

Plan de cours

IGL 301 – Spécification et vérification des exigences (hiver 2010)

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	08:30 à 10:20	D4-2022
Mercredi	08:30 à 10:20	D4-2022

Version et statut

1.0.0 - en vigueur

1	Introduction	2
1.1	Objet et portée du document	2
1.2	Définitions	2
1.3	Références	2
2	Présentation	4
2.1	Mise en contexte	4
2.2	Fiche signalétique	5
2.3	Objectifs spécifiques	5
3	Contenu	6
4	Organisation	7
4.1	Modalités d'enseignement	7
4.2	Modalités d'évaluation	8
4.3	Calendrier	9

1 Introduction

1.1 Objet et portée du document

Le document décrit l'activité IGL 301 « **Spécification et vérification des exigences** » offerte au trimestre d'hiver 2010 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>
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).

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

Département d'informatique, Faculté des sciences, Université de Sherbrooke, Sherbrooke, Canada, janvier 2009.

[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 2009.

1.3.2 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 aussi sous <http://pages.usherbrooke.ca/llavoie/projets/GLOGUS/wall-on-line.pdf>)

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)

[GDT]

Grand dictionnaire terminologique.
Office québécois de la langue française.
<http://www.granddictionnaire.com>
(consulté le 2008-12-15).

[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 12207, IEEE, New York, 1995.
[QA 76.76 S73I44 1998 – disponible au comptoir de la bibliothèque de Sciences et Génie]

[ISO12207]

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

- [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.
- [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].
- [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.
- [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.

Note : nous préférons réserver le terme « spécification des exigences » pour désigner une des activités de l'ingénierie des exigences. Succinctement, la spécification des exigences est l'activité par laquelle on déduit (on conçoit) et met en forme, selon des critères rigoureux, les exigences issues des activités préalables d'exploration et d'analyse. Dans de nombreux contextes, les deux expressions peuvent cependant être utilisées l'une pour l'autre.

2.2 Fiche signalétique

Objectif

Spécifier, valider et vérifier les exigences des clients; en déduire une architecture technologique.

Contenu

Spécifications fonctionnelles et non fonctionnelles. Diagramme de flux de données et modèles de données. Spécification textuelle des exigences. Cas d'utilisation et scénario. Validation des exigences. Génération de scénarios de tests d'acceptation. Élaboration de l'architecture. Présentation des normes de spécification IEEE.

Concomitante

IFT232 – Méthodes de conception orientées objet

Crédits

3

Organisation

Cours : 3 heures par semaine
Travaux dirigés : 1 heure par semaine
Travail personnel : 5 heures par semaine

Référence

<http://www.usherbrooke.ca/fiches-cours/igl301.htm>

2.3 Objectifs spécifiques

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

- ◇ appliquer le processus de spécification des exigences ;
- ◇ établir les relations entre le processus logiciel et la spécification des exigences ;
- ◇ établir la structure d'un document de spécification des exigences ;
- ◇ appliquer les techniques d'exploration des exigences ;
- ◇ appliquer les techniques de spécification des exigences ;
- ◇ vérifier les exigences ;
- ◇ générer des scénarios de test fonctionnel ;
- ◇ déduire une architecture technologique.

3 Contenu

1. Introduction

- 1.1. Mise en contexte historique Bray 1
- 1.2. Terminologie Bray 1, 2 ; GDT
- 1.3. Procédés et processus de développement logiciel Leffingwell 3, Pressman 2-4
 - 1.3.1. Prédicatifs (cascades, V, etc.)
 - 1.3.2. Itératifs (itératif simple, spirale, unifié, etc.)
 - 1.3.3. Agiles (XP, Scrum, etc.)
- 1.4. Spécification : le matériel versus le logiciel Leffingwell 3
- 1.5. Normes IEEE, ISO, militaires et aérospatiales IEEE 830, 1233, 12207

2. Processus de spécification des exigences

- 2.1. Exploration Bray 2, Jacobson2000 2.3
- 2.2. Analyse
- 2.3. Spécification
- 2.4. Conception externe (IPM et IMM)
- 2.5. Design (IPM)

3. Techniques d'exploration

- 3.1. Cartes d'acteurs WOL
- 3.2. Entrevues et questionnaires Bray 3, 9 ; WOL
- 3.3. Ateliers
 - 3.3.1. *Brain storming* Bray 9, Leffingwell 12, WOL
 - 3.3.2. *Mind mapping* et tri par cartes Notes de cours, WOL
 - 3.3.3. Analyse experte (Delphi) WOL
 - 3.3.4. *Focus Group* WOL
 - 3.3.5. *Storyboarding* Leffingwell 13
 - 3.3.6. Survol de quelques autres méthodes en atelier WOL
- 3.4. Analyse de documents Bray 3, 9 ; WOL
- 3.5. Observation et analyse des tâches Bray 3, 9 ; WOL
- 3.6. Méthodes participatives WOL, notes de cours

