

Projet éducatif  
Formation en génie logiciel



ÉCOLE  
POLYTECHNIQUE  
MONTREAL

*Le génie  
sans frontières*

Département de génie informatique

23 décembre 2004

# Table des matières

<b>Introduction</b>	<b>1</b>
0.1 Programme de génie logiciel . . . . .	1
0.2 Département de génie informatique . . . . .	1
0.3 Équipe pédagogique de génie logiciel . . . . .	1
0.4 Organisation du document . . . . .	1
<b>1 Vision et objectifs</b>	<b>2</b>
1.1 Analyse de la situation actuelle . . . . .	2
1.1.1 Bref historique du programme de génie logiciel . . . . .	2
1.1.2 Analyse du tronc commun actuel . . . . .	3
1.1.3 Besoin de révision des cours du curriculum . . . . .	4
1.1.4 Les stages en génie logiciel . . . . .	6
1.1.5 Forces et faiblesses actuelles . . . . .	7
1.2 Valeurs poursuivies . . . . .	10
1.3 Définition de l'ingénieur . . . . .	10
1.3.1 Le rôle de l'ingénieur logiciel . . . . .	10
1.3.2 Tâches de l'ingénieur logiciel . . . . .	11
1.3.3 Particularités du génie logiciel à l'École . . . . .	11
1.4 Compétences prônées par le nouveau programme . . . . .	11
1.4.1 Compétences spécifiques au génie logiciel . . . . .	11
1.4.2 Compétences transversales . . . . .	12
1.5 État du consensus des professeurs . . . . .	12
<b>2 Tableau des cours du programme</b>	<b>14</b>
2.1 Tableau du cheminement . . . . .	15
2.1.1 La première année . . . . .	17
2.1.2 La deuxième année . . . . .	19
2.1.3 La troisième année . . . . .	21
2.1.4 La quatrième année . . . . .	23
2.1.5 Les orientations et les concentrations . . . . .	25
2.2 Principes généraux . . . . .	26
2.2.1 Le balisage du nouveau programme . . . . .	26

2.2.2	Les cours de spécialité en première année . . . . .	30
2.2.3	Les mathématiques dans le nouveau programme . . . . .	30
2.2.4	Les études complémentaires dans le nouveau programme . . . . .	31
2.2.5	Les sciences fondamentales dans le nouveau programme . . . . .	31
2.2.6	Les cours mis en place conjointement avec le programme de génie informatique . . . . .	32
2.2.7	Les cours spécifiques au génie logiciel . . . . .	32
2.2.8	Les cours offerts aux autres départements . . . . .	32
<b>3</b>	<b>Forces incontournables citées dans le Cahier des charges</b>	<b>35</b>
3.1	Formation scientifique solide . . . . .	35
3.2	Formation à la conception . . . . .	36
3.3	Habilités personnelles et relationnelles . . . . .	38
3.4	Aspect international du programme . . . . .	39
<b>4</b>	<b>Autres composantes du programme</b>	<b>41</b>
4.1	Intégration des matières . . . . .	41
4.2	Caractère pratique accentué . . . . .	41
4.3	Réduction de la charge de travail étudiante et du contenu dans les cours . . . . .	42
4.3.1	Les cours de programmation . . . . .	42
4.3.2	Les mathématiques spécialisées . . . . .	43
4.4	Méthodes pédagogiques et étudiants responsables . . . . .	44
4.4.1	La grille-horaire et l'absentéisme . . . . .	44
4.5	Encadrement . . . . .	45
4.6	Évaluation des apprentissages . . . . .	46
4.6.1	Schéma en J . . . . .	46
4.7	Projets intégrateurs . . . . .	48
4.7.1	Les thèmes des projets intégrateurs . . . . .	49
4.7.2	La pierre angulaire de la quatrième année . . . . .	50
4.8	Stages . . . . .	51
4.9	Mécanismes d'évaluation continue du programme . . . . .	52
4.10	Modalités pour le passage aux études supérieures . . . . .	52
	<b>Bibliographie</b>	<b>53</b>
<b>A</b>	<b>Description des diverses catégories dans le balisage</b>	<b>54</b>
A.1	Bloc général . . . . .	54
A.2	Bloc GI/GL . . . . .	55
<b>B</b>	<b>Enquête du Conseil des ressources humaines du logiciel</b>	<b>56</b>
	<b>Index</b>	<b>58</b>

# Table des figures

1.1	Comparaison des cours du programme actuel avec certaines universités canadiennes	9
2.1	Code de couleur utilisé pour le programme de génie logiciel	14
2.2	Tableau global du nouveau programme de génie logiciel	16
2.3	Première année du nouveau programme de génie logiciel	18
2.4	Deuxième année du nouveau programme de génie logiciel	20
2.5	Troisième année du nouveau programme de génie logiciel	22
2.6	Quatrième année du nouveau programme de génie logiciel	24
2.7	Balisage des programmes de génie logiciel et de génie informatique	27
2.8	Balisage du programme de génie logiciel	29
2.9	Mathématiques dans le nouveau programme de génie logiciel	30
2.10	Études complémentaires dans le nouveau programme de génie logiciel	31
2.11	Sciences fondamentales dans le nouveau programme de génie logiciel	31
2.12	Cours mis en place avec le programme de génie informatique	33
2.13	Cours spécifiques au programme de génie logiciel	34
3.1	Pourcentage des employés qui jugent “très importantes” les compétences données (secteur privé)	37
3.2	Pourcentage des employés qui jugent “très importantes” les aptitudes données (secteur privé)	39
4.1	Schéma en J : arrimage de trois cours pour assurer une continuité intertrimestre	47
4.2	Thèmes des projets intégrateurs du nouveau programme de génie logiciel	49
B.1	Répartition (%) de la main d’oeuvre en TI par profession	57

## Liste des tableaux

2.1	Cours de 1 <sup>re</sup> année en génie logiciel . . . . .	17
2.2	Cours de 2 <sup>e</sup> année en génie logiciel . . . . .	19
2.3	Cours de 3 <sup>e</sup> année en génie logiciel . . . . .	21
2.4	Cours de 4 <sup>e</sup> année en génie logiciel . . . . .	23
4.1	Semaine typique à l'hiver H1 . . . . .	45
4.2	Cours dont l'évaluation sera partagée . . . . .	48
4.3	Maturité acquise dans les projets intégrateurs annuels . . . . .	50
4.4	Volets techniques accentués dans les projets intégrateurs annuels . . . . .	50

# Introduction

## 0.1 Programme de génie logiciel

Ce programme est le plus récent programme du baccalauréat en génie à l'École Polytechnique de Montréal. Démarré à l'automne 2001, il diplômera donc ses premiers étudiants à l'hiver 2005.

## 0.2 Département de génie informatique

Ce département est aussi le plus récent département créé dans les murs de l'École Polytechnique de Montréal. Créé en 2001 et issu de la scission du département de génie électrique et de génie informatique, il compte présentement 28 professeurs pour une population étudiante frôlant le millier.

## 0.3 Équipe pédagogique de génie logiciel

- Yves BOUDREAU, responsable projets et formule pédagogique
- Michel DESMARAIS, cohérence des cours spécifiques du génie logiciel
- Michel GAGNON, concepts de base et curriculum de 1<sup>re</sup> année
- Gilles PESANT, respect des normes du BCAPAPI et du cahier des charges
- Robert ROY, responsable actuel de l'équipe

Support technique :

- Éric GERMAIN, analyste au département.

## 0.4 Organisation du document

Le document utilise un système de notation pour les sigles de cours qui est arbitraire. Des sigles ont été octroyés afin de faciliter la lisibilité des graphiques et d'interrelier rapidement la place des différents cours dans le curriculum. Un index des sigles est contenu à la fin du document afin de permettre au lecteur de localiser facilement une information particulière sur un cours donné.

# Chapitre 1

## Vision et objectifs

### 1.1 Analyse de la situation actuelle

#### 1.1.1 Bref historique du programme de génie logiciel

Démarré à l'automne 2001, le programme avant-gardiste de génie logiciel était contingenté à une classe de 60 étudiants et devait initialement intégrer l'ordinateur portable dans la formation. Depuis ses débuts, le programme offre et soutient une filière coopérative (avec stages obligatoires) ou une filière régulière (avec stages optionnels) au choix de l'étudiant, avec un tronc de spécialité de 71 crédits et des orientations, certaines restant à définir. Le programme n'a encore aucun diplômé et n'est pas encore accrédité.

#### **L'intégration de l'ordinateur portable : expérience d'un changement de paradigme**

La formule ESSAIM avait retenu un système de location obligatoire dans lequel les étudiants recevaient chaque année un modèle très récent d'ordinateur portable. Les professeurs et coordinateurs impliqués dans la formule ESSAIM ont alors collaboré pour utiliser pleinement les ressources (exercices, soumissions Web de travaux, entrelacement en juste à temps des activités de laboratoire et de cours théorique). L'esprit d'équipe et la qualité de l'organisation pédagogique ont été des éléments clés du succès de cette première année expérimentale. La formule, exclusive au programme de génie logiciel, devait se déployer dans d'autres programmes, mais les contraintes budgétaires et humaines n'ont pas permis de poursuivre cette intégration. L'obligation de louer un portable est devenue facultative pour les cohortes subséquentes. Depuis, la formule ESSAIM a été remplacée par une nouvelle formule techno-pédagogique, la formule A2P2 : les étudiants de n'importe quel programme qui en font partie s'engagent à se procurer un ordinateur portable et s'en servent activement dans certains cours ciblés de la première année du baccalauréat en ingénierie. La formule A2P2 a aussi été abandonnée.

## Le comité des programmes du département de génie informatique

Regroupant les programmes de génie logiciel, de génie informatique et les programmes aux études supérieures, le comité des programmes du département de génie informatique réunit des professeurs et des étudiants intéressés par le développement du programme. Le responsable actuel du comité est le Pr. Hai Hoc Hoang.

### 1.1.2 Analyse du tronc commun actuel

#### Les spécificités inhérentes au génie logiciel

Même s'il y a de fortes similarités entre le génie logiciel et les autres disciplines de l'ingénierie, le génie logiciel comporte les spécificités suivantes :

- Spécificité 1.** les fondements sont ceux de l'informatique, et non des sciences dites naturelles ;
- Spécificité 2.** les mathématiques discrètes y occupent une place prépondérante ;
- Spécificité 3.** les entités abstraites ou logiques prévalent sur les artefacts concrets ou physiques.

En fonction de ces spécificités, nous pouvons effectuer une analyse du tronc commun actuel. Lorsque nous aurons besoin de présenter des arguments basés sur l'une de ces spécificités, nous utiliserons son numéro et rappellerons brièvement sa définition.

#### La pertinence des cours actuels au tronc commun

Dans le tronc commun actuel, un certain nombre de cours sont pertinents et devraient demeurer obligatoires dans le cadre de la formation en génie logiciel :

**ING1005 – Calcul I** Cours pertinent, pour régulariser les connaissances des nouveaux admis ; une exemption demeurerait toutefois possible.

**ING1006 – Algèbre linéaire** Cours pertinent, qui pourrait inclure les solveurs linéaires.

**ING1025 – Informatique** Cours essentiel, qui devrait demeurer le départ d'un apprentissage solide en programmation.

**ING1040 – Introduction au génie et aux projets d'ingénierie** Cours essentiel, qui sera adapté au contexte du génie informatique et du génie logiciel où les pratiques d'encadrement de projet sont différentes.

La plupart des autres cours sont à retirer ou à transformer substantiellement pour la formation en génie logiciel :

**ING1003 – Équations différentielles** Cours complémentaire. La spécificité 2 demanderait plutôt un (ou plusieurs) cours de mathématiques discrètes. Une solution serait d'envisager un cours plus léger de 2 crédits dans lequel les équations aux dérivées partielles ne seraient pas abordées.

**ING1007 – Calcul II** Cours à retirer du programme. La seule pertinence de ce cours est de permettre une certaine couverture de méthodes parfois utilisées en évaluation de performance ou pour intégrer certaines formules statistiques.

**ING1010 – Mécanique pour ingénieur** Cours à transformer. Les principes mécaniques y sont trop simplement calqués sur ceux vus au collégial ; un cours plus axé sur la consistance des représentations physiques serait un atout pour le génie logiciel, surtout dans la perspective de l'orientation Multimédia.

**ING1015 – Résistance des corps déformables** Cours à retirer.

**ING1020 – Communication graphique** Cours non pertinent à retirer ou à transformer substantiellement. La spécificité 3 demande une autre catégorie de représentation. Les plans/devis en génie logiciel utilisent des outils de modélisation standardisés, tel l'UML, pour représenter les entités abstraites (composantes, objets).

**ING1030 – Chimie pour ingénieur** Cours à retirer.

**ING1035 – Matériaux** Cours à retirer.

### 1.1.3 Besoin de révision des cours du curriculum

Nous ferons maintenant un survol des besoins du génie logiciel concernant les matières hors du tronc de spécialité :

#### Les apprentissages en mathématiques

Considérant que les mathématiques du continu sont en surnombre par rapport aux curriculums standards de génie logiciel, un certain nombre de cours devraient être créés ou remaniés pour parfaire la formation en génie logiciel :

**Mathématiques discrètes de base** La spécificité 2 nous demande de mettre l'accent sur les mathématiques discrètes plutôt que sur celles du continu. Dans ce sens, le programme de génie logiciel contient un cours (*LOG2900 – Éléments de structures discrètes*) regroupant les principales notions de cet apprentissage. Des discussions plus récentes au comité des programmes de génie informatique ont permis que les éléments de base de ce cours de structures discrètes soient également intégrées au projet de formation en génie informatique.

**Modélisation computationnelle** Une autre partie des apprentissages reliés aux mathématiques discrètes concerne la modélisation computationnelle. L'introduction de langages et de grammaires et le niveau de formalisme nécessaires pour comprendre les machines à états finis pourraient être regroupés dans un second cours.

**Modélisation et simulation numérique** Pour remplacer le cours de calcul scientifique pour ingénieur, un cours de modélisation et simulation numérique sera mis sur pied. Le calcul en point flottant et la propagation d'erreur y seront abordés, mais dans la perspective où les étudiants auront les connaissances d'architecture informatique nécessaires pour en mesurer les incidences dans les systèmes ordinés.

Nous sommes heureux que les discussions lors des rencontres avec les membres des autres programmes aient permis d'enlever les notions de représentation internes et de troncature qui étaient vues en mathématiques et qui devaient plutôt apparaître dans le cours d'introduction à l'informatique.

### Les apprentissages en sciences fondamentales

Certains cours de sciences fondamentales devraient être accessibles pour parfaire la formation en génie logiciel. La spécificité 1 nous demande une couverture assez large de l'informatique. Ceci limitera naturellement l'ensemble des cours disponibles pour d'autres sciences fondamentales. Afin d'assurer la motivation des étudiants et de leur permettre de choisir un domaine dans lequel ils pourraient éventuellement appliquer leurs connaissances en génie logiciel, nous proposons de leur laisser un certain choix parmi un ensemble élargi de cours en sciences fondamentales. Dans ce contexte d'ouverture, nous avons proposé de mettre à la disposition des étudiants certains cours au choix offerts sur le campus de l'Université de Montréal. Citons par exemple des cours d'introduction à la biochimie (nécessaires pour une spécialisation en bioinformatique). Toutefois, il est difficile de mettre en oeuvre de tels cours dans le cadre de l'horaire des études à Polytechnique.

Dans l'état actuel des discussions, nous proposons d'inclure les suggestions suivantes dans le programme :

**ING1010IL Physique des corps articulés** Les principes mécaniques et physiques y seraient présentés dans une optique de cohérence par rapport à la réalité. Ce cours axé sur la consistance des représentations physiques serait un atout pour le génie logiciel, surtout dans la perspective de l'orientation Multimédia.

**GCH201x Sciences du vivant** Ce cours a été suggéré par le département de génie chimique et il pourrait servir au curriculum du génie logiciel.

