
Plan de cours IFT424

Laboratoire de génie logiciel – été 2005

Enseignant

Luc Lavoie

Luc.A.Lavoie@USherbrooke.ca

Date

2005-05-11

Statut

Version 1.0.1, en vigueur

1	Introduction	2
1.1	Objet et portée du document.....	2
1.2	Références	2
1.3	Définitions.....	5
1.4	Mise en contexte.....	5
1.5	Vue d'ensemble du document	5
2	Objectifs.....	5
2.1	Fiche signalétique	5
2.2	Objectifs spécifiques	6
3	Organisation	6
3.1	Modalités d'enseignement.....	6
3.2	Modalités d'évaluation	6
3.3	Calendrier	7
3.4	Modalités d'organisation	7
4	Contenu.....	8

1 Introduction

1.1 Objet et portée du document

Le document s'adresse aux étudiants inscrits à l'activité IFT424 « Laboratoire de génie logiciel » du trimestre d'été 2005. On y présente les objectifs, le contenu et l'organisation de l'activité.

1.2 Références

[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]

[Bray03]

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

[Hull04]

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

[IEEE Std 1233-1998]

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

[IEEE Std 830-1998]

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

[IEEE/EIA 12207]

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

[ISO/IEC 12207-1995]

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]

[Leffingwell03]

D. Leffingwell, D. Widrig;
Managing Software Requirements: A Use Case Approach;
2/E, Addison Wesley, 2003;
[QA 76.76 D47L44 2003]

[Maciaszek03]

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

[Pressman92]

Rober S. Pressman;
Software Engineering;
Mc Graw Hill, 1992;
ISBN 0-07-050814-3.

[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]

1.3 Définitions

CDE	contrat d'équipe.
MDS	mandat de service.
PAQ	plan d'assurance de la qualité.
PGC	plan de gestion de configuration (IEEE CMP <i>configuration management plan</i>).
PGP	plan de gestion de projet (IEEE PMP <i>project management plan</i>).
SAS	spécification d'architecture du système.
SES	spécification des exigences du système (IEEE SRS <i>software requirements specification</i>).

1.4 Mise en contexte

L'activité IFT424 « Laboratoire de génie logiciel » permet aux participants de mettre en pratique les enseignements des cours antérieurs du programme d'informatique de gestion par la réalisation d'un petit projet en équipe de 5 à 7 personnes. L'activité de trois crédits (135 heures de travail personnel) est divisée en trois parties :

- ◇ un cours magistral où sont présentés le cycle de vie d'un produit logiciel et les éléments constitutifs d'un dossier de projet de développement logiciel (12 heures de cours magistral, 12 heures de lecture et d'études pour un total de 24 heures) ;
- ◇ la réalisation du projet proprement dit, incluant la préparation et la présentation d'un exposé oral (111 heures).

1.5 Vue d'ensemble du document

Le plan est divisé comme suit :

- ◇ Introduction : la présente section.
- ◇ Objectifs : description des objectifs de l'activité.
- ◇ Organisation : organisation de l'activité, modalités d'évaluation, calendrier.
- ◇ Contenu : inventaire du contenu des cours magistraux.

2 Objectifs

2.1 Fiche signalétique

Objectif

Être capable d'organiser une équipe de projet informatique et de produire efficacement un bien livrable de haute qualité demandé par un utilisateur typique.

Contenu

Organisation d'une équipe de projet informatique. Planification et contrôle du travail. Analyse de besoins. Révision structurée. Outils et normes de documentation. Réalisation, en équipe, d'un dossier d'analyse et de conception sur un projet soumis par la professeure ou le professeur.

Préalable

IFT 324

2.2 Objectifs spécifiques

- ◇ Connaître les principales étapes de la gestion d'un projet de développement logiciel.
- ◇ Être capable d'identifier les processus, les activités et les phases du cycle de vie d'un produit logiciel.
- ◇ Être capable de constituer un dossier de projet de développement logiciel.
- ◇ Être capable d'évaluer la qualité d'un livrable.
- ◇ Être capable d'évaluer les ressources requises pour la production d'un livrable simple.
- ◇ Être capable de travailler au sein d'une équipe en respectant un plan de projet.
- ◇ Être capable d'insérer la démarche de conception dans le cadre général du génie logiciel.

