

# Détermination du bloc de connaissances en génie logiciel destiné aux ingénieurs du Canada

## 1.0 Introduction

**A**u Canada, l'exercice de la profession d'ingénieur est régi par des associations provinciales et territoriales. Créées en vertu d'une loi provinciale ou territoriale, ces associations sont habilitées à délivrer des permis d'ingénieurs aux candidats qui relèvent de leur compétence. Le Conseil canadien des ingénieurs (CCI) est l'organisme national qui chapeaute ces associations et assure leur coordination. Il exerce notamment cette dernière fonction en élaborant des lignes directrices. Il s'agit en fait de principes directeurs généraux faisant l'objet d'un vaste consensus, qui respectent et soutiennent l'autonomie de chacune des associations quant à leur capacité d'administrer leur législation respective en la matière.

Le Bureau canadien des conditions d'admission en génie (BCCAG) et le Bureau canadien d'accréditation des programmes d'ingénierie (BCAPI) sont deux des quatre comités de travail du CCI. Comme son nom l'indique, le BCCAG a pour mandat d'accréditer les programmes d'ingénierie de premier cycle qui permettent aux étudiants d'acquérir les connaissances requises pour se voir accorder un permis d'ingénieur au Canada. Le BCCAG, quant à lui, s'occupe principalement de l'élaboration des directives nationales sur les conditions d'admission en génie ainsi que des normes d'exercice et du code de déontologie de la profession. Il est également chargé de publier le Programme des examens du CCI, c'est-à-dire ceux qui ont pour objet d'évaluer le niveau de connaissances des personnes qui n'ont pas obtenu leur diplôme après avoir suivi un programme d'ingénierie accrédité par le BCCAG.

En septembre 2000, le BCCAG et le BCCAPI ont envisagé la possibilité d'élaborer un bloc de connaissances pour chaque discipline du génie, soit un ensemble constitué d'un bloc de connaissances de base (BDCB), d'un bloc de connaissances complémentaires (BDCC) ainsi que de toutes autres connaissances scientifiques, mathématiques et complémentaires fondamentales déterminées par le BCCAPI et le BCCAG pour toutes les disciplines du génie. Le BDCB d'une discipline donnée devrait comprendre toute la matière et tous les sujets que doit étudier toute personne appartenant à cette discipline. Les domaines de spécialisation ne faisant pas partie du BDCB constitueraient le BDCC. Un comité, dont les membres sont les auteurs du rapport préliminaire [1] résumé dans le présent document, a été formé en février 2001; il avait pour mandat de définir le BDCB et le BDCC destinés au génie logiciel. Au même moment, un autre comité a été mis sur pied et chargé de déterminer un bloc de connaissances en génie chimique. Le présent document fait état d'un certain nombre de points soulevés lors des délibérations des membres du comité et des conclusions auxquelles ils sont parvenus.

### 1.1 Définition du terme «génie logiciel»

La définition présentée ci-après est tirée de la définition du terme «ingénieur» énoncée dans le *Guide sur l'exercice de l'ingénierie au Canada*, produit par le CCI [2].

L'exercice du génie logiciel comprend des activités de planification, de conception, de création, d'évaluation, de consultation, d'établissement de rapports, de direction ou de surveillance, et de gestion de produits ou de procédés à fort contenu logiciel qui nécessitent l'application de principes d'ingénierie et mettent en cause la protection de la vie, de la santé, de la propriété, d'intérêts économiques, du bien-être public ou de l'environnement.

Les ingénieurs en logiciel s'intéressent à l'analyse, à la conception, à la programmation, à la mise à l'essai, à l'intégration, au soutien ainsi qu'à la mise en service et hors service des systèmes logiciels, en particulier ceux qui sont d'une importance capitale pour la santé et la sécurité du

par R.D. Dony - Université de Guelph, Guelph, Ontario  
P. Botman - True North Systems Consulting, Vanc., C.-B.  
W.E. Briggs - Université du N.-Brunswick, Fred., N.-B.  
R. Haggart - Olorin Enterprise Inc., Ottawa, Ontario  
P.A. Taylor - Université McMaster, Hamilton, Ontario

## Sommaire

Dans le but de définir les exigences en matière de formation requises l'obtention d'un permis d'ingénieur au Canada, le Bureau canadien d'accréditation des programmes d'ingénierie a formé un comité dont le mandat était de définir les blocs de connaissances de base et de connaissances complémentaires en génie logiciel. Certains des points soulevés lors des délibérations des membres du comité et les conclusions auxquelles ils sont parvenus sont présentés dans cet article. Après avoir examiné un certain nombre d'éléments, notamment les programmes d'études et d'examens provinciaux et nationaux accrédités en génie logiciel, les membres du comité ont retenu les sujets fondamentaux suivants : mathématiques discrètes, structures de données et algorithmes, développement de logiciel (processus de génie logiciel, exigences, conception, création, essais, maintenance et gestion de la configuration, etc.), fiabilité et sécurité des systèmes, systèmes numériques, architecture d'ordinateur, systèmes d'exploitation, fichiers et bases de données, systèmes et commandes. En général, ces sujets s'accordent bien avec les programmes d'examens et les programmes d'études existants. Les domaines complémentaires ne sont pas aussi bien définis; ils visent l'approfondissement des sujets de base et des domaines de spécialisation et l'élargissement des domaines d'application.