### Le bagage nécessaire en études complémentaires

Un certain nombre de cours pourraient être remaniés pour parfaire la formation en génie logiciel. La spécificité 3 transforme de façon importante la pratique courante de l'ingénieur logiciel. La gestion de projets impliquant des logiciels et l'opération/entretien de systèmes logiciels comporte des aspects spécifiques (tout comme les outils de conception). Une partie du cours *INF3300 – Gestion de projets informatiques* couvre certains de ces aspects. Nous proposons de transférer l'essentiel des apprentissages en gestion de projets vers les projets intégrateurs annuels. Certains éléments particuliers (gestion du risque, suivi de l'avancement d'un projet, arrimage entre risque,

planification et cycle de développement logiciel) pourraient se donner sous forme de modules à l'intérieur des cours-projets.

Les aspects économiques de l'ingénierie logicielle pourraient également être mis en valeur par un module particulier complétant le cours d'économique de l'ingénieur.

Le cours d'éthique appliquée à l'ingénierie est certainement une composante importante dans l'acquisition de connaissances en sciences humaines et en philosophie.

Le contrat entre un client et l'ingénieur logiciel (ou la participation d'un ingénieur logiciel à un projet d'envergure) comporte également certains aspects spécifiques. La communication devient un élément important lors de la phase d'analyse des besoins et de la spécification des exigences et, comme l'ingénieur logiciel doit recourir à des prototypes abstraits, des besoins particuliers sont issus de la pratique de l'ingénierie logicielle :

1. La notion de propriété logicielle se rapproche plus de celles des droits d'auteur, et il serait pertinent d'introduire des notions de droit pertinentes (licences, protection de la propriété intellectuelle) ;
2. La conception d'une application logicielle conviviale et bien adaptée à la clientèle visée, la documentation et l'aide à l'utilisation, ainsi que la formation des utilisateurs, sont des éléments importants de la pratique. Un logiciel mal adapté aux besoins et difficile à maîtriser ne sera pas utilisé à sa pleine capacité. Les compétences pour bien identifier, communiquer et spécifier ces éléments de la conception de logiciels doivent être supportées par l'enseignement de principes de conception de logiciels interactifs.
3. Récemment, de nombreuses lois ont été votées par les gouvernements afin de responsabiliser les entreprises quant à la sécurité informatique et à la confidentialité des données-clients.

#### 1.1.4 Les stages en génie logiciel

Le programme actuel offre une **filière coopérative** (avec stages obligatoires) ou une **filière régulière** (avec stages optionnels) au choix de l'étudiant. Les étudiants admissibles à la filière coopérative doivent toutefois avoir une moyenne suffisante. Cette polyvalence est l'une des caractéristiques du programme. Toutefois, il est difficile d'assurer le programme coopératif ; les étudiants du programme ont d'ailleurs manifesté leur inquiétude par rapport à cette difficulté :[8]

Les vingt-cinq étudiants coop ont appliqué en moyenne sur un peu plus de cinq stages qu'ils jugeaient assez pertinents à leur formation. Neuf d'entre eux ont appliqué sur plus de dix stages. Parmi ces 25 étudiants, *un seul* a obtenu une entrevue par l'entremise du Service de placement. Dans le sondage effectué auprès des étudiants en deuxième année de génie logiciel, la presque totalité des trente-trois participants ont répondu à la question *Suis-je satisfait des efforts faits par l'École pour l'application du mode coopératif en génie logiciel ?* par la négative !

L'équipe pédagogique est en accord avec un stage obligatoire de quatre mois pour tous les étudiants. Les paramètres plausibles de ce stage sont les suivants :

- Le stage serait effectué en troisième année du curriculum, afin de pouvoir intégrer certaines connaissances des pratiques industrielles lors de la quatrième année ;
- Une des options du projet intégrateur de la quatrième année, identifiable au projet de fin d'études du BCAP, serait de permettre certaines extensions au stage, soit sous la forme :
  - d'un cours clinique, où des projets industriels seraient réalisés à l'École ;
  - d'un projet d'une équipe logicielle effectué en industrie, mais avec un encadrement particulier.

### 1.1.5 Forces et faiblesses actuelles

Parmi les *points forts* du programme actuel de génie logiciel, mentionnons :

**L'arrimage avec le génie informatique** Les programmes actuels de génie logiciel et de génie informatique partagent une année et demie en commun.

**Les laboratoires et les projets** Les pratiques en laboratoire sont présentes dans la plupart des cours de spécialité du programme de génie logiciel ; le programme comporte déjà une bonne proportion de cours où les projets de développement de systèmes logiciels sont priorisés.

**Les ressources pédagogiques** Une salle interactive procure une facilité d'interconnexion pour les ordinateurs portatifs, un tableau interactif et une configuration en tables rondes. La plupart des cours ont des supports TIC modernes (didacticiels, sites Web développés).

Les *points faibles* du programme actuel sont :

**Les cheminements non respectés** Beaucoup d'étudiants tardent à faire certains cours de base du tronc commun actuel, les conservant pour la dernière année. La promotion par année réglerait ces problèmes.

**L'absentéisme** L'accessibilité du matériel de cours et les nombreuses évaluations (devoirs, laboratoires, projets) conduisent fréquemment à un absentéisme trop important.

**Les cours offerts sur une seule session** Les cours dit *noirs* dans le programme actuel rendent difficiles les réajustements de cheminement en cas d'échec.

### Balisage de l'actuel programme de génie logiciel

Un balisage effectué en décembre 2003 dans la perspective de fournir des données susceptibles d'aider l'accréditation du programme actuel est illustré à la figure 1.1. Les principales carences proviennent des lignes rouges. Ces données ont mis en lumière un certain nombre de défauts (qu'on pourrait aussi considérer comme des *points faibles*) au tronc de spécialité actuel :

**Assurance qualité** La plupart des programmes contiennent un cours d'assurance qualité obligatoire. Le nouveau programme devrait permettre de rétablir cette situation dès que possible.

**Sécurité informatique** L'importance accrue de cette composante se retrouve maintenant dans plusieurs programmes, quoiqu'elle ne fasse pas explicitement partie des exigences en génie logiciel.

**Conception et architecture** Une certaine faiblesse du programme actuel par rapport à la conception architecturale de logiciels peut également être observée. Plusieurs universités ont un cours spécifique d'architecture logicielle ; mais il est également possible d'intégrer ces notions dans le cadre d'une séquence de cours de conception bien arrimés à des projets.

Les cours actuels de spécialité arrivent trop tard dans la formation de la future ingénieure (ou du futur ingénieur) du logiciel, et il est difficile pour elle (ou lui) d'orchestrer l'architecture de grands systèmes logiciels. Le curriculum actuel ne permet pas d'amener rapidement les étudiantes et étudiants à un niveau de maturité suffisante en conception logicielle.

Version actuelle

Sujet	Idéal	Poly	ETS	Concordia	Carleton	Calgary	Ottawa	McMaster	Victoria
Statut An-Accr.	Obl/Opt	2004?	2003	2002	2003	2002	2001	2001	2005?
<b>CCPE-Obl. 1998 (...désuet!)</b>									
A1- Structures de données	Obl	INF1110	LOG320	CO352	2100	CP331	CSI2114	2C03	CS225
A2- Architecture systèmes numériques	Obl?	INF2500	???	SOEN228?	3601	EN369	EG2131/31:	3MG3	CS230
A3- Systèmes d'exploitation	Obl	INF3600	LOG710	CO346	4001	CP457	CSI3310	3SH3	CS360
A4- A+C algorithmes	opt	INF4705	LOG620	CO465				4TD3	CS326/425
A5- Conception de logiciels	Obl	LOG2000	LOG120	SOEN343	2100	311	2500/2501	2AA4	265
A6- Fichiers et BD	Obl	INF4700	L630/660	CO353	3001/5703	CP471	SI2131/331	3H03	CS370
A7- A+C+R orienté objet	Obl	LOG3700	LOG220	???	2004/2101		3110/3310	3BB4	430
<b>CIPS- grid- SE accreditation</b>									
Spécification et requis	Obl	LOG3300	LOG410	SOEN342		523(FM)		4EF3	321
Analyse (A)	?	LOG2000			3100	EN007	2100 +	2AA4+	
Conception ( C )	?	LOG3700	LOG120	SOEN343	2002/3100	team AC	2500	3A04+	265
<b>Architecture</b>	Obl	{?}	LOG420	SOEN344		443		3B04	221
C systèmes embarqués	opt	INF4602		COEN421					440
C systèmes temps réel	opt	{INF4600}	LOG550	COEN320	3303	EN491	CEG4153	4A03	CE455
C systèmes répartis	opt	INF4402	LOG730	SOEN423	4504	CP441	4183	4F03	462
C+Evaluation ( E ) interfaces usagers	Obl	LOG2700	LOG340	SOEN357		513(Web)	3120	4D03	310
Réalisation ( R ) - documentation		LOG4315					4000		499
Réalisation ( R ) - outils	projet	projet	projet	projet			projet	projet	454(comp)
Réalisation ( R ) - composantes	année ?	session					année		
Processus (P) - maintenance	?	LOG3301	LOG230	SOEN341		531	3100		371(evol)
<b>Processus (P) - assurance qualité</b>	Obl	LOG3301	LOG330	SOEN384	5105	511	4111		426
Processus (P) - métriques	?	LOG3301	LOG510	SOEN337		421			
Tests (V&V)	?	LOG4500		SOEN337	4101	521		3R03	
<b>Sécurité</b>	?	INF4420		SOEN321			4138	4C03	360
Performance des systèmes	opt	INF4601P			5101			4E03	424(fia)
<b>Non requis, mais Obl actuellement</b>									
Réseaux	?	INF3402	LOG610	CO445		EN573	ELG3300		450/460
Langages et compilateurs	opt	INF4303		CO442		CP411	CSI4115		CS435

FIG. 1.1 – Comparaison des cours du programme actuel avec certaines universités canadiennes

## 1.2 Valeurs poursuivies

Les principales valeurs à poursuivre sont :

1. Le sens des responsabilités et le respect de normes adéquates à la pratique du génie logiciel, tout en intégrant les valeurs propres à la profession d'ingénieur, en particulier les droits et devoirs liés à la profession.
2. Une responsabilisation par rapport aux apprentissages, en vue d'acquérir la maturité requise pour poursuivre une formation continue même après sa diplomation.
3. Un sens des valeurs sociales, légales et éthiques nécessaires lors de l'élaboration de solutions logicielles de grande envergure.

## 1.3 Définition de l'ingénieur

La première conférence jamais tenue sur le sujet du génie logiciel a eu lieu à Garmish (Allemagne) en octobre 1968.[4] Plusieurs aspects fondamentaux qui ont mené à la création de la discipline du génie logiciel y ont été alors abordés, notamment :

- les difficultés de rencontrer les échéanciers et les exigences lors de projets logiciels de grande envergure ;
- l'épineuse question de savoir si le logiciel devrait être acheté indépendamment du matériel ;
- les principes de la formation des ingénieurs logiciels.

Avant de décrire le rôle des ingénieures et ingénieurs du logiciel, d'abord une brève définition du génie logiciel :[5]

*The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software.*

### 1.3.1 Le rôle de l'ingénieur logiciel

Le génie logiciel applique les pratiques professionnelles de l'ingénierie à la conception de systèmes logiciels, intégrés et à grande échelle. L'ingénieur logiciel maîtrise la théorie et les méthodes de l'informatique. Plus spécifiquement, il élabore et gère des projets concernant la création, la maintenance et l'évolution des systèmes logiciels. En mars 2001, l'Ordre des ingénieurs du Québec a reconnu officiellement cette spécialisation qu'est le génie logiciel. Depuis 2001 également, les universités d'Ottawa, de McMaster et de Western Ontario offrent un programme de formation accrédité par le BCAPSI en génie logiciel. L'École de Technologie Supérieure forme également des ingénieurs du logiciel accrédités.

### 1.3.2 Tâches de l'ingénieur logiciel

En plus des tâches administratives inhérentes au suivi de tout projet d'ingénierie dans les domaines qui incombent à son champ d'exercice, l'ingénieur logiciel doit être en mesure d'effectuer les tâches professionnelles spécifiques suivantes :

- Analyser et pouvoir spécifier les besoins et les exigences du développement d'un produit logiciel ou d'un système logiciel ;
- Concevoir des logiciels tout en assurant d'abord l'intérêt du public, puis celui de ses clients, à partir de spécifications conformes aux besoins ;
- Développer des produits logiciels ou des systèmes basés sur des logiciels en suivant les normes professionnelles adéquates ;
- S'assurer par des tests adéquats que les produits logiciels sont conformes aux spécifications énoncées ;
- Si nécessaire, voir à l'entretien et à la certification logicielle.

L'encadrement de ces activités doit se faire dans un contexte de planification des coûts et des ressources.

### 1.3.3 Particularités du génie logiciel à l'École

Le nouveau programme met l'accent sur les savoirs et habiletés nécessaires qui permettront à la future ingénieure ou au futur ingénieur d'agir efficacement dans le cadre d'équipe d'ingénieurs multidisciplinaires. Cette approche *conceptuelle* employée lors de la présentation des connaissances est en opposition avec une approche plus *technologique*, pratiquée dans plusieurs établissements. En plus d'acquérir les bases d'une pratique saine et consistante permettant la production et la maintenance de produits logiciels, les connaissances conceptuelles de l'ingénieure ou l'ingénieur logiciel de Polytechnique lui permettront d'accroître le *niveau de maturité* du processus logiciel de l'entreprise dans laquelle elle ou il travaillera.

Après quelques années de pratique professionnelle, elle ou il sera en mesure d'exercer un contrôle quantitatif du processus de développement logiciel et saura gérer efficacement les activités d'ingénierie logicielle afin de réaliser des projets d'envergure.

## 1.4 Compétences prônées par le nouveau programme

### 1.4.1 Compétences spécifiques au génie logiciel

En relation avec les tâches décrites au paragraphe précédent, les principales compétences à développer sont :

1. Résoudre des problèmes complexes appartenant à des domaines d'applications divers et spécialisés.
2. Appliquer des modèles et des techniques standardisés qui permettent l'analyse et l'identification des problèmes qui ont trait à la conception, au développement, à la vérification, à la validation et à la mise en service de produits logiciels.

3. Planifier et gérer des projets en ingénierie du logiciel.
4. Travailler efficacement individuellement et en équipe en vue de développer des artefacts logiciels de qualité ; ceci implique une facilité de communication et une capacité à respecter des échéanciers.
5. Exercer des capacités d'analyse, d'abstraction, de synthèse et de créativité.
6. Se familiariser rapidement avec le domaine d'application et réaliser une évaluation critique nécessaire pour produire une solution conforme aux besoins du client ; ceci met en valeur le leadership, la pratique de négociations, l'esprit de synthèse en vue de clarifier les exigences.
7. Trouver des compromis acceptables permettant de concevoir un logiciel réalisable, en tenant compte des ressources disponibles, qu'elles soient budgétaires, temporelles ou humaines.
8. Utiliser les habiletés de base nécessaires pour démarrer une pratique professionnelle en génie logiciel ; ceci met en lumière l'importance que l'aspirant-ingénieur puisse acquérir ces habiletés dans un cadre réaliste par des projets d'envergure et des stages industriels.
9. Exercer la profession d'ingénieur avec professionnalisme.

#### 1.4.2 Compétences transversales

Les apprentissages transversaux sont nécessaires afin de compléter la formation de l'ingénieur logiciel. Certaines connaissances pratiques seront acquises lors des projets intégrateurs annuels, mais celles-ci seront difficilement mesurables dans l'environnement homogène des cours de spécialité d'un programme. Une suggestion qui permettrait de rallier un certain nombre d'objectifs dans ce sens, en plus de développer les habiletés personnelles et relationnelles entre futurs ingénieurs, serait d'établir à l'École un *projet multidisciplinaire*, qui permettrait l'intégration de diverses disciplines du génie dans un projet conjoint.

