

Plan de cours IGL301 – hiver 2007

Spécification et vérification des exigences

Enseignant	Date
Luc LAVOIE	2007-05-13
Statut	Version
En vigueur	1.0.0

1	Introduction	2
1.1	Objet et portée du document	2
1.2	Définitions	2
1.3	Références.....	2
2	Présentation.....	6
2.1	Mise en contexte (à venir)	6
2.2	Fiche signalétique.....	7
2.3	Objectifs spécifiques	7
3	Contenu.....	7
4	Organisation.....	9
4.1	Modalités d'enseignement.....	9
4.2	Modalités d'évaluation.....	9
4.3	Calendrier.....	10

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 IGL301 « **Spécification et vérification des exigences** » offerte au trimestre d'hiver 2007. On y présente les objectifs, le contenu, l'organisation et les modalités d'évaluation du cours.

1.2 Définitions

IEEE	The Institute of Electrical and Electronics Engineers, inc.
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/lavoie/glogus.php>

Département d'informatique, Faculté des sciences, Université de Sherbrooke, Sherbrooke, Canada ; dernière modification : avril 2007 ; consulté le 2007-05-20.

1.3.2 Références importantes

[Bray]

K. BRAY;
An Introduction to requirements engineering;
Addison-Wesley, 2003;
[QA 76.758 B744 2002]

[Leffingwell]

D. LEFFINGWELL, D. WIDRIG;
Managing software requirements – A use case approach ;
2nd edition, Addison-Wesley, 2003;
[QA 76.76 D4L44 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
dernière mise à jour : 17 décembre 2004, consulté le 11mai 2007

1.3.3 Références utiles

[Arnold94]

Robert S. ARNOLD;
Software reengineering;
IEEE Computer Society Press, 1994;
ISBN 0-8186-3272-0.

[Braude01]

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

[Davis07]

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

[Gamma95]

Erich GAMMA, Richard HELM, Ralph JOHNSON, John VLISSIDES;
Design patterns - elements of reusable object-oriented software;
Addison-Wesley, Longman (MA), 1995;
ISBN 0-201-63361-2.

[Hull04]

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.

[Jackson01]

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

[Jackson95]

Michael JACKSON;
Software Requirements & Specifications;
Addison Wesley, 1995;
[QA 76.76 D47]33 1995]

[Jacobson94]

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

[Kotonya98]

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

[Kovitz98]

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.

[Lai97]

Michel LAI;
UML - La notation unifiée de modélisation objet;
Masson, Paris, 1997;
ISBN 2-7296-0659-9.

[Larman04]

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), 2004;
ISBN 0-13-148906-2.

[Lauesen02]

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

[Lavi03]

J. Z. Lavi, J. Kudish;
Systems Modeling & Requirements Specification Using ECSAM: Embedded Computer-Based Systems Analysis;
Dorset House, 2003;
[TK7895 .E42 L38 2004]

[Maciaszek03]

L. Maciaszek;
Requirements Analysis and Systems Design: Developing Information Systems with UML;
2/E, Addison-Wesley, 2004.

[Pressman2005]

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

[Robertson99]

S. Robertson, J. Robertson;
Mastering the Requirements Process;
Addison-Wesley, 1999;
[QA 76.76 D47R636 1999]

[Rumbaugh91]

James RUMBAUGH, Michael BLAHA, William PREMERLANI, Frederick EDDY, William LORENSEN;
Object-oriented Modeling and design;
Prentice-Hall, Englewood Cliffs (NJ), 1991;
ISBN 0-13-629841-9.

[Sawyer97]

P. Sawyer, I. Sommerville;
Requirements Engineering: A good practice guide;
John Wiley & Sons, 1997;
[QA 76.76 D47S658 1997]

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. Son nom et sa constitution comme un domaine de connaissance propre coïncide avec un séminaire organisé par l'OTAN à Garmisch-Partenkirchen en Autriche en 1968. Böhm et Bauer en sont probablement les parrains.

Avant de construire une machine ou un artefact qui doit atteindre un but donné, il est préférable de déterminer et de détailler 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 utiliser 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.

