-ci, l'évaluation des apprentissages de cette activité est sujette à changement.

4.2.1 Projet de session

Voir plan de cours d'IGL 301.

4.2.2 Examens et contrôles

Voir plan de cours d'IGL 301.

4.2.3 Rapport de synthèse d'un article de recherche

Le rapport, comprenant une synthèse et une analyse critique, doit faire l'objet d'une présentation soignée, conforme aux exigences applicables aux mémoires et aux essais². La synthèse de l'article scientifique doit rendre compte des hypothèses, des résultats et de la méthode utilisée par les auteurs. Une analyse critique personnelle doit compléter la synthèse. Le rapport doit être transmis par courriel à l'enseignant.

Le rapport écrit comprend :

- ◇ la référence complète et exacte à l'article ;
- ◇ la présentation de la problématique de recherche ;
- ◇ la présentation de la solution proposée ;
- ◇ la présentation des principaux arguments soutenant la solution ;
- ◇ une critique motivée de l'article dans son ensemble.

Le séminaire doit présenter du rapport. La durée de la présentation et de l'ordre de 45 minutes, incluant une période de questions de 10 minutes. Le support de présentation utilisé lors du séminaire doit être remis le lendemain du séminaire par courriel.

Les modalités et les dates de la présentation orale seront fixées au retour de la semaine de relâche.

¹ http://www.usherbrooke.ca/accueil/documents/politiques/pol_2500-008/pol_evaluation/sciences.html

² <http://www.usherbrooke.ca/informatique/intranet/etudes-superieures/>

4.3 Dispositions relatives au plagiat

Dispositions générales

Toute situation de plagiat sera traitée en conformité, entre autres, avec l'article 8.1.2 du Règlement des études³ de l'Université de Sherbrooke.

Dispositions particulières

Un document dont le texte et la structure se rapportent à des textes intégraux tirés d'un livre, d'une publication scientifique ou même d'un site Internet, doit être référencé adéquatement. Lors de la correction de tout travail individuel ou de groupe, une attention spéciale sera portée au plagiat, défini dans le Règlement des études comme « le fait, dans une activité pédagogique évaluée, de faire passer indument pour siens des passages ou des idées tirés de l'œuvre d'autrui ». Le cas échéant, le plagiat est un délit qui contrevient à l'article 8.1.2 du Règlement des études : « tout acte ou manœuvre visant à tromper quant au rendement scolaire ou quant à la réussite d'une exigence relative à une activité pédagogique ». À titre de sanction disciplinaire, les mesures suivantes peuvent être imposées : a) l'obligation de reprendre un travail, un examen ou une activité pédagogique et b) l'attribution de la note E ou de la note 0 pour un travail, un examen ou une activité évaluée. Tout travail suspecté de plagiat sera transmis au Secrétaire de la Faculté des sciences.

4.4 Calendrier

Tableau 2 – Calendrier des activités

N ^o	Semaine	Activité	Contenu	Évaluation (remises)
1	2012-01-09	cours + TD	0, 1	
2	2012-01-16	cours + TD	2, 3	
3	2012-01-23	cours + TD	4	
4	2012-01-30	cours + TD	4	TP1
5	2012-02-06	cours + TD	5	
6	2012-02-13	cours + TD	6.1 - 6.3	
7	2012-02-20	examen	(0 - 5.11)	examen de mi-parcours
8	2012-02-27	cours + TD	6.4 - 6.5	TP2
9	2012-03-05	relâche	--	
10	2012-03-12	cours + TD	6.6 - 6.7	TP3
11	2012-03-19	cours + TD	6.8 - 6.11	
12	2012-03-26	cours + TD	6.12	TP4
13	2012-04-02	cours + TD	6.13	
14	2012-04-09	cours + TD	7, 8	TP5
15	2012-04-16	cours + TD	9	TP6
16	à déterminer	examen	(1 - 9)	examen récapitulatif
17	à déterminer	séminaire	article de recherche	Rapport et support de présentation

Notes :

³ <http://www.usherbrooke.ca/programmes/etude/>



- Les remises sont les vendredis, jusqu'à 23:59 (l'heure du serveur de courriel universitaire en fait foi).
- Il y a relâche le 6 avril et le 9 avril, une période de récupération sera déterminée en début de trimestre.
- La date de l'examen de mi-parcours est fixée par la Faculté (entre le 20 et le 25 février).
- La date de l'examen final est fixée par la Faculté (entre le 17 et le 28 avril).