### 1.5 État du consensus des professeurs

Pour parler de l'état du consensus des professeurs, il semble nécessaire de préciser le processus de mise en oeuvre du projet de formation (PDF) au département de génie informatique (DGI). Au DGI, c'est le *Comité des programmes* qui est responsable de la mise en oeuvre du PDF pour les deux programmes du 1<sup>er</sup> cycle en génie informatique et en génie logiciel. Ce comité comprend huit à dix professeurs et trois à quatre étudiants. Le travail de fond pour le PDF de chacun des deux programmes est assuré par un sous-comité formé de trois à quatre professeurs, qui produit les documents de travail discutés au Comité des programmes. Les propositions du Comité des programmes sont discutées au *conseil de département* et au *COCEP départemental* avant d'être soumises à l'*Assemblée des professeurs* pour approbation départementale. En préparant les documents de travail, les membres des deux sous-comités sondent souvent les opinions de leurs collègues et voisins de façon informelle à travers des discussions de corridors.

Les travaux du PDF au DGI au cours du trimestre d'hiver 2004 se sont ainsi déroulés et ont mené à deux documents sur la vision d'ensemble et les objectifs de chacun des deux programmes

sous la responsabilité du DGI. Avant d'être déposés au *Comité d'implantation* le 4 juin 2004, ces deux documents ont été distribués à tous les professeurs du DGI pour fin de consultation à la fin du mois de mai 2004. Il est à noter qu'à ce moment, le DGI était en attente de la nomination de son nouveau directeur et de sa nouvelle structure organisationnelle. Toutefois, l'état de consensus était reflété par la proportion des professeurs qui étaient membres du Comité des programmes (dix sur vingt-sept) et les discussions informelles entre les professeurs.

Depuis le mois de septembre 2004, le nouveau Comité des programmes poursuit la mise en oeuvre du PDF selon le même processus adopté l'année dernière en se réunissant deux fois par mois. Il a présenté, lors d'une assemblée spéciale des professeurs tenue le 20 octobre 2004, les résultats intermédiaires de ses travaux sous la forme des tableaux du cheminement des cours pour les deux programmes de génie informatique et de génie logiciel, et recueilli un grand nombre de suggestions provenant des professeurs durant un *tour de table* très constructif. L'Assemblée a accepté à l'unanimité les tableaux des cours proposés par le Comité des programmes, nonobstant des ajustements futurs. En outre, les mêmes propositions ont été discutées à la réunion du COCEP départemental le 25 octobre ; les membres du COCEP ont exprimé leur appui au moyen de plusieurs commentaires constructifs. Ce Livrable A complet a été discuté au Conseil de département du 17 novembre, et sera maintenant envoyé aux membres du COCEP et à tous les professeurs pour fin de consultation. Le but est toujours d'obtenir le consensus général de tous les professeurs du DGI.

## Chapitre 2

### Tableau des cours du programme

Le tableau de cheminement et les différents graphiques l'illustrant dans ce chapitre a été construit en utilisant le code de couleur mentionné à la figure 2.1. Ce code de couleur provient du document préliminaire décrivant le curriculum du génie logiciel.[2]

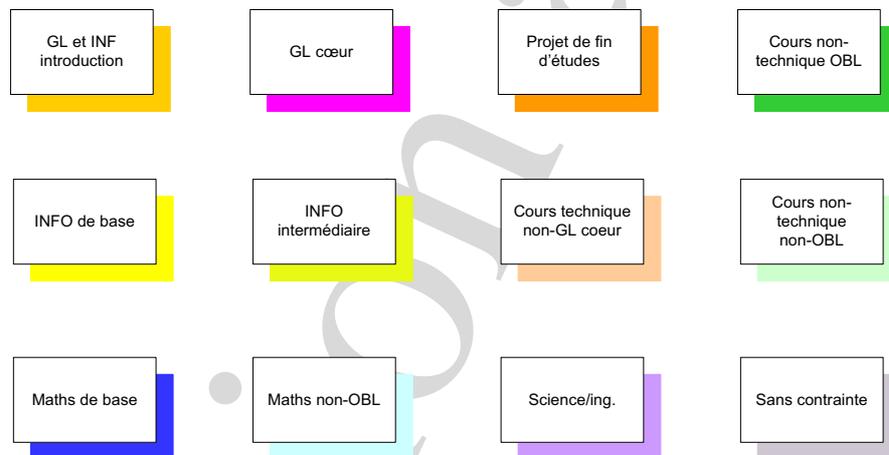


FIG. 2.1 – Code de couleur utilisé pour le programme de génie logiciel

## 2.1 Tableau du cheminement

À la figure 2.2, le tableau global des cours du nouveau programme est présenté. Le comité des programmes a tenté de maintenir le plus de cours en commun entre les nouveaux programmes de génie logiciel et de génie informatique. Différentes consultations (portes ouvertes, discussions avec des parents) ont fait émerger ce constat qu'il faut éviter une surspécialisation qui empêche l'étudiant de transformer son choix d'entrée. Le nouveau programme reconduit le programme actuel où il est possible de changer de programme sans trop de pénalité. Toutes ces raisons ont conduit à construire une première année de cours qui soit entièrement commune entre les deux programmes. Il est également à noter que, pour le moment du moins, seul un stage de quatre mois obligatoire est envisagé dans le nouveau cheminement. La possibilité actuelle de choisir entre un programme coopératif ou non est abolie. Les possibilités d'avoir un programme purement coopératif avec au moins 12 mois de stages obligatoire seront analysées ultérieurement.

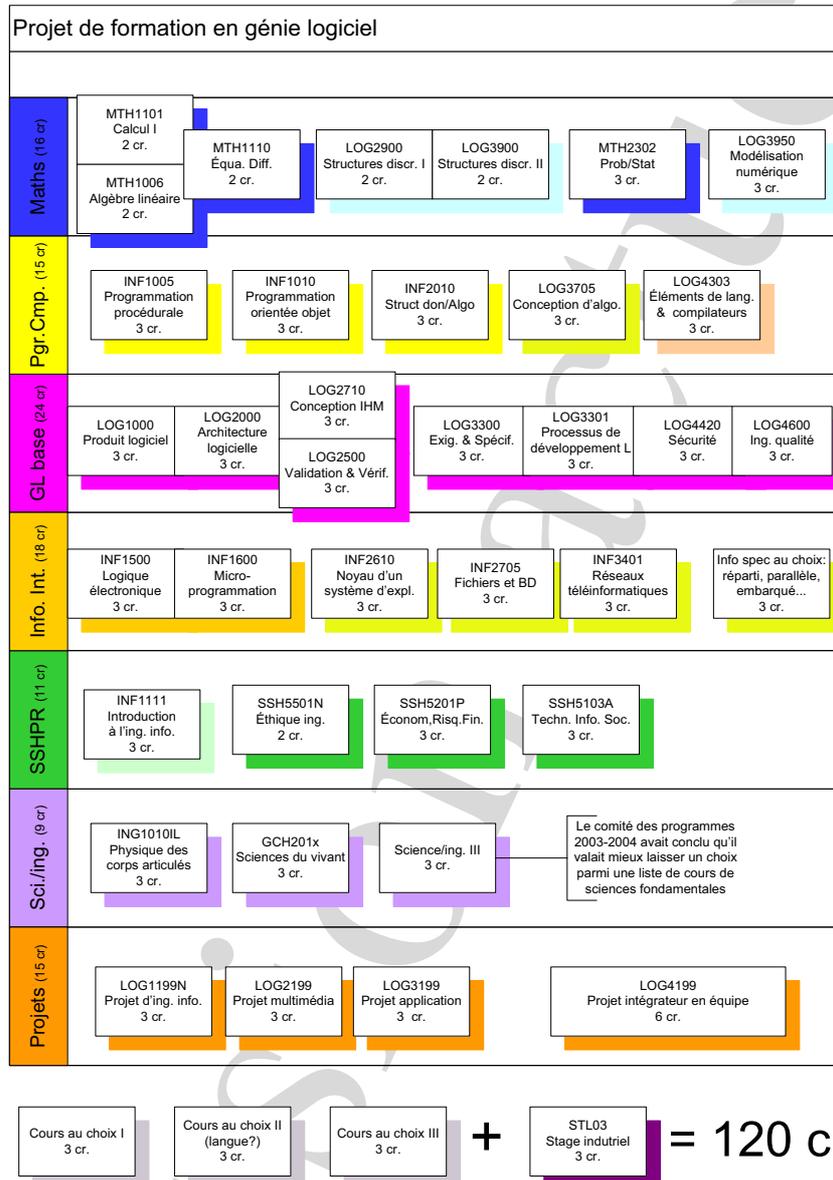


FIG. 2.2 – Tableau global du nouveau programme de génie logiciel

### 2.1.1 La première année

La figure 2.3 décrit cette première année commune du programme. Des informations plus précises concernant les préalables et les corequis sont données dans le tableau 2.1.

TAB. 2.1 – Cours de 1<sup>re</sup> année en génie logiciel

Sigle temporaire	# cr.	Titre temporaire	Cours préalable	Cours corequis
MTH1101	2	Calcul I	—	—
MTH1006	2	Algèbre linéaire	—	—
INF1111	3	Introduction à l'ingénierie informatique	—	—
INF1500	3	Logique élect.	—	—
INF1005	3	Programmation proc.	—	—
MTH1110	2	Équations différentielles	—	MTH1101
LOG1199	3	Projet d'ingénierie informatique	INF1111	LOG1000
INF1600	3	Micro-programmation	INF1500	—
INF1010	3	Programmation OO	INF1005	—
LOG1000	3	Produit logiciel	INF1005	—
Total	27 crédits pour la 1 <sup>re</sup> année			

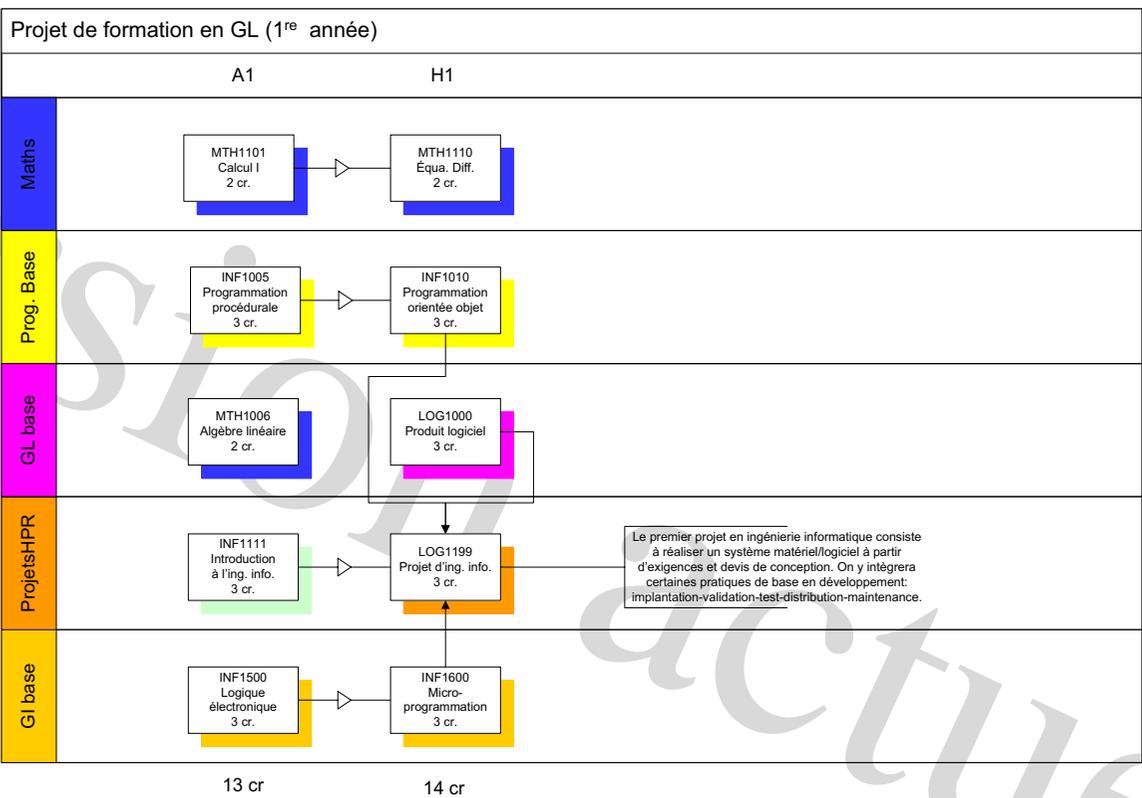


FIG. 2.3 – Première année du nouveau programme de génie logiciel

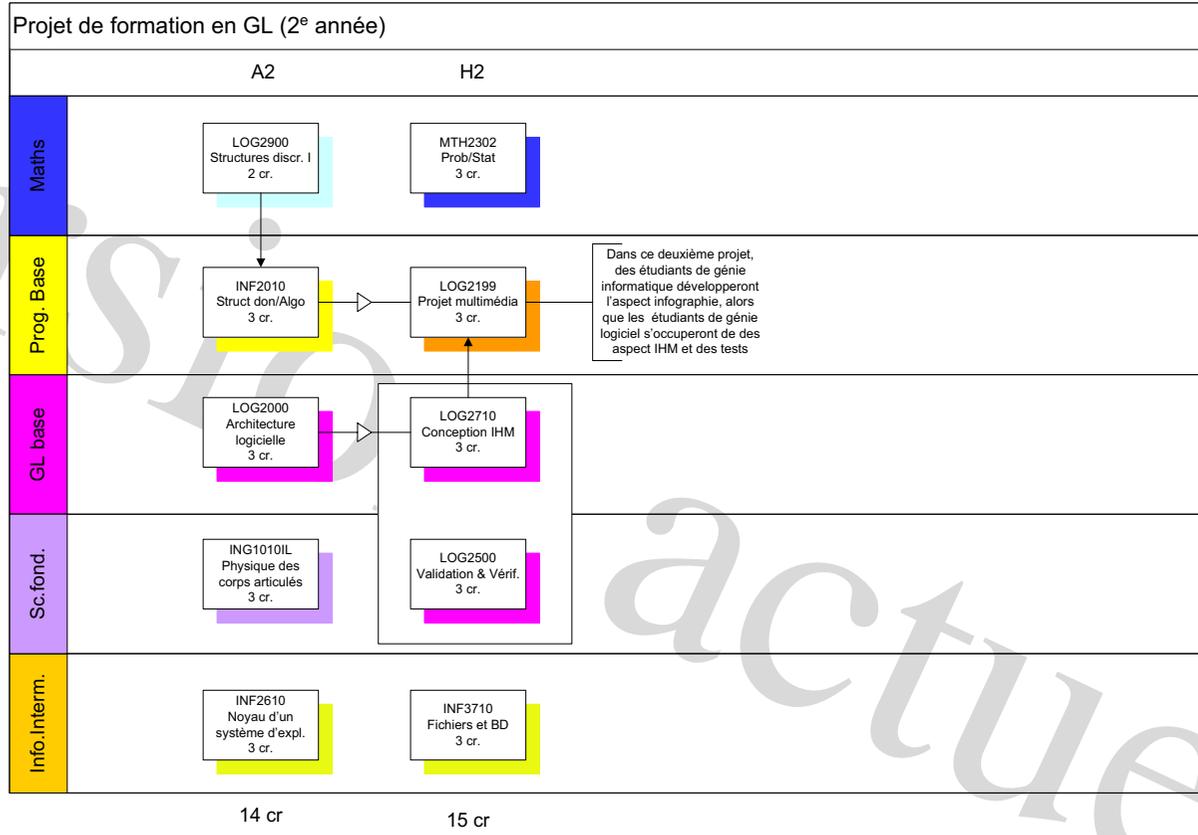
### 2.1.2 La deuxième année

Le programme de génie logiciel se distingue à partir de la deuxième année. Les cours *LOG2710 – Conception d’interface personne-machine* et *LOG2500 – Validation et tests* sont choisis plutôt que les cours de la chaîne matérielle du génie informatique et le cours d’infographie. Toutefois, les étudiants de génie logiciel et de génie informatique sont à nouveau rassemblés à la fin de la deuxième année afin de participer à un projet conjoint dont le thème sera le développement d’une application multimédia. Il est important de mentionner que les étudiants seront alors mis en équipes bidisciplinaires ; les fonctionnalités concernant les interfaces personne-machine et les tests seront du ressort des étudiants du génie logiciel, alors que les aspects infographiques, les performances du système seront du ressort des étudiants de génie informatique. La figure 2.4 décrit cette deuxième année du programme. La session d’automne de la deuxième année n’est différente de celle du génie informatique que par un seul cours. Des informations plus précises concernant les préalables et les corequis sont données dans le tableau 2.2.