2.3 Objectifs spécifiques

À la fin de cette activité pédagogique, l'étudiante ou l'étudiant sera capable de :

- ◇ Connaître le processus de spécification des exigences
- ◇ Connaître les relations entre le processus logiciel et la spécification des exigences
- ◇ Connaître la structure d'un document de spécification des exigences
- ◇ Maîtriser les techniques d'élicitation des exigences
- ◇ Maîtriser 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. Processus logiciels | Leffingwell 3 |
| 1.4. Spécification : le matériel versus le logiciel | Leffingwell 3 |
| 1.5. Structure d'un document de spécification des exigences | GLOGUS |
| 1.6. Normes IEEE, ISO, militaires et aérospatiales | IEEE 830, 1233 |

2. Processus de spécification des exigences

Bray 2

3. Techniques d'exploration

- | | |
|----------------------------------|-----------------|
| 3.1. Entrevues et questionnaires | Bray 3, 9 ; WOL |
| 3.2. Ateliers | Bray 3, 9 ; WOL |
| 3.2.1. <i>Brain storming</i> | |

3.2.2. <i>Mind mapping</i>	
3.2.3. Delphi	
3.2.4. <i>Focus Group</i>	
3.2.5. <i>etc.</i>	
3.3. Analyse de documents	Bray 3, 9 ; WOL
3.4. Observation et analyse des tâches	Bray 3, 9 ; WOL
3.5. Storyboarding	Leffingwell 13
4. Document de vision	Leffingwell 16
5. Technique de spécification	
5.1. Présentation	Bray 4, 5
5.2. Analyse structurée	Bray 4.3, 13.1
5.2.1. Diagramme de contexte	
5.2.2. Diagramme de flux de données	
5.2.3. Modèle conceptuel de données	
5.2.4. Dictionnaire de données	
5.3. Langue naturelle	Bray 14
5.4. Tables de décision	Bray 14
5.5. Pseudo-code	Bray 14
5.6. Diagrammes état-transition	Bray 12.6
5.6.1. Automate	
5.6.2. Machine à états	
5.6.3. <i>State Chart</i>	
5.6.4. SDL	
5.7. Diagramme de structure JSD et méthode EB3	Bray 13.4
5.8. Diagramme d'activités	Leffingwell 24
5.9. Réseau de Petri	Bray 12.7
5.10. Maquettage	Notes de cours
5.11. Prototypage	Notes de cours
5.12. Exigences fonctionnelles	Notes de cours
6. De la spécification à l'architecture	Leffingwell 25
6.1. Modélisation de la structure et du comportement avec UML	
6.2. Diagramme de classes	
6.3. Diagramme de séquence	
7. Génération d'essais depuis la spécification	Notes de cours

4 Organisation

4.1 Modalités d'enseignement

L'activité est offerte en mode intensif, à raison 15 jours de cours répartis sur trois semaines.

Les périodes de cours hebdomadaires serviront aux exposés théoriques et aux exemples. Les travaux dirigés présentent des exercices individuels ou en groupe selon les exigences du programme et les besoins des étudiantes et des étudiants.

Les séances de cours durent trois heures et les séances de travaux dirigés (TD) deux heures.

4.2 Modalités d'évaluation

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

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

Le correcteur ou la correctrice peut soustraire jusqu'à 5% de chaque évaluation pour la qualité du français.

La durée des examens 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	20 %	Individuel
examen final	40 %	Individuel et récapitulatif
TP 1	5 %	Individuel ou équipe de deux
TP 2	5 %	Individuel ou équipe de deux
TP 3	5 %	Individuel ou équipe de deux
TP 4	5 %	Individuel ou équipe de deux
Projet	20 %	Individuel ou équipe de deux
Total	100 %	

En raison du mode de prestation intensif, le dernier travail est réalisé après l'examen final et un délai d'une semaine est accordé.

4.3 Calendrier

Tableau 2 – Calendrier des activités

N°	jour	date	activité	contenu	travaux
1.	lu	2007-05-14	cours + TD	1, 2	
2.	ma	2007-05-15	cours + TD	3.1, 3.2	énoncé TP1
3.	me	2007-05-16	cours + TD	3.3, 3.4	
4.	je	2007-05-17	cours + TD	3.5, 4	remise TP1 / énoncé TP2
5.	ve	2007-05-18	cours	5.1, 5.2	
6.	lu	2007-05-21	cours + TD	5.3, 5.4	remise TP2
7.	ma	2007-05-22	cours	5.5	
8.	me	2007-05-23	examen de mi-session	(1 – 5.2)	
9.	je	2007-05-24	cours + TD	5.6	énoncé TP3 / énoncé du projet
10.	ve	2007-05-25	cours	5.7	
11.	lu	2007-05-28	cours + TD	5.8	remise TP3 / énoncé TP4
12.	ma	2007-05-29	cours + TD	5.9, 5.10	
13.	me	2007-05-30	cours + TD	5.11-5.13	
14.	je	2007-05-31	cours	6, 7	remise TP4
15.	lu	2007-06-04	examen final	(1 – 7)	
16.	ma	2007-06-05	remise du projet		