

---

## Plan d'activité didactique IFT424

Laboratoire de génie logiciel – été 2004

### Enseignant

Luc Lavoie

Luc.A.Lavoie@USherbrooke.ca

### Date

2004-04-29

### Statut

Version 1.0.0, en vigueur

---

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

# 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 2004. On y présente les objectifs, le contenu et l'organisation.

## 1.2 Références

[Arnold94]

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

[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.

[Jackson01]

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

[Jacobson94]

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

[Lai97]

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

[Larman98]

Craig LARMAN;  
*Applying UML and patterns - an introduction to object-oriented analysis and design*;  
Prentice-Hall, Upper Sadel River (NJ), 1998;  
ISBN 0-13-748880-7.

[Pressman92]

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

[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.

### 1.3 Définitions

Non applicable.

### 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 6 à 8 personnes. L'activité de trois crédits (135 heures) 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 (16 heures de cours magistral, 8 heures de lecture et d'études pour un total de 24 heures) ;
- ◇ la réalisation du projet proprement dit (105 heures) ;
- ◇ la préparation et la présentation d'un exposé oral (6 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

16 exposés magistraux d'une heure sont prévus dont 6 les deux premières semaines et 10 par la suite selon le calendrier présenté en 3.3.

À compter de la troisième 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 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 :

- ◇ 2004-04-27 : cours mardi et jeudi, mercredi : organisation des équipes ;
- ◇ 2004-05-04 : cours mardi et jeudi, mercredi : choix du projet ;
- ◇ 2005-05-11 : cours mardi, rencontre de suivi de projet mercredi et jeudi ;
- ◇ 2005-05-18 : cours mardi, rencontre de suivi de projet mercredi et jeudi ;
- ◇ 2005-05-25 : cours mardi, rencontre de suivi de projet mercredi et jeudi ;
- ◇ 2005-06-01 : cours mardi, rencontre de suivi de projet mercredi et jeudi ;
- ◇ 2005-06-08 : cours mardi, rencontre de suivi de projet mercredi et jeudi ;
- ◇ 2005-06-15 : relâche ;
- ◇ 2005-06-22 : relâche ;
- ◇ 2005-06-14 : cours mardi, rencontre de suivi de projet mercredi et jeudi ;
- ◇ 2005-06-29 : cours mardi, rencontre de suivi de projet mercredi et jeudi ;
- ◇ 2005-07-06 : cours mardi, rencontre de suivi de projet mercredi et jeudi ;
- ◇ 2005-07-13 : cours mardi, rencontre de suivi de projet mercredi et jeudi ;
- ◇ 2005-07-20 : cours mardi, rencontre de suivi de projet mercredi et jeudi ;
- ◇ 2005-07-27 : présentation des projets, lundi, mercredi et jeudi ;
- ◇ 2005-08-03 : remise des travaux et fin de l'activité.

## 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. Exploration**

- 3.1. Besoins
- 3.2. Opportunité.
- 3.3. Faisabilité
- 3.4. Vérification et validation
- 3.5. Intégration au cycle de vie du logiciel
- 3.6. Documentation
- 3.7. Revues
- 3.8. Facteurs humains

### **4. Analyse**

- 4.1. Vérification et validation
- 4.2. Intégration au cycle de vie du logiciel
- 4.3. Documentation
- 4.4. Revues
- 4.5. Facteurs humains

### **5. Architecture**

- 5.1. Vérification et validation
- 5.2. Intégration au cycle de vie du logiciel
- 5.3. Documentation
- 5.4. Revues
- 5.5. Facteurs humains

### **6. Conception et mise en oeuvre**

- 6.1. Vérification et validation

- 6.2. Intégration au cycle de vie du logiciel
- 6.3. Niveaux (système, classe, méthode)
- 6.4. Mode (fermé, ouvert)
- 6.5. Documentation
- 6.6. Revues
- 6.7. Facteurs humains

## **7. Essais**

- 7.1. Vérification et validation
- 7.2. Niveaux (système, classe, méthode)
- 7.3. Mode (fermé, ouvert)
- 7.4. Intégration au cycle de vie du logiciel
- 7.5. Documentation
- 7.6. Revues
- 7.7. Facteurs humains

## **8. Gestion de projet**

- 8.1. Planification et programme
- 8.2. Estimation
- 8.3. Outils et méthodes
- 8.4. Intégration au cycle de vie du logiciel
- 8.5. Documentation
- 8.6. Revues
- 8.7. Facteurs humains

## **9. Gestion de configuration**

- 9.1. Outils et méthodes
- 9.2. Intégration au cycle de vie du logiciel
- 9.3. Estimation
- 9.4. Documentation
- 9.5. Revues
- 9.6. Facteurs humains

## **10. Documentation**

- 10.1. Outils et méthodes
- 10.2. Intégration au cycle de vie du logiciel
- 10.3. Estimation
- 10.4. Documentation
- 10.5. Revues
- 10.6. Facteurs humains

## **11. Vérification et validation**

- 11.1. Outils et méthodes
- 11.2. Intégration au cycle de vie du logiciel
- 11.3. Estimation
- 11.4. Documentation
- 11.5. Revues

11.6. Facteurs humains

## **12. Mesure et support logistique**

- 12.1. Outils et méthodes
- 12.2. Intégration au cycle de vie du logiciel
- 12.3. Estimation
- 12.4. Documentation
- 12.5. Revues
- 12.6. Facteurs humains

## **13. Assurance et contrôle de qualité**

- 13.1. Outils et méthodes
- 13.2. Intégration au cycle de vie du logiciel
- 13.3. Estimation
- 13.4. Documentation
- 13.5. Revues
- 13.6. Facteurs humains

## **14. Critères et techniques de conception**

- 14.1. Présentation des principaux critères
  - 14.1.1. Maintenabilité
  - 14.1.2. Disponibilité
  - 14.1.3. Sécurité
  - 14.1.4. Robustesse
  - 14.1.5. Flexibilité
  - 14.1.6. Coût
  - 14.1.7. Performance
  - 14.1.8. Interopérabilité
  - 14.1.9. Autres critères
- 14.2. Présentation des principales techniques
  - 14.2.1. Découplage
  - 14.2.2. Cohésion structurelle
  - 14.2.3. Cohésion logique
  - 14.2.4. Cohésion temporelle
  - 14.2.5. Cohésion procédurale
  - 14.2.6. Cohésion fonctionnelle
  - 14.2.7. Traçabilité
  - 14.2.8. Autres techniques



---

IFT424 Laboratoire de génie logiciel  
été 2004  
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 désirés apprendre et qui vous auraient été utiles en stage ?

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

*Vocabulaire*

Définissez brièvement les termes suivants :

Analyse

---

---

---

Conception

---

---

---

Cycle de vie

---

---

---

Exigence

---

---

---

Vérification

---

---

---

Validation

---

---

---