TAB. 2.2 – Cours de 2<sup>e</sup> année en génie logiciel

Sigle temporaire	# cr.	Titre temporaire	Cours préalable	Cours corequis
INF2600	3	Noyau d’un système d’expl.	INF1600	—
LOG2000	3	Architecture logicielle	LOG1000	—
INF2010	3	Struc. données et algorithmes	INF1010	LOG2900
LOG2900	2	Struc. discrètes I	INF1010	INF2010
ING1010IL	3	Sciences I – Physique des corps articulés	—	—
MTH2302	3	Probabilités et statistiques	—	MTH1101
INF2705	3	Fichiers et base de données	INF2010	—
LOG2199	3	Projet multimédia	INF2010	LOG2710
LOG2500	3	Validation et vérification	LOG2000	LOG2710
LOG2710	3	Conception d’interface personne-machine	LOG2000	LOG2500
Total	29 crédits pour la 2 <sup>e</sup> année			

FIG. 2.4 – Deuxième année du nouveau programme de génie logiciel



### 2.1.3 La troisième année

La figure 2.5 décrit la troisième année du programme. Des informations plus précises concernant les préalables et les corequis sont données dans le tableau 2.3. Il est à remarquer que le comité des programmes du département a jugé bon d'isoler les crédits associés au stage industriel et au cours d'éthique, puisque ces cours seront pris par les étudiants au moment qu'ils jugeront opportuns.

TAB. 2.3 – Cours de 3<sup>e</sup> année en génie logiciel

Sigle temporaire	# cr.	Titre temporaire	Cours préalable	Cours corequis
GCH201x	3	Sciences II – Sciences du vivant	—	—
INF3705	3	Conception d'algorithmes	LOG2900	—
LOG3900	2	Struc. discrètes II	LOG2900	—
LOG3300	3	Exigences et spécifications	LOG2500	—
	3	Cours au choix I	—	—
INF3401	3	Réseaux téléinformatiques	—	—
LOG3199	3	Projet d'application	LOG3300	LOG3301
LOG3301	3	Processus de développement logiciel	LOG3300	LOG3199
LOG3950	3	Modélisation et simulation numériques	INF2600	—
SSH5201P	3	Économie, risque et finances	—	—
Total	29 crédits pour la 3 <sup>e</sup> année			
STL03	3	Stage industriel	(60 cr.)	—
SSH5501N	2	Éthique	(60 cr.)	—
Total	5 crédits mobiles (contraintes étudiants)			



### 2.1.4 La quatrième année

La figure 2.6 décrit la quatrième année du programme. Des informations plus précises concernant les préalables et les corequis sont données dans le tableau 2.4. Il est à noter que plusieurs choix sont offerts pour la réalisation du projet intégrateur de quatrième année :

**Le projet annuel.** Le cheminement est alors :

A4 – LOG4199A, soit la première partie du projet ;

H4 – LOG4199H, la seconde partie du projet et un cours au choix.

**Le projet trimestriel.** Le cheminement devient plutôt :

A4 – le choix III en sciences ou le cours III au choix est réalisé ;

H4 – LOG4199, soit un projet de 6 crédits.

**Le cours clinique.** Le cheminement est alors :

A4 – le choix III en sciences est réalisé ;

H4 – le cours clinique de 9 crédits supprime le troisième cours au choix.

Il est encore trop tôt pour valider les combinaisons à privilégier, celles-ci dépendront du marché du travail dans quelques années.

TAB. 2.4 – Cours de 4<sup>e</sup> année en génie logiciel

Sigle temporaire	# cr.	Titre temporaire	Cours préalable	Cours corequis
LOG4199A	3	Projet industriel (partie 1)	STL03	—
LOG4303	3	Éléments de langages et compilateurs	INF3705	—
LOG4600	3	Ingénierie de la qualité	LOG3301	—
SSH5103A	3	Technologie de l'information et société	—	—
	3	Cours au choix II	—	—
LOG4199H	3	Projet industriel (partie 2)	LOG4199A	—
LOG4420	3	Sécurité	(85 cr.)	—
	3	Informatique au choix	—	—
	3	Sciences III –	—	—
	3	Cours au choix III	—	—
Total	30 crédits pour la 4 <sup>e</sup> année			

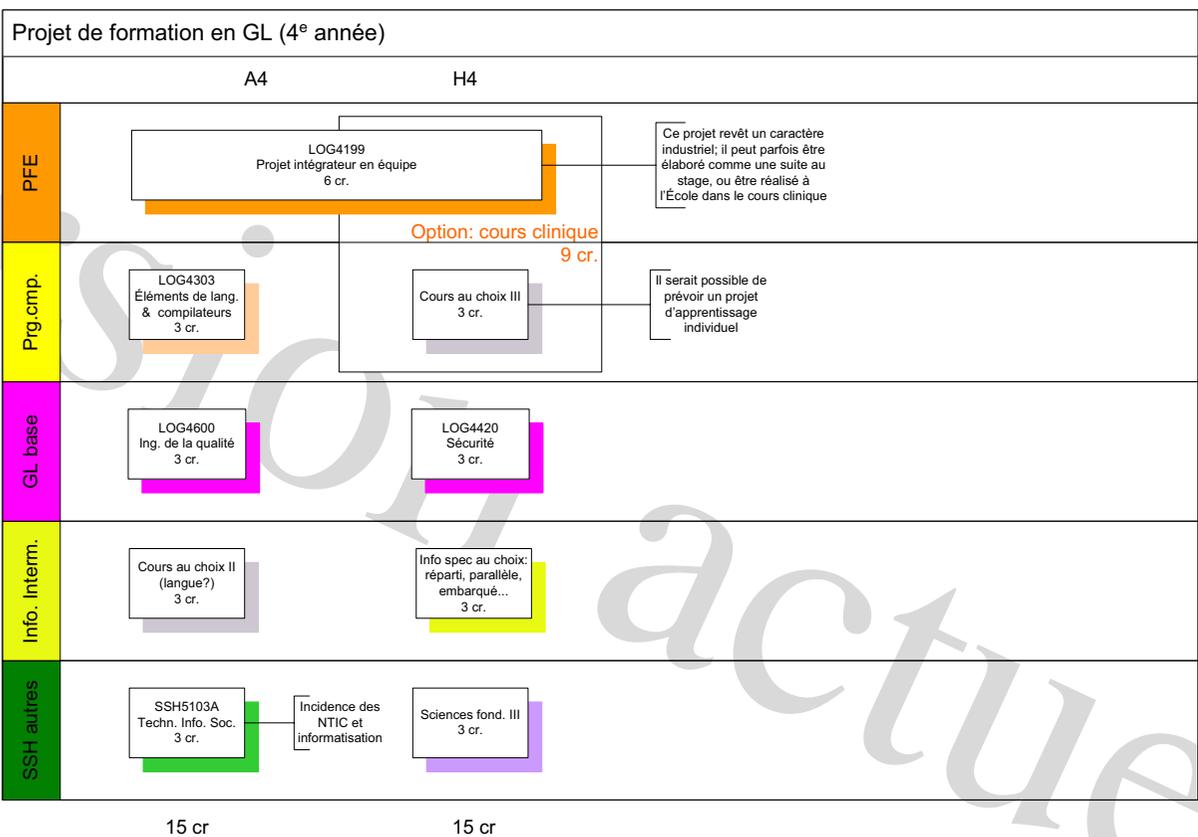


FIG. 2.6 – Quatrième année du nouveau programme de génie logiciel

### 2.1.5 Les orientations et les concentrations

Dans l'actuel curriculum, l'orientation *Multimédia*, offerte par le département de génie informatique, est disponible à la fois pour le génie logiciel et pour le génie informatique. Il sera possible de maintenir cette orientation dans le nouveau programme. Pour ce faire, le cours d'informatique spécialisée prévu en quatrième année deviendrait un cours d'infographie, et devrait être pris plus tôt. Les cours libellés *cours au choix I, II et III* seraient évidemment éliminés au profit des cours de l'orientation.

L'orientation thématique *Innovation technologique* s'avère d'un intérêt limité pour une future ingénieure ou un futur ingénieur du logiciel. En effet, la propriété intellectuelle du produit logiciel est couverte par la loi des *droit d'auteur*, ce qui change significativement les pratiques usuelles de mise en marché et de commercialisation. Toutefois, les apprentissages de cette orientation peuvent s'avérer d'intérêt pour les gestionnaires des technologies de l'information.

Si la clientèle étudiante le permet, d'autres orientations seront également développées. Des discussions ont présentement lieu pour développer des concentrations en *Multimédia* et en *Biomédical*.

## 2.2 Principes généraux

### 2.2.1 Le balisage du nouveau programme

Un travail de balisage a été réalisé afin d'orienter l'élaboration du programme. Ce balisage a été effectué en comparant les programmes de génie logiciel et de génie informatique de l'École, nouveaux et anciens, avec les programmes équivalents offerts par les universités Laval, McGill et Waterloo. Un survol moins formel des programmes offerts par certaines autres institutions a également été effectué. Ces institutions sont :

- l'École de technologie supérieure (ÉTS),
- le Rochester Institute of Technology (RIT),
- les universités canadiennes de Carleton, Concordia, McMaster, Waterloo, de Calgary, d'Ottawa et l'UWO (Western).

Le balisage à l'international est compliqué par le fait que, à l'extérieur du Canada, de l'Australie, des Indes et des États-Unis, le nombre de programmes de génie logiciel en activité est extrêmement bas. De plus, il y a plus de programmes en Australie qu'il y en a aux États-Unis. Les résultats combinés pour les programmes de génie logiciel et de génie informatique sont donnés à la figure 2.7.



Les résultats spécifiques du balisage du programme de génie logiciel sont présentés à la figure 2.8. La classification a été inspirée par celle utilisée dans le document "Computing Curriculum - Software Engineering" (Rapport final - 21 mai 2004, ACM, IEEE Computer Society).[2] Il ressort clairement que notre proposition de programme est tout à fait en ligne avec ce qui est enseigné dans ces établissements. Les principales caractéristiques de notre programme, vues sous l'angle de ces données, se décrivent comme suit :

1. Le programme proposé se démarque principalement par l'importance accordée aux stages et cours-projets. Ce constat découle directement des orientations prises par l'École en la matière. Aucun des autres programmes ayant fait l'objet d'un survol, à l'exception de celui offert par l'ÉTS, ne consacre autant de crédits universitaires à ce type d'activité.
2. Le nombre de crédits alloués aux mathématiques "continues" est de 9, en baisse de 3 par rapport à l'ancien programme. Par comparaison, les trois autres programmes étudiés prévoient entre 11 et 15 crédits pour ce type d'activités.
3. L'ancien programme comprenait un nombre de crédits de génie informatique plus élevé que les trois programmes étudiés. Le nouveau programme ramène ce nombre tout juste en deçà de l'offre de Laval et McGill, mais au delà de ce qui est offert à Waterloo.
4. Le nouveau programme ne comporte pas de crédits pour des cours de sciences humaines autres que ceux ayant un lien direct avec la profession (éthique, économique, sociologie). Par comparaison, les trois autres programmes étudiés prévoient entre 6 et 15 crédits pour ces activités.
5. La formation spécifique en génie logiciel demeure le point focal du programme. Pour chacune des catégories propres au champ du génie logiciel (analyse, conception, validation et vérification, évolution, processus, qualité, gestion), le programme se situe dans le peloton de tête pour le nombre de crédits consacrés, à l'exception de la catégorie "modélisation et analyse" où le programme de Waterloo fait bande à part. En outre, notre programme se démarque par la présence de cours complets axés vers les processus et la qualité du logiciel.

En résumé, le programme proposé est focalisé vers la formation accrue en génie logiciel et vers les cours-projets et stages. Ce focus se fait au détriment d'une légère diminution, dans la perspective des autres programmes, du nombre de crédits alloués aux autres catégories. Cette diminution est répartie de manière à ce qu'aucun champ spécifique ne souffre véritablement de la situation.

GL										
Poly (Ancien)	Poly (Nouveau)	Laval	McGill	Waterloo	Poly (Ancien)	Poly (Nouveau)	Laval	McGill	Waterloo	
12,0	9,0	11,0	15,0	15,0	10	8	9	14	12	MTH
2,0	4,0	5,0	3,0	6,0	2	3	4	3	5	DSC
3,0	3,0	3,0	3,0	3,0	3	3	3	3	2	PRS
12,0	9,0	9,0	0,0	16,5	10	8	8	0	13	SCI
2,0	0,0	0,0	3,0	0,0	2	0	0	3	0	OEN
4,0	14,0	6,0	6,0	2,3	3	12	5	6	2	ST
2,0	0,0	11,0	6,0	15,0	2	0	9	6	12	SSH
3,3	3,0	3,0	3,0	3,0	3	3	3	3	2	FND
6,5	7,0	3,0	8,0	0,5	5	6	3	7	0	PP
3,3	4,0	3,0	8,5	3,0	3	3	3	8	2	CMP
10,6	10,8	12,2	11,9	12,5	9	9	10	11	10	PRF
1,4	2,0	3,0	0,0	3,0	1	2	3	0	2	HCI
12,0	12,0	15,0	12,0	15,0	10	10	13	11	12	SAS
24,0	20,8	20,8	24,3	15,0	20	17	17	22	12	GI
4,8	4,0	3,5	0,6	7,3	4	3	3	1	6	MAA
5,1	4,0	3,0	0,6	4,3	4	3	3	1	3	DES
3,5	4,0	3,0	3,6	2,0	3	3	3	3	2	VAV
0,0	0,0	0,5	0,6	0,0	0	0	0	1	0	EVO
3,3	3,0	1,0	0,0	0,3	3	3	1	0	0	PRO
1,0	2,5	1,0	0,0	1,8	1	2	1	0	1	QUA
4,5	4,0	3,0	0,0	3,0	4	3	3	0	2	MGT
120	120	120	109	128	100	100	100	100	100	Total
cr.	cr.	cr.	cr.	cr.	%	%	%	%	%	

FIG. 2.8 – Balisage du programme de génie logiciel

## 2.2.2 Les cours de spécialité en première année

Un cours d'introduction au génie logiciel LOG1000 sera dorénavant déployé lors de la première année dans les programmes de génie logiciel et de génie informatique. Dans le contexte d'une année commune partagée avec les étudiants de génie informatique, ce cours remplit un double objectif :

- servir d'introduction aux pratiques en ingénierie logicielle pour les étudiants de génie logiciel et de génie informatique ;
- permettre la présentation d'un certain nombre de notions nécessaires au projet intégrateur de première année, et qui devront être mises en pratique par les étudiantes ou les étudiants de génie logiciel.