3 Organisation

3.1 Modalités d'enseignement

12 exposés magistraux d'une heure sont prévus selon le calendrier présenté en 3.3.

À compter de la deuxième semaine, chaque équipe est tenue d'organiser une rencontre hebdomadaire de suivi de projet réunissant tous les membres de l'équipe et l'enseignant. Cette rencontre, d'une durée approximative d'une heure doit être bien préparée et tous les documents requis, transmis au professeur au moins 24 heures à l'avance.

Au sein de l'équipe, chaque membre est responsable individuellement d'une partie des livrables. Il appartient à chaque équipe de définir de façon équitable les responsabilités de chacun des membres. Chaque équipe doit désigner :

- ◇ un chef d'équipe, responsable notamment de l'élaboration et de la mise à jour du plan de projet, de la collecte et de la synthèse des feuilles de temps, des communications avec l'enseignant ;
- ◇ un adjoint au chef d'équipe, responsable de seconder voire de remplacer le chef d'équipe dans toutes les situations qui le commandent ;
- ◇ un secrétaire, responsable notamment de l'élaboration du plan de gestion de configuration, de son application et de la production finale des livrables.

3.2 Modalités d'évaluation

L'évaluation comprend une partie individuelle (70%) et une partie collective (30%) :

Évaluation individuelle :

- ◇ livrables spécifiques (50%) ;
- ◇ assiduité aux réunions, transmission hebdomadaire de la feuille de temps (10%) ;
- ◇ évaluation par les pairs (10%).

Évaluation collective :

- ◇ présentation du bilan de fin de projet (10%) ;
- ◇ réalisation du projet dans son ensemble (20%).

3.3 Calendrier

La planification est présentée de façon hebdomadaire, en prenant le mardi (jour usuel du cours) comme référence :

Semaine	Début	Fin	Cours	Autres activités
1	2005-05-02	2005-05-07	mardi (3)	organisation des équipes
2	2005-05-09	2005-05-14	mardi (3)	choix du projet ; remise du CDE
3	2005-05-16	2005-05-21	mardi (2)	suivi ; remise du MDS
4	2005-05-23	2005-05-28	mardi (2)	suivi ; remise du PGP
5	2005-05-30	2005-06-04	mardi (2)	suivi ; esquisse de la SES
6	2005-06-06	2005-06-11	--	suivi ; remise du PGC
7	2005-06-13	2005-06-18	--	suivi ; remise de la SES ; esquisse de la SAS
8	2005-06-20	2005-06-25	--	suivi
9	2005-06-27	2005-07-02	--	suivi ; remise de la SAS
10	2005-07-04	2005-07-09	--	suivi
11	2005-07-11	2005-07-16	lundi (2) mardi (3) mercredi (2)	présentation des projets : le lundi 11 juillet, de 10:30 à 12:30, sera exceptionnellement mis à contribution
12	2005-07-18	2005-07-23	--	finalisation des livrables
13	2005-07-25	2005-07-30	--	finalisation des livrables
14	2005-08-01	2005-08-06	--	finalisation des livrables
15	2005-08-08	2005-08-13	--	remise des livrables

3.4 Modalités d'organisation

- ◇ Toutes les remises intérimaires doivent être faites par courriel. La remise finale doit être faite sur cédérom ; les plans et les documents de spécification qu'elle comprend doivent être imprimés, rassemblés dans un classeur (« cartable ») et joints au cédérom.
- ◇ Tout courriel adressé à l'enseignant ou au à l'auxiliaire doit comprendre un objet débutant par « IFT424 »

4 Contenu

1. Introduction

- 1.1. Objet et portée du cours
- 1.2. Documents de référence
- 1.3. Définitions
- 1.4. Préalables
- 1.5. Présentation du contenu
- 1.6. Organisation