4. Documentation du processus d'IE

- 4.1. Document de vision, énoncé de portée et mandat Leffingwell 16, GLOGUS
- 4.2. Document de spécification des exigences GLOGUS
- 4.3. Le glossaire GLOGUS
- 4.4. La liste des références GLOGUS

5. Méthodes d'analyse et de spécification

- 5.1. Analyse structurée Bray 4.3, 13.1
 - 5.1.1. Présentation
 - 5.1.2. Quelques exemples avec SA, SADT et SSADM
- 5.2. Analyse orientée objet Leffingwell
 - 5.2.1. Présentation

- 5.2.2. Quelques exemples avec UML
- 5.3. Analyse orientée par le problème
 - 5.3.1. Présentation
 - 5.3.2. Quelques exemples avec les Frames de Jackson
- 5.4. Méthodes formelles
 - 5.4.1. Présentation
 - 5.4.2. Un exemple avec EB3
- 6. Techniques d'analyse et de spécification**
 - 6.1. Présentation
 - 6.2. Langue naturelle
 - 6.3. Diagramme de contexte
 - 6.4. Diagramme de flux de données
 - 6.5. Modèles conceptuels de données
 - 6.5.1. Diagramme entité-relation
 - 6.5.2. Dictionnaire de données
 - 6.5.3. Diagramme entité-relation étendu
 - 6.7. Pseudo-code
 - 6.8. Tables de décision
 - 6.9. Diagrammes état-transition
 - 6.9.1. Automate
 - 6.9.2. Machine à états
 - 6.9.3. *State Chart*
 - 6.9.4. SDL
 - 6.10. Réseau de Petri
 - 6.11. Cas d'utilisation
 - 6.12. Diagrammes statiques d'UML
 - 6.13. Diagrammes dynamiques d'UML
 - 6.14. Maquettage
 - 6.15. Prototypage
- 7. Des spécifications aux essais**
- 8. Des spécifications à l'architecture**
 - 8.1. Modélisation de la structure et du comportement avec UML
 - 8.2. Dérivation des diagrammes de classes
 - 8.3. Dérivation des diagrammes de séquence

4 Organisation

4.1 Modalités d'enseignement

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 des deux examens individuels, l'évaluation porte sur :

- ◇ deux travaux pratiques individuels ;
- ◇ un projet à être réalisés en équipe de deux personnes ou trois personnes.

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 de mi-trimestre est de 110 minutes et celle de l'examen final est de trois heures – aucune documentation n'est permise et l'usage d'appareils informatiques, électroniques ou de communication (ordinateur, calculatrice, téléphone, etc.) est interdit.

Tableau 1 – Sommaire des évaluations

Évaluation	Valeur	Commentaire
Examen de mi-parcours	30 %	Individuel
Examen récapitulatif	40 %	Individuel
TP 1	5 %	Individuel
TP 2	5 %	Individuel
Projet	20 %	En équipe de deux ou trois
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 et du respect de la méthodologie de gestion de projet. 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.

Toute situation de plagiat sera traitée en conformité, entre autres, avec l'article 8.1.2 du Règlement des études² de l'Université de Sherbrooke.

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.

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

² <http://www.usherbrooke.ca/programmes/etude>

4.3 Calendrier

Tableau 2 – Calendrier des activités

N ^o	date	activité	contenu	évaluation
1	2009-01-04	cours	1, 2	
2	2009-01-11	cours + TD	3.1 – 3.3	énoncé TP1
3	2009-01-18	cours + TD	3.3 – 3.6	
4	2009-01-25	cours + TD	4, 6.1, 6.2	remise TP1 / énoncé TP2
5	2009-02-01	cours + TD	5.1 (début), 6.3, 6.4	
6	2009-02-08	cours + TD	5.1 (fin), 6.5 – 6.10	remise TP2
7	2009-02-15	étude		
8	2009-02-22	examen	(1 – 5.1) + (6.1 – 6.10)	examen de mi-trimestre
9	2009-03-02	relâche		
10	2009-03-08	cours + TD	5.2 (début), 6.11	énoncé du projet
11	2009-03-15	cours + TD	5.2 (fin), 6.12, 6.13	
12	2009-03-22	cours + TD	5.3 (début)	
13	2009-03-29	cours + TD	5.3 (fin), 6.14, 6.15	
14	2009-04-05	cours + TD	5.4, 5.5, 7	
15	2009-04-12	cours + TD	8	remise projet
16	à déterminer	examen	(1 – 8)	examen récapitulatif

Notes :

- Les jours de remise sont les mercredis.
- Il y a relâche le 4 janvier (Nouvel An), 5 avril (lundi de Pâques) et le 20 janvier (Carnaval).
- La date de l'examen de mi-trimestre est fixée par la Faculté (le 22 ou le 24 février).
- La date de l'examen final est fixée par la Faculté (entre le 13 et le 23 avril).