Parmi la liste des cours de première année donnée à la figure 2.3, la moitié ont des objectifs de formation à caractère informatique :

<p><b>LOG1000 – Produit logiciel</b></p> <p><b>INF1005 – Programmation procédurale</b></p> <p><b>INF1010 – Programmation orientée objet</b></p> <p><b>INF1500 – Logique électronique</b></p> <p><b>INF1600 – Micro-programmation</b></p>
--

En ajoutant le cours d'introduction INF1111 et le projet intégrateur de la première année LOG1199, on obtient un total de sept (7) cours d'intérêt immédiat pouvant motiver l'étudiante ou l'étudiant.

## 2.2.3 Les mathématiques dans le nouveau programme

La figure 2.9 décrit les cours de mathématiques de base confiés à l'unité **mathématiques**.

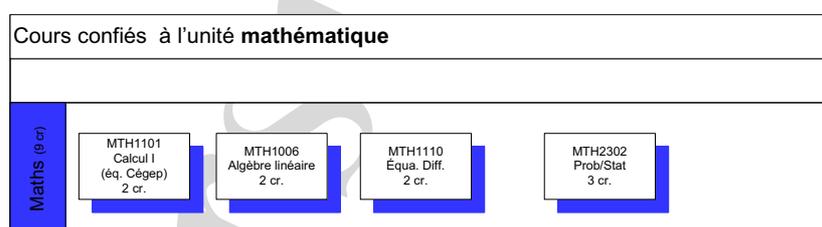


FIG. 2.9 – Mathématiques dans le nouveau programme de génie logiciel

## 2.2.4 Les études complémentaires dans le nouveau programme

La figure 2.10 décrit les cours d'études complémentaires confiés à l'unité **sciences humaines**.

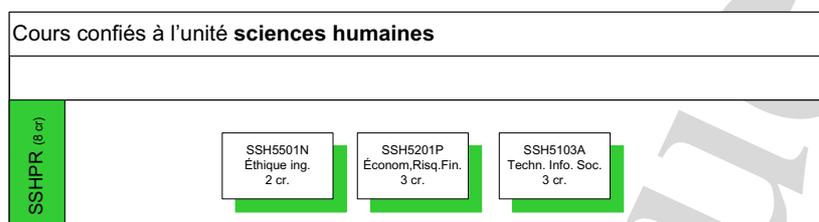


FIG. 2.10 – Études complémentaires dans le nouveau programme de génie logiciel

## 2.2.5 Les sciences fondamentales dans le nouveau programme

La figure 2.11 décrit les cours de sciences fondamentales confiés à d'autres départements. Une couverture partielle des sciences fondamentales est suggérée par les deux premiers choix, soit ING1010IL – *Physique des corps articulés* et GCH201x *Sciences du vivant*. Le troisième choix n'est pas déterminé, mais il serait quand même contraint à améliorer la couverture de connaissances en sciences fondamentales.

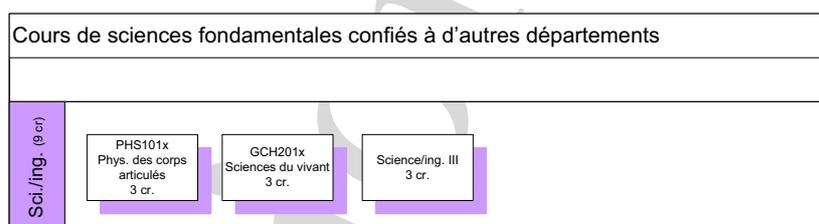


FIG. 2.11 – Sciences fondamentales dans le nouveau programme de génie logiciel

### **2.2.6 Les cours mis en place conjointement avec le programme de génie informatique**

La figure 2.12 décrit les cours mis en place conjointement avec le programme de génie informatique.

### **2.2.7 Les cours spécifiques au génie logiciel**

Les cours spécifiques au génie logiciel sont présentés à la figure 2.13

### **2.2.8 Les cours offerts aux autres départements**

Le département de génie informatique a réagi rapidement aux demandes des autres départements concernant certains ajustements aux cours de programmation. L'offre du cours de base en programmation ING1025 a été révisée pour se concentrer sur une approche *procédurale* avec un choix de langages d'introduction à la programmation. Les membres de l'équipe pédagogique de génie logiciel ont défini une séquence de trois nouveaux cours de programmation. Après discussion avec le département de génie électrique, les deux premiers cours (INF1005 et INF1010) feront partie du curriculum du programme de génie électrique.

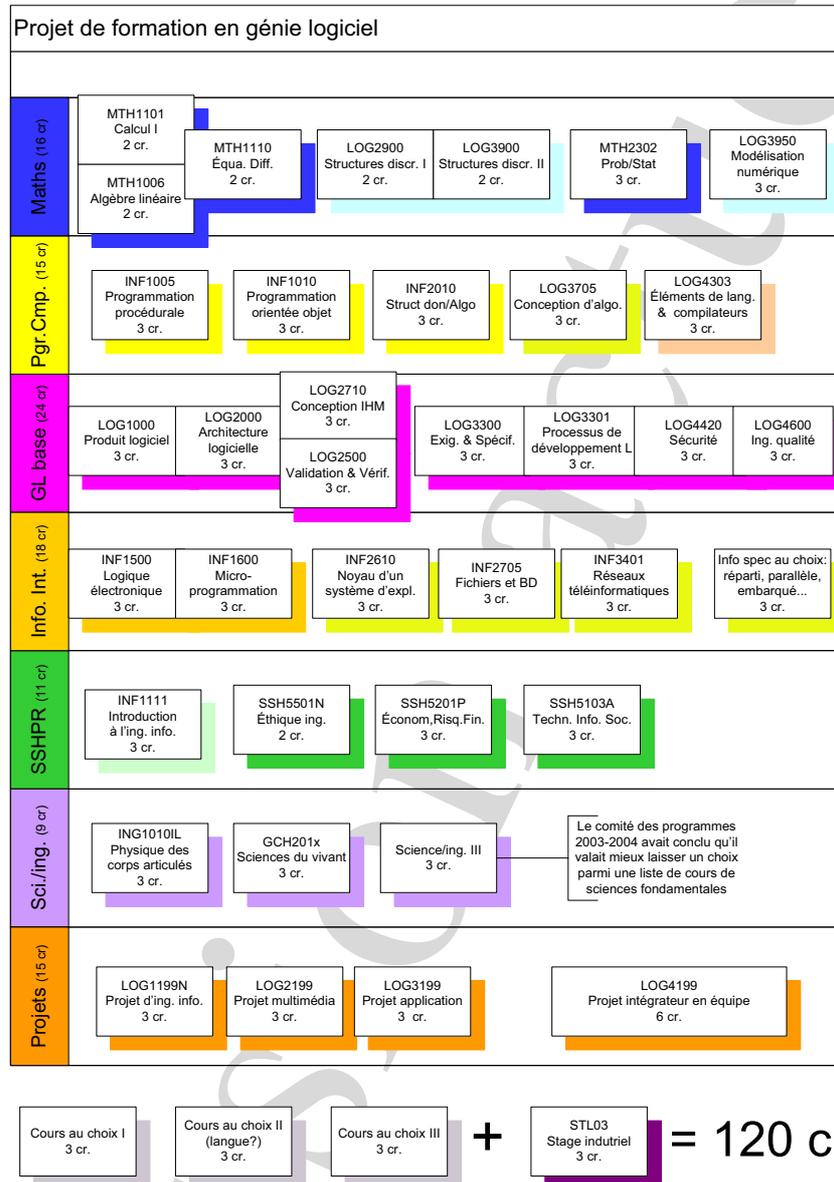
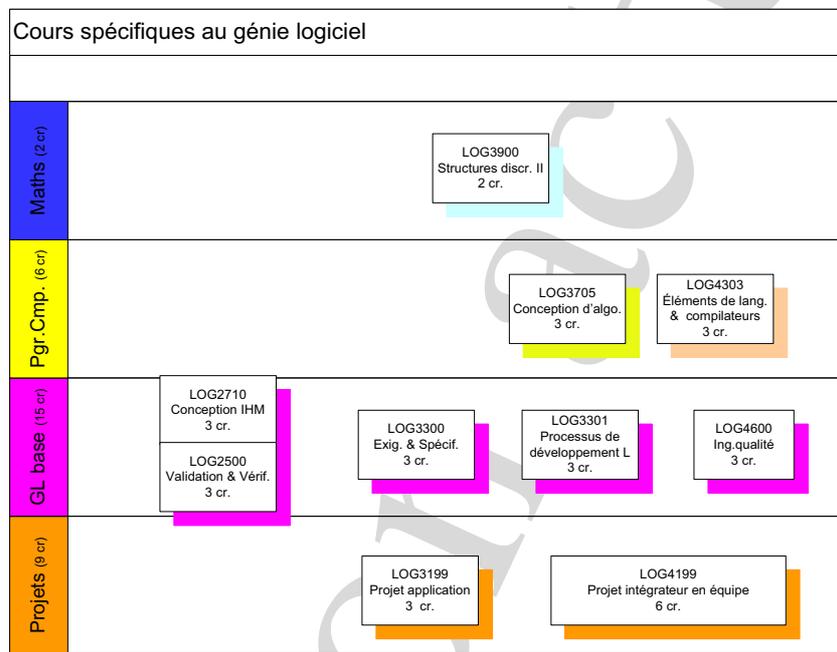


FIG. 2.12 – Cours mis en place avec le programme de génie informatique



= 32 cr

FIG. 2.13 – Cours spécifiques au programme de génie logiciel

## Chapitre 3

# Forces incontournables citées dans le Cahier des charges

Dans ce chapitre, nous expliquerons comment le nouveau programme remplit les exigences du cahier des charges.

### 3.1 Formation scientifique solide

Pour ce qui est des *mathématiques*, il y a des besoins diversifiés soit du côté des applications du génie logiciel, dont plusieurs requièrent des connaissances en mathématiques du continu, soit du côté de la pratique comme telle. Les cours de base MTH1005 - *Calcul I*, MTH1006 - *Algèbre linéaire* et MTH1003 - *Équations différentielles* sont conservés, ainsi qu'un cours plus spécialisé MTH2302 - *Probabilités et statistiques*. Puisque l'enveloppe des crédits destinés aux mathématiques n'est pas illimitée, il a été nécessaire de faire certains compromis dans la formation : réduire le poids des équations différentielles pour faire place aux structures discrètes (cours LOG2900- *Structures discrètes I* et LOG3900 - *Structures discrètes II*), une importance essentielle en génie logiciel, et enseigner les méthodes d'analyse numérique dans le contexte plus approprié de la modélisation de systèmes informatiques (cours LOG3950 - *Modélisation numérique*). Au total, la formation en mathématiques sera aussi solide qu'auparavant, mais beaucoup mieux contextualisée aux besoins des futures ingénieures ou des futurs ingénieurs développant des produits logiciels.

Pour ce qui est des *sciences fondamentales*, la formation s'étend sur trois cours obligatoires. Le premier cours obligatoire, à savoir ING1010IL *Physique des corps articulés*, vise la compréhension des phénomènes et lois physiques permettant de modéliser et analyser le comportement dynamique des artefacts technologiques (en particulier, pour le rendu de la réalité virtuelle et la cohérence des interfaces haptiques). Le second cours obligatoire, à savoir GCH201x *Sciences du vivant*, vise la couverture de certaines notions de biologie, possiblement importante dans les applications de la bioinformatique ou d'autres spécialités du génie proches des sciences de la santé. Le troisième cours est au choix parmi une liste fermée ; il permet de répondre à l'aspiration de l'étudiant et d'élargir l'horizon de ses intérêts scientifiques. Cette liste (non définitive à ce jour) pourra contenir des cours sur des sujets tels que : la physique des composants électroniques, les

champs électromagnétiques, et l'optique ou la photonique.

En plus des cours de l'informatique de base dispensée dans notre programme, le contenu des cours de génie informatique et de génie logiciel est plus axé sur les concepts fondamentaux et non les technologies. Dans la chaîne de cours de programmation, le langage de programmation C++ a été choisi comme véhicule, mais les notions enseignées couvrent la philosophie orientée objet de façon plus élargie. Par conséquent, le langage est utilisé essentiellement pour mettre en application les concepts à travers les travaux pratiques. De la même façon, dans le cours d'interfaces graphiques, une librairie est choisie pour développer une interface pour le projet mais plusieurs librairies sont exposées et discutées en classe afin de fournir aux étudiants des concepts génériques qui sont indépendants des outils utilisés. De cette façon, l'étudiant aura acquis une formation scientifique solide qui lui permettra d'apprendre les nouvelles technologies utilisées sur le marché du travail avec une grande souplesse. En effet, il est irréaliste de vouloir exposer l'étudiant à toutes les technologies existantes, ce qui amène à présenter plutôt les principes de base qui ont donné naissance à ces différentes technologies.

En ce qui a trait à la spécificité 1 qui demande une bonne couverture des fondements de l'informatique, le nouveau programme de génie logiciel propose également une chaîne cohérente de cinq cours d'informatique qui permet un survol des principaux concepts fondamentaux :

**INF1500 – Logique électronique**  
**INF1600 – Micro-programmation**  
**INF2600 – Noyau d'un système d'exploitation**  
**INF2705 – Fichiers et bases de données**  
**INF3401 – Réseaux téléinformatiques**

Dans cette approche, le futur ingénieur est amené à accumuler des connaissances fondamentales d'informatique.

## 3.2 Formation à la conception

Une contrainte forte que l'équipe pédagogique s'est donnée lors des premières discussions sur le projet de formation dès l'hiver 2003 et qui a dicté l'ordre de présentation des cours de spécialité du programme est la suivante :

Envoyer en stage des étudiants pouvant effectuer une bonne conception logicielle documentée avec un plan de tests rigoureux.

Cette contrainte est importante sur le plan de nos relations avec les industriels. Avec un stage industriel en troisième année, les implications de cette contrainte sont de présenter un certain nombre de cours de spécialité très tôt dans le programme, notamment :

**LOG1000 – Produit logiciel** Cours qui présentera une introduction à l'ingénierie des tests logiciels, à la spécification des exigences et au processus de développement logiciel.

**LOG2000 – Architecture logicielle** Cours où on apprendra à utiliser de façon cohérente des patrons de conception lors de l'élaboration d'une architecture logicielle, et qui fera un survol des langages et standards de conception.

**LOG2500 – Validation et vérification** Cours qui présentera les principales méthodes employées pour valider et vérifier un logiciel.

**LOG2710 – Conception d'interface personne-machine** Cours où on apprendra à utiliser les pratiques de la conception d'interface personne-machine.

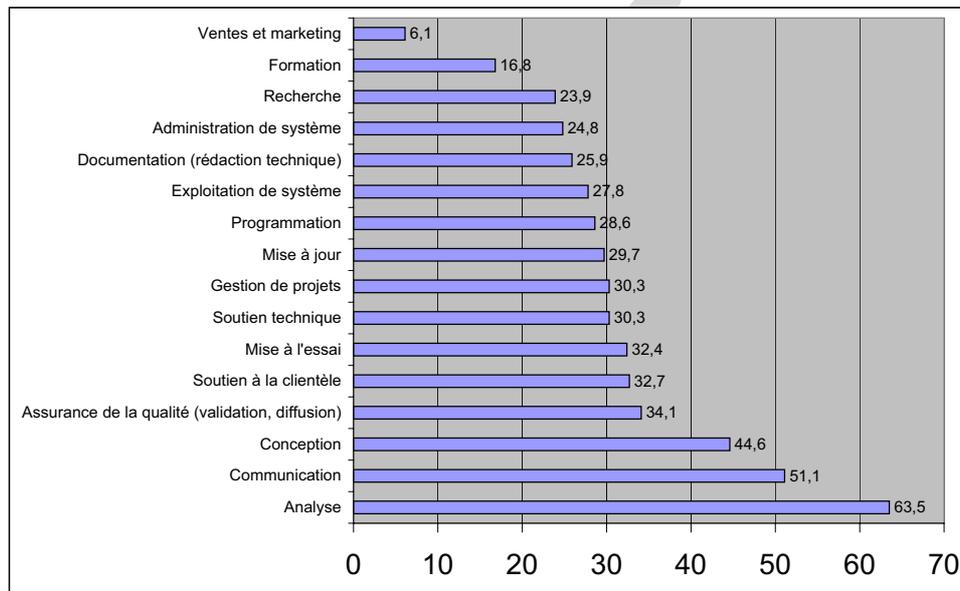


FIG. 3.1 – Pourcentage des employés qui jugent “très importantes” les compétences données (secteur privé)

Après ses deux premières années, l'étudiant aura donc suivi ces quatre cours de spécialité, les trois nouveaux cours de programmation et participé à un projet d'équipe en conception multimédia. Il sera donc bien équipé pour s'immerger dans un milieu industriel. Dans une étude récente de l'industrie des technologies de l'information, le Conseil des ressources humaines du logiciel (CRHL) du Canada a tenté de recueillir un portrait juste de la main-d'oeuvre et du marché du travail en interrogeant 35000 employés.[10] Toute une variété de questions ont été posées tant aux employés qu'à leurs employeurs pour tenter de comprendre les facteurs importants de l'industrie. Une question demandait aux employés d'identifier les champs de compétence jugés importants

(question non exclusive). Les résultats de la figure 3.1 font ressortir que l'analyse, la communication et la conception sont jugées très importantes par une grande proportion des employés du secteur privé.