2. Présentation

- 2.1. Historique
- 2.2. Démarche qualité
- 2.3. Processus qualité
- 2.4. Normes courantes
- 2.5. Assurance de la qualité versus contrôle de la qualité
- 2.6. Vérification versus validation
- 2.7. Audit versus inspection
- 2.8. Critères de qualité
- 2.9. Cycle de vie
 - 2.9.1. Exploration
 - 2.9.2. Analyse
 - 2.9.3. Architecture
 - 2.9.4. Conception
 - 2.9.5. Mise en oeuvre
 - 2.9.6. Mise en service
 - 2.9.7. Entretien
 - 2.9.8. Retrait de service
- 2.10. Processus continus
 - 2.10.1. Gestion de projet
 - 2.10.2. Gestion de configuration
 - 2.10.3. Documentation
 - 2.10.4. Vérification et validation
 - 2.10.5. Mesure et support logistique
 - 2.10.6. Assurance et contrôle de qualité
- 2.11. Les normes applicables
- 2.12. Les critères applicables

3. Gestion de projet

- 3.1. Planification et programme
- 3.2. Estimation
- 3.3. Outils et méthodes
- 3.4. Intégration au cycle de vie du logiciel
- 3.5. Documentation
- 3.6. Revues

3.7. Facteurs humains

4. Exploration

- Besoins
- Opportunité.
- Faisabilité
- Vérification et validation
- Intégration au cycle de vie du logiciel
- Documentation
- Revue
- Facteurs humains

5. Gestion de configuration et documentation

- Outils et méthodes
- Intégration au cycle de vie du logiciel
- Estimation
- Documentation
- Revue
- Facteurs humains

6. Vérification, validation, assurance et contrôle de qualité

- Outils et méthodes
- Intégration au cycle de vie du logiciel
- Estimation
- Documentation
- Revue
- Facteurs humains

7. Analyse

- Vérification et validation
- Intégration au cycle de vie du logiciel
- Documentation
- Revue
- Facteurs humains

8. Architecture

- Vérification et validation
- Intégration au cycle de vie du logiciel
- Documentation
- Revue
- Facteurs humains

9. Conception et mise en oeuvre

- Vérification et validation
- Intégration au cycle de vie du logiciel
- Niveaux (système, classe, méthode)
- Mode (fermé, ouvert)

Documentation
Revue
Facteurs humains

10. Essais

Vérification et validation
Niveaux (système, classe, méthode)
Mode (fermé, ouvert)
Intégration au cycle de vie du logiciel
Documentation
Revue
Facteurs humains

11. Mesure et support logistique

Outils et méthodes
Intégration au cycle de vie du logiciel
Estimation
Documentation
Revue
Facteurs humains

12. Critères et techniques de conception

12.1. Présentation des principaux critères

- 12.1.1. Maintenabilité
- 12.1.2. Disponibilité
- 12.1.3. Sécurité
- 12.1.4. Robustesse
- 12.1.5. Flexibilité
- 12.1.6. Coût
- 12.1.7. Performance
- 12.1.8. Interopérabilité
- 12.1.9. Autres critères

12.2. Présentation des principales techniques

- 12.2.1. Découplage
- 12.2.2. Cohésion structurelle
- 12.2.3. Cohésion logique
- 12.2.4. Cohésion temporelle
- 12.2.5. Cohésion procédurale
- 12.2.6. Cohésion fonctionnelle
- 12.2.7. Traçabilité
- 12.2.8. Autres techniques

IFT424 Laboratoire de génie logiciel
été 2005
Questionnaire

Nom

Courriel

Matricule

Programme

IFT 324

Quand avez-vous suivi le cours IFT324 ?

Qui était l'enseignant ?

Quel résultat avez-vous obtenu ?

Quels sont les éléments que vous avez le mieux assimilés ?

Quels sont les éléments que vous avez le moins bien assimilés ?

Quels sont les éléments qui vous ont servis en stage ?

Quels sont les éléments que vous auriez désiré apprendre et qui vous auraient été utiles en stage ?

Vocabulaire

Définissez brièvement les termes suivants :

Analyse

Conception

Cycle de vie

Exigence

Vérification

Validation