La pratique industrielle acquise lors du stage sera renforcée par le projet intégrateur de troisième année qui vise à enrichir l'expérience pratique des équipes de développement logiciel par l'apprentissage d'une juste définition des exigences d'un produit logiciel et la recherche des spécifications précises de celui-ci. Le projet intégrateur de quatrième année vise l'accomplissement et la maîtrise de la profession d'ingénieur logiciel, dans le contexte d'une équipe de travail. À chaque année du programme, un projet intégrateur est donc destiné à parfaire la formation à la conception et à la réalisation de systèmes logiciels réalistes dont l'envergure s'accroît avec la maturité de l'étudiant. Ces projets intégreront l'ensemble des matières acquises durant l'année et permettront aux étudiantes et aux étudiants de les mettre en application. En plus des 15 crédits attribués à ces cours-projet, la plupart des cours de spécialités comporteront des heures en laboratoire, qui permettront de mettre en pratique divers outils facilitant la conception et la réalisation efficace de systèmes logiciels.

### 3.3 Habiletés personnelles et relationnelles

Le fait que la première année soit commune aux programmes de génie logiciel et de génie informatique est un atout important qui permettra aux futurs ingénieurs d'avoir une vue plus étendue du spectre d'applications du travail d'ingénierie des systèmes complexes.

Étant donné que le travail en équipe est courant dans la vie professionnelle d'un ingénieur informaticien, notre programme envisage la préparation de nos étudiants à cette forme d'activité. À travers les cours-projet répartis sur les quatre années de formation, la pratique du travail en équipe permettra aux étudiants de développer les habiletés suivantes :[11]

- 1° leur capacité de partager le leadership dans un groupe,
- 2° leur sens de l'organisation, de l'appartenance à un groupe et du partage des tâches,
- 3° leur capacités à tenir des réunions efficaces, et
- 4° leur sens des responsabilités, de l'autonomie et de l'initiative.

À travers ces projets d'envergure qui seront réalisés par des équipes constituées d'au moins quatre personnes, les étudiants seront amenés à identifier et valoriser une cible commune, créer des relations interpersonnelles harmonieuses entre les membres de l'équipe, surmonter les obstacles qui entravent le bon déroulement des activités et s'engager sur tous les plans.

Afin de préparer les étudiants à cette nouvelle formule, le nouveau cours INF1111 (successeur de l'actuel ING1040) dispensé dès la première session aura pour objectif d'exposer les notions théoriques permettant la formation de bonnes équipes ; en l'occurrence, parmi les thèmes envisagés, on retrouve : veiller à établir un bon contact entre les membres de l'équipe, comparer les perceptions individuelles de la cible commune, définir et se partager les tâches et veiller à ce que chaque membre de l'équipe s'exprime et s'engage.

En pratique, de nouvelles méthodes d'évaluation des apprentissages seront élaborées de façon à favoriser une participation efficace des différents membres de l'équipe. Ainsi, chaque étudiant

sera amené à participer de façon positive, à exercer un leadership, à faire preuve d'attention envers les autres et à communiquer ses propositions avec une argumentation solide. Par conséquent, les méthodes d'évaluation doivent être en mesure d'estimer les capacités individuelles et collectives. Des concours, encourageant le meilleur projet pour chaque année de formation, seront instaurés afin de motiver les étudiants à s'engager pour une réalisation concrète. Depuis de nombreuses années, les étudiants du département de génie informatique ont remporté les premiers prix lors du concours annuel d'infographie tenu à Laval en France, dont celui de la dernière compétition au *Laval Virtual 2004*.

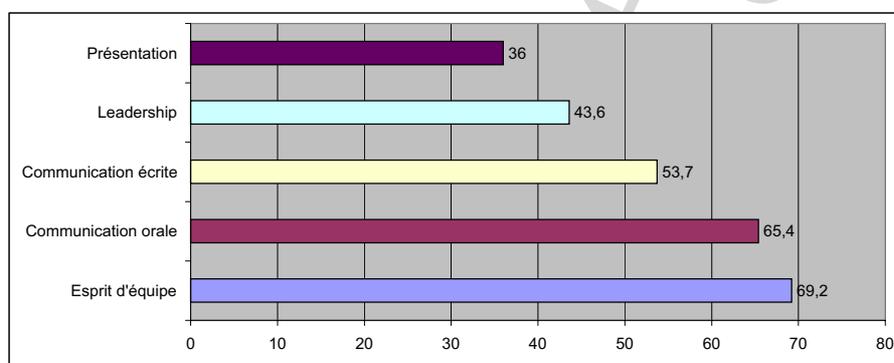


FIG. 3.2 – Pourcentage des employés qui jugent “très importantes” les aptitudes données (secteur privé)

De plus, les pratiques actuelles nous permettent déjà d'affirmer que le *travail en équipe* est bien ancré au département. En effet, les statistiques montrent que les programmes de génie logiciel et de génie informatique favorisent déjà une approche par projets pour plus de 15% des cours du curriculum. En continuité avec les travaux actuels, le département de génie informatique entend raffermir les habiletés particulières que sont la *communication* et le *travail en équipe*. Dans son étude récente de l'industrie des technologies de l'information, le CRHL a également demandé aux employés quelles aptitudes personnelles et interpersonnelles étaient très importantes.[10] Les résultats de la figure 3.2 font ressortir l'esprit d'équipe et la communication orale.

### 3.4 Aspect international du programme

Le nombre d'institutions offrant un curriculum complet de génie logiciel reste très limité à l'extérieur du Canada. En France, certaines composantes plus formelles du génie logiciel sont souvent enseignées dans des départements d'informatique. Parmi les spécialisations d'intérêt en génie logiciel, il y a quand même tout le champ des domaines d'applications.

Tel que mentionné au chapitre précédent, l'orientation *Multimédia* sera supportée par le nouveau programme. Il est possible que cette orientation se transforme à terme en une concentration. Le département de génie informatique a déjà entrepris un balisage partiel des exigences du mastère *Technologie Multimédia*, offert à l'*INP-Toulouse*. Une année de spécialisation pourrait être envisagée à Toulouse. Dans le cas d'une telle collaboration avec Toulouse dans le domaine du multimédia, il faudrait cependant rajouter un cours de traitement de signal, pour accroître les connaissances mathématiques.

Au Canada anglais, un certain nombre d'universités offrent des programmes de génie logiciel. Il serait également possible d'envisager d'y envoyer nos étudiants, mais aucun repérage n'a été fait encore sur ce sujet.

# Chapitre 4

## Autres composantes du programme

### 4.1 Intégration des matières

Le moyen que nous mettrons de l'avant pour favoriser l'intégration de la matière est basé sur un *modèle horizontal de coordination* des cours d'une même session. Ce modèle se distingue du modèle traditionnel de coordination par matière. De plus, un coordonnateur prendra en charge la coordination des activités pédagogiques dans l'environnement d'apprentissage des dix cours de première année suivis par le groupe d'étudiants génie informatique/génie logiciel. En termes de nombre de professeurs ou chargés de cours à superviser, le nombre reste sensiblement le même que dans l'ancien modèle de coordination vertical du tronc commun, environ dix personnes. Il s'agit d'amener les professeurs à mieux connaître les sujets abordés par leurs collègues des autres cours, d'amener les professeurs et chargés de cours à construire sur les apprentissages spécifiques que s'emploient à réaliser les étudiants dans les autres cours. Plus encore, nous voulons que les professeurs et chargés de cours traitent régulièrement les sujets de leur discipline en regard des difficultés requises par le projet intégrateur. Le coordonnateur fera partie du *Comité qualité* dont nous reparlerons dans la section 4.9.

Le modèle horizontal de coordination des cours ouvre d'immenses possibilités pédagogiques en ce qui concerne l'intégration pluridisciplinaire des matières, la diminution de la charge de travail pour les étudiants, l'évaluation transdisciplinaire des apprentissages, le suivi et l'encadrement individuel de chaque étudiant, le renouvellement des moyens d'évaluation des apprentissages, etc. Par la coordination horizontale des cours, nous croyons aussi pouvoir contrôler plus aisément l'interaction dynamique (qui nous échappe actuellement) dans l'environnement d'apprentissage des étudiants de certains aspects disciplinaires, pédagogiques, humains, physiques et même administratifs.

### 4.2 Caractère pratique accentué

L'enseignement en génie s'inscrit de plus en plus dans la voie de la pédagogie active, centrée sur l'étudiant plutôt que l'enseignant. Pour ce faire, nous exploiterons trois stratégies qui permet-

tront à l'étudiant de s'engager pleinement, soit :

- 1° l'apprentissage par problèmes,
- 2° l'apprentissage collaboratif, et
- 3° l'approche par projet.

Par les activités proposées, nous changerons la forme passive-réactive des cours vers une pédagogie plus active, plus stimulante et

Ceci afin de répondre aux besoins de motivation et d'engagement des étudiants à faire quelque chose de concret et de pertinent dans les cours, aux besoins de contacts humains plus prononcés avec leurs pairs et leurs professeurs. Dans la majorité des cours de spécialité, nous prévoyons des heures de laboratoire où l'étudiant pourra mettre en pratique les notions enseignées. Ces laboratoires favorisent l'acquisition de savoir-faire en proposant à l'étudiant une situation de résolution de problèmes. Nous estimons que ces laboratoires constituent une grande force de nos programmes et nous entendons poursuivre dans cette voie.

### 4.3 Réduction de la charge de travail étudiante et du contenu dans les cours

La grille des cours proposée à la figure 2.2 montre le souci d'offrir une charge de travail réaliste ainsi que des contenus de cours adéquats. Notre modèle systémique de coordination des cours d'une session permettra également de s'assurer de présenter les notions d'un cours au bon moment selon le déroulement des autres cours.

Afin de favoriser l'insertion des étudiants de première année, nous offrons une première session de 13 crédits et une deuxième session de 14 crédits. De façon globale, la matière des cours a également été allégée afin de permettre un approfondissement des concepts fondamentaux. Nous passerons maintenant en revue certains de ces allègements.

#### 4.3.1 Les cours de programmation

La chaîne de cours actuelle comporte deux cours de base pour aborder les techniques de programmation. Ces cours sont les suivants :

**ING1025 – Informatique** Cours qui introduit les concepts de base en programmation (typage, structure de contrôle, pointeurs, fichiers...) Le langage C y est principalement abordé. Une introduction au C++ est également faite.

**INF1101 – Algorithme et structure de données** Cours qui regroupe à la fois la présentation du paradigme orienté objet et un survol des structures de données élémentaires.

Les études de la charge des étudiants et la difficulté des coordonnateurs à couvrir la matière du cours INF1101 ont généré des discussions au comité des programmes du département. Afin d'alléger la charge de l'apprentissage des éléments fondamentaux de programmation, il a été suggéré de répartir le matériel à l'intérieur de trois cours suivant l'approche *imperative-first* couramment

utilisée dans les universités en Amérique du Nord.[2, 3] La chaîne de cours du nouveau programme serait maintenant :

**INF1005 – Programmation procédurale** Cours qui introduira les concepts de base en programmation procédurale (typage, structure de contrôle, ...) Le langage C y serait encore abordé.

**INF1010 – Programmation orientée objet** Cours qui s’occuperait principalement de la présentation du paradigme orienté objet et des bonnes pratiques de ce paradigme.

**INF2010 – Struct. de données et algorithmes** Cours qui ferait un survol de la programmation des structures de données élémentaires et enseignerait leur application dans un cadre algorithmique.

Cette approche permettra à la fois :

- de réduire la charge globale en première année des programmes de génie logiciel et de génie informatique ;
- d’approfondir les bonnes pratiques de programmation afin de permettre aux étudiants de maîtriser plus rapidement les principes de la programmation.

De plus, le département de génie informatique s’engage à supporter ces cours de façon à les rendre accessibles aux autres programmes qui ont besoin d’une introduction à la programmation.

### 4.3.2 Les mathématiques spécialisées

La chaîne de cours actuelle comporte deux cours de base qui permettent d’aborder les mathématiques spécialisées. Ces cours sont les suivants :

**LOG2900 – Éléments de structures discrètes** Cours qui introduit les concepts de base en mathématiques discrètes dans une perspective où les étudiants ont déjà des acquis en structures de données (le préalable étant INF1101) ;

**MTH2210 – Calcul scientifique pour ingénieurs** Cours générique offert par l’unité *mathématiques* qui présente l’analyse numérique sans connaissance préalable en programmation, mais qui utilise MATLAB comme support.

Afin d’alléger la charge de l’apprentissage des éléments fondamentaux nécessaires de ces mathématiques de spécialité, il est ici suggéré de répartir l’essentiel des parties utiles de ce matériel à l’intérieur de trois cours plus appropriés pour les étudiants du programme. Le curriculum IEEE/ACM du génie logiciel recommande deux cours de structures discrètes que nous redéfinirons dans le contexte du projet de formation.[2, 3] La chaîne de cours du nouveau programme serait maintenant :

**LOG2900 – Structures discrètes I** Cours qui introduit les concepts de base en mathématiques discrètes dans une perspective où les étudiants *n’ont pas* les acquis en structures de données. En fait, le troisième cours de programmation *INF2010 – Structures de données et algorithmes* serait corequis et les apprentissages seraient faits en juste-à-temps entre les cours LOG2900 et INF2010.

**LOG3900 – Structures discrètes II** Cours qui introduirait la modélisation computationnelle, soit les langages et les machines à état fini, ainsi que la théorie des groupes et les principes de l’encodage de l’information. Ce cours ne serait obligatoire que pour le programme de génie logiciel et servirait d’appui aux méthodes formelles.

**LOG3950 – Modélisation et simulation numérique** Cours qui introduirait les limites du calcul point flottant et les incidences sur les erreurs de modélisation. La perspective serait toutefois celle du génie logiciel et du génie informatique, et le cours de micro-programmation INF1600 serait un préalable.

## 4.4 Méthodes pédagogiques et étudiants responsables

Lors de la première année du programme de génie logiciel, un projet FRAPP permettra d’intégrer plus fondamentalement l’ensemble des cours, permettant ainsi aux étudiants de mieux réaliser leur insertion dans le monde universitaire. Le choix d’avoir une année commune entre le programme de génie logiciel et celui de génie informatique devrait faciliter l’insertion de nouveaux étudiants à l’École. Les objectifs plus spécifiques du projet départemental sont :

- 1° de personnaliser et d’humaniser davantage la formation en première année ;
- 2° d’augmenter la perception de la pertinence des cours de première année par les étudiants ;
- 3° d’augmenter le plaisir d’apprendre chez les étudiants et d’enseigner chez les professeurs et chargés de cours ;
- 4° de contribuer à donner aux étudiants de première année une réputation plus positive et même franchement enthousiaste par rapport à la première année ;
- 5° d’intégrer les recommandations du comité COSME sur l’évaluation des apprentissages [9] ;
- 6° de diminuer la charge de travail des étudiants par une meilleure intégration des matières et des évaluations des cours de première année ;
- 7° de former des femmes, des hommes, des citoyens et des professionnels conscients, avertis et réfléchis.

### 4.4.1 La grille-horaire et l’absentéisme

Pour réaliser un bon environnement d’apprentissage, pour laisser du temps aux étudiants, pour diminuer leur charge de travail, il faudrait modifier substantiellement la nature de l’horaire hebdomadaire. Il serait approprié de regrouper les exposés magistraux et de laisser certains après-midis (et soirées !) aux étudiants pour leurs autres activités d’apprentissage : laboratoires, étude, exercices, problèmes intégrateurs, projets, rencontres des professeurs. Pour les projets intégrateurs annuels qui auront lieu aux sessions d’hiver, il serait bon de réserver une *même journée* dans l’horaire pour les étudiants de toutes les cohortes. Dans ce cas, il serait possible que les étudiants de cohortes différentes assistent à certaines rencontres importantes. De plus, un telle approche permettrait d’étendre le champ des *relations personnelles inter-cohortes*. La présence lors de la *journée*

*projet* serait contrôlée, et une partie de l'évaluation finale serait établie par la liste des présences. Le tableau 4.1 donne une idée de l'horaire d'un étudiant. En effet, un ingénieur logiciel doit assister à ses rencontres avec le client et participer au travail de développement de l'équipe logicielle.

TAB. 4.1 – Semaine typique à l'hiver H1

	Jour 1	Jour 2	Jour 3	Jour 4	Jour 5 journée projet
AM	MTH1101 Équa. Diff.	INF1600 Micro-Pr.	INF1010 Prog. OO	libre	LOG1000 Produit L (Juste-à-temps)
PM	rencontre avec prof de maths	finaliser TP Micro. avec Marie	exercices de Prog.	pratique en équipe de l'exposé	LOG1199 Projet

## 4.5 Encadrement

Le projet pédagogique FRAPP mise sur une orchestration renouvelée de plusieurs principes partiellement exploités en première année :

- un modèle horizontal de coordination des cours ;
- un encadrement individuel accentué pour chaque étudiant ;
- des activités destinées à augmenter la motivation à apprendre des étudiants ;
- une forte intégration des matières (mono et pluridisciplinaires) ;
- des mécanismes d'évaluation des apprentissages différents de la pratique actuelle, diminuant la charge de travail (évaluation à caractère authentique) ;
- une utilisation des technologies de l'information et des communications (TIC) ; et
- une présentation des méthodes d'apprentissage de sorte à améliorer les stratégies des étudiants.

S'il y a un moment dans leur formation où les étudiants ont le plus besoin d'encadrement, c'est bien en première année. Le projet FRAPP vise à augmenter de plusieurs façons l'encadrement et le suivi de chaque étudiant. À chaque trimestre, chacun des professeurs ou chargés de cours impliqués ajoute à sa charge le suivi et l'encadrement individuel d'une dizaine d'étudiants du groupe classe, qu'il rencontrera sur une base très régulière (aux deux semaines, par exemple) pour faire le point sur la quantité de travail d'étude que chacun a fourni et sur la qualité du travail d'ensemble réalisé dans les cinq cours du trimestre. Dans le cas d'une difficulté disciplinaire, il pourra référer l'étudiant au collègue spécialisé de l'équipe, à un auxiliaire d'enseignement de la même spécialité ou encore à un collègue étudiant de son *équipe d'apprentissage*.

## 4.6 Évaluation des apprentissages

L'évaluation continue, telle que pratiquée dans beaucoup de cours au baccalauréat, peut être une approche pédagogique importante si elle est bien encadrée et qu'elle ne mène pas à une surcharge de travail de l'étudiant. Un des constats du rapport du Comité sur les modes d'évaluation (COSME) de 2001 est qu'il faudrait procéder à moins d'évaluations, mais qu'elles soient de plus grande envergure. Rappelons ces principes du rapport COSME qui sont encore d'actualité et qui seront appliqués dans le cadre du projet de formation en génie logiciel :[9]

- 1° viser l'amélioration plutôt que le contrôle ;
- 2° évaluer et noter les étudiants dans des tâches plus authentiques ;
- 3° afficher les critères d'évaluation dans tous les plans de cours ;
- 4° pondérer fortement le *raisonnement* ;
- 5° instaurer une collaboration inédite entre les professeurs.

Pour améliorer concrètement les procédures d'évaluation, il faut de toute façon également transformer les critères d'évaluation. Dans ce sens, les projets, dit intégrateurs, devraient avoir une importance considérable dans la promotion par année. Il serait *a priori* impossible de participer simultanément à deux cours-projets, et l'échec d'un bloc de cours impliquant des apprentissages nécessaires à un cours projet pourrait amener d'importantes distortions dans le cheminement, empêchant alors probablement la complétion des études sur une période de quatre ans.

### 4.6.1 Schéma en J

Présenté à la figure 4.1, le *schéma en J* regroupe :

**Cours préalable au projet** Ce cours préalable qui présente des notions importantes à la session précédant le projet intégrateur. Voici quelques particularités de ce cours :

- l'évaluation de ce cours ne sera que partielle ;
- les exercices y seront importants, car l'étudiant devrait démontrer sa capacité à exprimer correctement des éléments structuraux de solution.

Ce cours préalable est donc un cours qui utilise essentiellement le principe de l'*évaluation initiale*.

**Cours corequis au projet** Ce cours corequis qui présente des notions théoriques pertinentes au projet dans une demi-journée (probablement en juste-à-temps) ; dans l'autre demi-journée, le projet a lieu. Ce cours corequis s'inscrit dans une perspective de *formation*.

**Projet intégrateur** Ce projet met évidemment l'accent sur l'*application* des apprentissages des deux cours précédents. Ce cours évalue l'avancement dans le degré de maturité de l'étudiante ou de l'étudiant. L'évaluation de ces trois cours est donc *cumulative*, et elle part du principe qu'il faut laisser un délai entre les apprentissages et les renforcer entre les trimestres pour instaurer des bonnes pratiques.

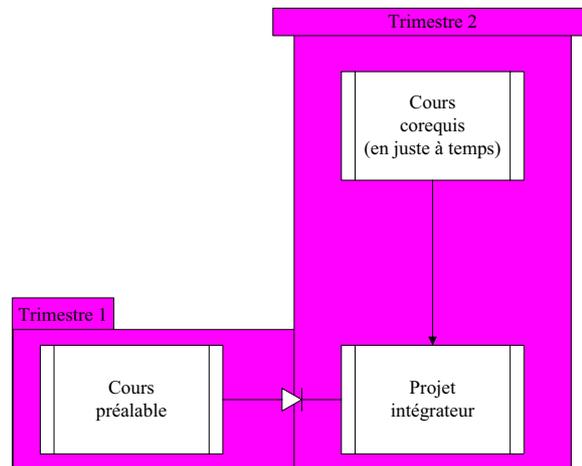


FIG. 4.1 – Schéma en J : arrimage de trois cours pour assurer une continuité intertrimestre

Cette transformation du mode d'évaluation est générique et elle s'appliquera *a priori* aux quatre années du programme. Comme on le voit au tableau 4.2, l'acquisition de la matière de certains cours sera contextualisée par les projets. Toutefois, le projet de quatrième année sera moins encadré et pourra s'accommoder au contexte industriel tel que mentionné à la prochaine section.

Voici un exemple concret d'application pratique en première année. Le cours INF1111 présentera des artefacts utilisés pour la communication graphique en génie logiciel et en génie informatique ; le cours présentera également la méthodologie standard de production de rapport (documentation, références, structure). Ces éléments seront évidemment évalués en première session ; mais, il s'agira ici d'effectuer une *évaluation initiale* puisque l'objectif serait ici l'apprentissage à long terme des notions. Toutefois, ils seront *vérifiés* en deuxième session, via les documents produits en support au projet intégrateur de première année. Un étudiant ayant eu une note faible en INF1111 pourra *améliorer* sa note s'il est capable de démontrer l'acquisition de compétences accrues. Les trois volets du cours LOG1000 (ingénierie des tests, spécification, introduction au processus de développement) pourront être *vérifiés* et évalués par des composantes du projet LOG1199, telles que le plan de test, les documents de spécification, le suivi du projet logiciel. Évidemment, on s'attend à ce que les étudiants de génie logiciel assument une responsabilité accrue sur ces composantes par rapport à ceux de génie informatique. On pourrait alors calibrer différemment les cotes pour les étudiants des deux programmes.

TAB. 4.2 – Cours dont l'évaluation sera partagée

Année	Cours préalable	Cours corequis	Projet intégrateur
1 <sup>re</sup>	INF1111	LOG1000	LOG1199
	Introd. ing. info.	Sujets GL	Système matériel/logiciel
2 <sup>e</sup>	LOG2000	LOG2710	LOG2199
	Conc. log.	Conc. IHM	Multimédia
3 <sup>e</sup>	LOG3300	LOG3301	LOG3199
	Exig. spec.	Processus GL	Autre génie
4 <sup>e</sup>	LOG4600	Context.	LOG4199
	Ing. qualité	À déterminer	Industrielle

## 4.7 Projets intégrateurs

Avant de décrire les projets intégrateurs, précisons que la matière théorique nécessaire pour acquérir une connaissance en gestion de projet sera répartie en trois modules insérés dans les projets intégrateurs selon la pondération suivante :

**Module 1** (valant 0,5 crédit) Les principes de base en définition et planification de projet seront intégrés au projet de *deuxième année* ;

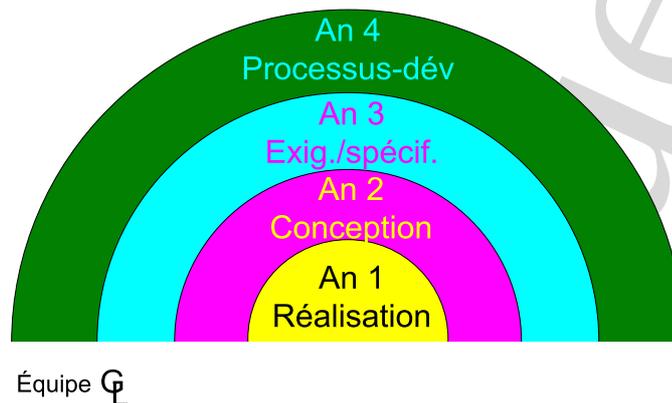
**Module 2** (valant 1 crédit) Les techniques et méthodes de gestion de projet seront intégrées au projet de *troisième année* ;

**Module 3** (valant 0,5 crédit) La gestion de projet dans un contexte d'entreprise sera intégrée au projet de *quatrième année*.

Des discussions ont lieu entre le Pr. Michel DESMARAIS, responsable au département des apprentissages en *gestion de projets informatiques*, et l'Unité études complémentaires pour déterminer comment seront partagés ces enseignements.

Dans la description des projets intégrateurs, nous utiliserons le modèle CMM qui s'applique normalement aux organisations, et non aux individus. [6] Cette description nous permettra de comprendre l'évolution de la maturité dans la formation prévue au curriculum. Il faut donc comprendre l'analogie de ce modèle avec le comportement d'une cohorte d'étudiantes et d'étudiants qui se déplacent d'année en année dans une entreprise modèle de mieux en mieux structurée.

### 4.7.1 Les thèmes des projets intégrateurs



Projet de formation en génie logiciel:  
Intégration des connaissances via un projet annuel

An 1. Réalisation: implémentation d'un logiciel  
An 2. Conception: à partir d'une spécification des exigences  
An 3. Exigences et spécifications: élaboration des besoins-clients  
An 4. Processus de développement logiciel

FIG. 4.2 – Thèmes des projets intégrateurs du nouveau programme de génie logiciel

Tel qu'illustré à la figure 4.2, les projets intégrateurs de chacune des années visent à améliorer le niveau de maturité du processus de développement logiciel de l'étudiant. En se référant au modèle classique d'évolution des capacités logiciel, appelé le CMM et développé au *Software Engineering Institute (SEI)*, les trois premiers parmi cinq niveaux de maturité sont définis pour qualifier la maturité au sein d'une organisation : [6]

**Niveau initial** Au niveau initial, l'organisation ne fournit habituellement aucun environnement stable de développement et de maintenance logiciel.

**Niveau reproductible** La capacité du processus logiciel des organisations de ce niveau peut être définie comme étant structurée, car la planification et le suivi des projets logiciels sont stables et les réussites antérieures peuvent être reproduites.

**Niveau défini** Au niveau défini, le processus standard de développement et de maintenance logiciel à travers toute l'organisation est documenté, intégrant en un tout cohérent les processus d'ingénierie logiciel et de gestion. Ce processus standard est régulièrement mentionné dans le CMM comme étant le processus logiciel standard de l'organisation.

Les deux autres niveaux de maturité, **niveau maîtrisé et d'optimisation**, pourront certes être enseignés théoriquement dans les cours de processus ou d'assurance qualité, mais il ne serait pas

TAB. 4.3 – Maturité acquise dans les projets intégrateurs annuels

Année	Application	Impact logiciel	Changement de maturité
1 <sup>re</sup>	Système matériel/logiciel	Réalisation	Initial
2 <sup>e</sup>	Multimédia	Conception	Initial $\implies$ Reproductible
3 <sup>e</sup>	Autre génie	Spéc./Exigences	Reproductible $\implies$ Défini
4 <sup>e</sup>	Industrielle	Processus dév.	Défini++

TAB. 4.4 – Volets techniques accentués dans les projets intégrateurs annuels

Année	Emphase logiciel	Pratique accentuée
1 <sup>re</sup>	Bonne réalisation	Ingénierie des tests logiciels
2 <sup>e</sup>	Bonne conception	Ingénierie de la conception
3 <sup>e</sup>	Spéc./Exigences	Ingénierie des exigences logicielles
4 <sup>e</sup>	Processus dév.	Planification, qualité, sécurité, ...

réaliste de tenter d'implanter de telles pratiques à l'intérieur d'un curriculum de quatre années d'études en ingénierie logicielle. Notre objectif est d'amener les étudiants à maîtriser les concepts du niveau défini au travers la suite des apprentissages des cours projets et du stage industriel.

Pour ce faire, les projets annuels ont des composantes thématiques diversifiées, tant du point de vue des applications à développer, du type d'ingénierie logicielle privilégiée que des objectifs d'évolution de capacité de développement. Le tableau 4.3 illustre ces composantes. En particulier, mais non exclusivement, certaines connaissances pratiques de base du génie logiciel seront mises de l'avant à chaque année, et ces connaissances seront essentielles pour un travail articulé dans les années suivantes. Ces données sont illustrées au tableau 4.4. Rappelons qu'il faut faire un lien entre cette démarche d'accumulation de savoir par les projets et la position du stage dans le curriculum de l'étudiant, telle que présenté à la section 3.2.

#### 4.7.2 La pierre angulaire de la quatrième année

Le projet de quatrième année vise l'accomplissement et la maîtrise de la profession d'ingénieur logiciel. Le projet devra inclure le développement du système logiciel significatif utilisant la connaissance gagnée des cours dans tout le programme. Ce système logiciel nécessitera une spécification des exigences, une bonne conception, une réalisation, ainsi qu'une garantie de qualité. Les étudiants peuvent suivre n'importe quel modèle de processus approprié, devront prêter une attention particulière aux impacts de qualité, et doivent contrôler le projet eux-mêmes, suivant les techniques appropriées de gestion de projet. Le succès du projet est déterminé en grande partie par le fait que les étudiants auront en juste proportion résolu le problème du client.

Les étudiants devraient pouvoir fournir une ou plusieurs itérations d'un système logiciel, avec

tous les artéfacts appropriés au modèle de processus qu'ils emploient. Ceux-ci incluraient probablement un plan de projet (peut-être mis à jour régulièrement, et contenant des évaluations de coût, l'analyse de risque, la division du travail dans des tâches, etc.), des exigences (y compris un cas d'utilisation), une documentation d'architecture et de conception, des plans d'essai, le code source et le système installé.

On ne prévoit pas que ce cours ait des séances magistrales formelles, bien qu'on s'attende à ce que des étudiants assistent à des présentations de progrès des autres groupes. Il serait hautement souhaitable que les étudiants soient requis d'avoir un client pour lequel ils développeraient leur logiciel. Ce client pourrait être une compagnie, un professeur ou plusieurs ingénieurs de domaines de compétence variés choisis en tant que représentation des personnes sur le marché potentiel. L'objectif du projet serait de résoudre le problème du client, et ce dernier aiderait l'instructeur en évaluant le travail. *Les étudiants travaillent sur ce projet dans des groupes d'au moins trois ou quatre.* Des stratégies devront être développées pour stabiliser les situations où la contribution des membres de l'équipe est inégale. Dans le cas où on choisira de diviser ce cours en deux parties, un cours de trois par trimestre par exemple, si les étudiants ne finissent pas le projet (c.-à-d. la seconde partie des deux cours), alors ils devront recommencer à partir du premier cours.

Le projet intégrateur de la dernière année d'études pourra donc être adapté aux besoins du marché logiciel. Parmi les composantes intéressantes, nous citerons :

- l'intégration possible à l'intérieur d'une équipe de développement logiciel en vue de produire un logiciel industriel à l'intérieur des murs de l'École. C'est ce que nous appelons le *cours clinique* ;
- la possibilité d'effectuer un projet de 4<sup>e</sup> dont le sujet soit en relation avec le stage industriel prévu en 3<sup>e</sup> ; le projet pourrait alors s'échelonner sur un ou deux trimestres.

## 4.8 Stages

Tel que soulevé lors de la 79<sup>e</sup> Commission des études du 1<sup>er</sup> novembre 2004, il est difficile de préciser la session exacte du stage obligatoire dans le curriculum afin de tenir compte à la fois :

1. de la réduction de charge nécessaire afin de faciliter la rétention à la première année ;
2. de la contrainte que le stage doive s'effectuer après un minimum de 60 crédits ;
3. de la nécessité de limiter la formation à quatre années ;
4. des contraintes du marché de l'emploi.

Dans le cheminement actuel,  $27 + 29 = 56$  crédits auront été accumulés après les deux premières années complètes. Évidemment, l'exemption en MTH1101 pourrait permettre de gagner des crédits supplémentaires. Toutefois, il est probable que la session A3 doive être réalisée par la majorité des étudiants avant de partir en stage, soit à la session H3 ou à la session E3. Une solution globale serait d'offrir la session A3 régulière en E2, ou au moins une partie de celle-ci.

Un certain nombre de dérogations sont également à prévoir pour certains étudiants, parmi celles-ci mentionnons :

**Stage humanitaire** le stage s'effectuerait alors auprès d'une organisation humanitaire ;

**Stage international** dans ce contexte, le stage s'effectuerait à l'extérieur du Québec, et ce particulièrement dans le cadre de programme d'échange ;

**Stage à l'interne** cette catégorie serait d'intérêt pour la préparation et la participation à des compétitions des sociétés savantes.

## 4.9 Mécanismes d'évaluation continue du programme

Nous comptons créer pour le nouveau programme, un *Comité qualité*, différent du *Comité des programmes*, pour évaluer la qualité de la formation et l'implantation du projet éducatif. Un comité de ce type, comprenant des professeurs et des étudiants, a été implanté au département de génie physique depuis plusieurs années et a contribué à alléger et à mieux répartir la charge de travail des étudiants.

Le Comité qualité devra assurer la cohérence des cours d'une chaîne définie par les liens de préalable et corequis, et la complémentarité entre ces chaînes, faire le suivi de l'implantation des projets intégrateurs (années 1, 2 et 3) et du projet de conception final (année 4), et superviser la coordination des cours d'un même semestre ou d'une même année du programme. Ceci pourra nécessiter la définition des fonctions de responsable de chaîne de cours, responsable de semestre ou année, responsable et coresponsable de projet intégrateur. C'est donc un défi organisationnel et motivationnel intéressant pour la dynamique de groupe d'un corps professoral d'une trentaine de membres.

## 4.10 Modalités pour le passage aux études supérieures

Sans connaître les nouvelles règles de passage aux études supérieures, il est quand même possible de parler de la souplesse de la quatrième année à cet égard. En supposant que le cours LOG4420 sur la sécurité informatique se donne aux sessions d'automne et d'hiver (ce qui est probable puisqu'il devient obligatoire dans les programmes de génie logiciel et de génie informatique), un scénario de passage en mode BMI aux études supérieures en génie informatique serait le suivant :

**Complétion hâtive des exigences en A4** Le cours LOG4420 serait pris à l'automne plutôt qu'à l'hiver de la quatrième année. Le cours *Sciences III* devrait possiblement être pris à l'automne en tenant compte du sujet d'application aux études supérieures.

**Début de formation ÉS en H4** Tout ceci permet une liberté relative à la quatrième session. Les deux cours au choix et le cours d'informatique complémentaires ainsi libérés pourraient être choisis parmi les cours offerts aux études supérieures.

**Projet intégrateur de quatrième** Les balises de transformation du projet devraient être définies par la direction de l'École. Obligera-t-on la participation des étudiants de BMI aux équipes de quatrième année ?

# Bibliographie

- [1] “Projet de formation pour le baccalauréat en ingénierie à l’École Polytechnique de Montréal – Cahier des charges”, 19 janvier 2004.
- [2] *Computing curriculum – Software Engineering*, Public Draft 2.1 (December 11, 2003), Task force on Computing curricula, IEEE computing society, Association for computing machinery.
- [3] *Computing curriculum – Computer Engineering*, Ironman Draft (February 15, 2004), Task force on Computing curricula, IEEE Computing Society, Association for computing machinery.
- [4] *Software Engineering*, NATO Conference (October 7-11, 1968), Ed. Peter Naur and Brian Randell, Bruxelles, January 1969.
- [5] IEEE STD 610.12-1990, *IEEE Standard Glossary of Software Engineering Terminology*, IEEE Computer Society, 1990.
- [6] CMU/SEI-93-TR-24, *Modèle d’évolution des capacités logiciel, version 1.1*, Mark C. Paulk, Bill Curtis, Mary Beth Chrissis, Charles V. Weber, Software Engineering Institute, Carnegie Mellon University Pittsburgh, Penn., 1993.
- [7] “Enquête auprès des diplômés du baccalauréat (1998-2002) – Programme de génie informatique”, document préparé par Lina Forest, École Polytechnique de Montréal, octobre 2003.
- [8] “Programme coopératif en génie logiciel – Analyse et commentaires sur la situation”, document présenté par le CEGL (Comité étudiant en génie logiciel), 23 mars 2003.
- [9] “COSME – Rapport final”, document présenté à la Commission des études, 9 mars 2001.
- [10] “Le projet *Marché du travail canadien dans le secteur des TI – L’enquête nationale*”, avant-première dévoilée le 10 novembre 2004, voir détails : <http://www.shrc.ca/>
- [11] “La préparation d’un cours”, Richard Prigent, Éditions de l’École Polytechnique de Montréal (1990).

# Annexe A

## Description des diverses catégories dans le balisage

### A.1 Bloc général

Le bloc général couvre tous les secteurs qui ne sont pas propres au génie informatique ou au génie logiciel. Voici la correspondance des codes :

**MTH** : Mathématiques (général) : inclut toute la formation en mathématiques sauf celle incluse aux catégories DSC et PRS.

**DSC** : Mathématiques discrètes.

**PRS** : Probabilités et statistiques.

**SCI** : Sciences : couvre tous les cours de science générale.

**OEN** : Ingénierie, autre que GI/GL : couvre les cours d'ingénierie de type tronc commun, comme la formation en dessin technique traditionnel.

**ST** : Stages, projets spéciaux et séminaires : couvre les diverses activités d'intégration dont l'étendue ne permet pas de les restreindre à un petit nombre de catégories.

**SSH** : Études complémentaires optionnelles : couvre la formation complémentaire autre que celle imposée par le BCAPI, comme les cours de "liberal arts" au choix.

**FND** : "Engineering fundamentals" : couvre les activités d'ingénierie ou en support direct à l'ingénierie qui, sans être propres au génie informatique ou au génie logiciel, constituent des incontournables dans des programmes de ce type. Dans la plupart des cas, ces crédits ont trait aux aspects économiques de l'ingénierie.

**PP** : "Professional practice" : couvre des activités telles que la communication orale et écrite ou encore les notions de professionnalisme.

## A.2 Bloc GI/GL

Le bloc GI/GL couvre les notions techniques communes aux disciplines de génie informatique et de génie logiciel.

**CMP** : "Computing essentials" : couvre les notions fondamentales non couvertes par la catégorie PRF, incluant les outils et technologies de construction avancée ainsi que les méthodes formelles de construction.

**HCI** : "Human-computer interaction" : couvre les diverses notions inhérentes au domaine des interfaces-usagers et de l'infographie de base.

**PRF** : "Programming fundamentals" : couvre tous les aspects de base liés à la programmation, incluant les notions de base en algorithmique et structures de données.

**SAS** : Cours de spécialisation / orientation : couvre tous les cours à option généralement offerts en fin de programme.

On retrouve également une classification de la matière selon les champs de spécialité des deux programmes.

**Bloc GI** Ce bloc couvre toute la matière propre à la spécialité du génie informatique, à l'exclusion des éléments couverts dans les autres blocs.

**Bloc GL** Ce bloc couvre toute la matière propre à la spécialité du génie logiciel, à l'exclusion des éléments couverts dans les autres blocs.

## Annexe B

# Enquête du Conseil des ressources humaines du logiciel

Le 10 novembre 2004, le Conseil des ressources humaines du logiciel (CRHL) dévoilait certaines constatations observées lors d'une enquête pancanadienne sur les professions reliées aux technologies de l'information (TI). Cette enquête est marquante car elle couvre 70% de la population active en TI. En effet, 35 000 employés des secteurs public et privé et plus de 25 000 employeurs d'un peu partout au Canada y ont participé. L'ensemble des compétences nécessaires aux professionnels des TI y ont été scrutées du point de vue de la perception des employeurs et des employés. La portée de cette enquête permet de définir des profils professionnels plus précis afin que les employeurs et les employés puissent avoir les mêmes points de repère.

La figure B.1 donne la répartition des emplois telle qu'observée dans les secteurs public et privé. Il est à noter qu'une plus grande proportion d'*ingénieurs logiciels* semblent présente dans le secteur privé (9,9%) par rapport au secteur public (seulement 2,5%).

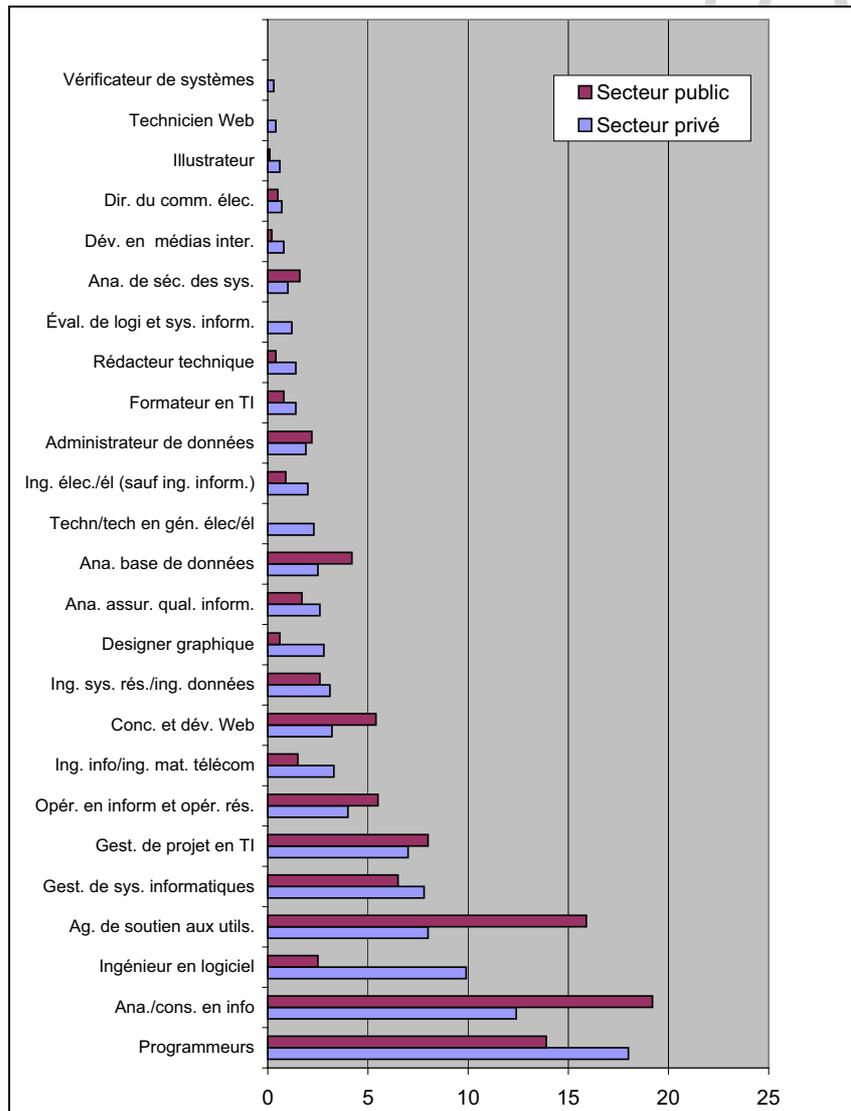


FIG. B.1 – Répartition (%) de la main d’oeuvre en TI par profession

# Index

GCH201x .....	5, 21, 31, 35	LOG3301 .....	21, 48
INF1005 .....	17, 30, 32, 43	LOG3900 .....	21, 35, 44
INF1010 .....	17, 30, 32, 43, 45	LOG3950 .....	21, 35, 44
INF1101 .....	42	LOG4199 .....	23, 48
INF1111 .....	17, 30, 38, 47, 48	LOG4199A .....	23
INF1500 .....	17, 30	LOG4199H .....	23
INF1600 .....	17, 30, 45	LOG4303 .....	23
INF2010 .....	19, 43	LOG4420 .....	23, 52
INF2600 .....	19	LOG4600 .....	23, 48
INF2705 .....	19	MTH1003 .....	35
INF3401 .....	21	MTH1005 .....	35
INF3705 .....	21	MTH1006 .....	17, 35
ING1003 .....	4	MTH1101 .....	17, 45, 51
ING1005 .....	3	MTH1110 .....	17
ING1006 .....	3	MTH2210 .....	43
ING1007 .....	4	MTH2302 .....	19, 35
ING1010 .....	4	spécif. 1 .....	3, 5, 36
ING1010IL .....	5, 19, 31, 35	spécif. 2 .....	3, 4
ING1015 .....	4	spécif. 3 .....	3-5
ING1020 .....	4	SSH5103A .....	23
ING1025 .....	3, 32, 42	SSH5201P .....	21
ING1030 .....	4	SSH5501N .....	21
ING1035 .....	4	STL03 .....	21
ING1040 .....	3, 38		
LOG1000 .....	17, 30, 37, 45, 47, 48		
LOG1199 .....	17, 30, 45, 47, 48		
LOG2000 .....	19, 37, 48		
LOG2199 .....	19, 48		
LOG2500 .....	19, 37		
LOG2710 .....	19, 37, 48		
LOG2900 .....	4, 19, 35, 43		
LOG3199 .....	21, 48		
LOG3300 .....	21, 48		