## Abstract

In response to the need to define the academic requirements for licensing professional engineers in Canada, the Canadian Engineering Qualifications Board formed a committee whose mandate was to define the core and supplemental bodies of knowledge for Software Engineering. Some of the issues and the outcome of the software engineering committee's deliberations are presented in this paper. After examining a number of inputs including national and provincial examination syllabi and curricula of accredited software engineering programs in Canada, a number of core topic areas were defined. These are: Discrete Mathematics, Data Structures and Algorithms, Software Development (includes Software Engineering Process, Requirements, Design, Construction, Testing, Maintenance, and Configuration Management), System Reliability and Safety, Digital Systems, Computer Architecture, Operating Systems, File and Database, and Systems and Control. These topics generally match well with the existing examination syllabi and curricula. The supplemental areas are less well-defined and include depth in core subjects, depth in areas of specialization, and breadth in application domains.

## Remerciements

Cet article est protégé par le droit d'auteur et a été reproduit avec la permission de IEEE. *IEEE Canadian Conference on Computer and Electrical Engineering*, CCECE 2002, Winnipeg, mai 2002.

**NDLR** : En raison de l'importance que revêt cet article pour notre industrie, nous l'avons reproduit en totalité. Vos commentaires sont les bienvenus.

public et pour l'environnement. Les ingénieurs en logiciel mettent en application des méthodes et des principes d'ingénierie reconnus afin de produire des systèmes fiables, destinés à l'utilisation prévue.

## 1.2 Portée

Le présent document traite, comme il se doit, des domaines de connaissance qu'on trouve dans d'autres disciplines du génie, et notamment de sujets faisant partie des «études de base» du programme du BCCAG tels que les équations différentielles, la statique, la dynamique, l'électromagnétique de base et même la thermodynamique et la mécanique des fluides. Les connaissances de base qu'ils acquièrent dans un vaste éventail de sciences fondamentales, à la fois dans le domaine des sciences appliquées et du génie, permettent aux ingénieurs de toutes les disciplines d'avoir un minimum de connaissances dans les domaines d'étude de leurs collègues. Comme de nombreux ingénieurs en logiciel exerceront vraisemblablement leur profession dans le cadre d'autres secteurs du génie, ces connaissances de base revêtent une importance particulière. Évidemment, nombre d'entre elles seront communes à une ou deux autres disciplines telles que le génie informatique et le génie électrique.

On insiste particulièrement sur le fait qu'un ingénieur en logiciel, outre les connaissances universitaires susmentionnées, doit posséder des connaissances pratiques et de l'expérience dans les diverses sous-disciplines du génie logiciel. Ce document n'a pas pour objet de déterminer la nature et l'importance des connaissances requises en la matière. Cependant, les sujets qui y sont abordés pourraient faciliter les futures discussions portant sur les connaissances pratiques et expérimentales requises dans certaines sous-disciplines particulières.

## 1.3 Objet

Préparé par un sous-comité du BCCAG, le présent document vise à faciliter le processus d'admission aux programmes de génie logiciel. Au Canada, les trois critères suivants doivent être respectés: niveau de scolarité minimal, expérience de travail appropriée et réussite de l'examen sur les méthodes professionnelles. Les membres du comité ont concentré principalement leurs efforts sur la détermination des normes de formation minimales auxquelles doit satisfaire un candidat pour obtenir l'autorisation d'exercer la profession d'ingénieur en logiciel. À cette fin, ils ont conclu que les programmes du BCCAG et du BCAPL constituaient deux des blocs de connaissance les plus concrets pour toute discipline du génie.

Les examens du BCCAG peuvent être considérés comme faisant partie intégrante d'un bloc de connaissances (BDC), car il devrait exister un lien direct entre les domaines de connaissance et les sujets d'examen. Cependant, sur le plan technique, les critères du BCAPL ne font référence à aucun BDC pour une discipline donnée; ils font simplement état des catégories générales des sujets à l'étude. On suppose que les candidats qui examinent le programme ont déjà une idée de ce que devrait être le BDC et considèrent que les cours offerts leur permettront d'acquérir les connaissances requises. Si on décide de codifier le BDC et de le rendre plus explicite, il faudrait également que le lien entre les domaines de connaissance et les sujets à l'étude soit évident. En conséquence, il serait utile que les éléments du BDC d'une discipline donnée qui correspondent parfaitement aux sujets d'examen d'un programme particulier soient présentés de façon à faciliter l'établissement de cette correspondance.

## 2.0 Travaux précédents

Plusieurs organismes ont entrepris des travaux dans ce domaine, avant même que le comité ne s'y intéresse. L'introduction d'un programme d'examens en génie logiciel au Programme d'examens du BCCAG avait déjà été proposée en 1998 [3]. Pour apaiser les préoccupations concer-

nant l'évolution rapide de cette discipline, certains ordres et associations ont apporté des modifications au programme national [4,5].

En 1998-1999, un comité formé de membres de Professional Engineers Ontario (PEO), appelé Engineering Disciplines Task Group (EDTG), a élaboré un guide destiné à évaluer les connaissances des diplômés du BCAPL qui sont passés au domaine du génie logiciel après l'obtention de leur diplôme [6]. Pour concilier les différences existant entre ses exigences et celles du programme original du BCCAG, la PEO a élaboré un nouveau programme [4]. William a rédigé un article [7] à ce sujet.

L'Ordre des ingénieurs du Québec (OIQ) [8] a rédigé un document exhaustif qui fait état des conclusions du groupe de travail spécial sur le génie logiciel. On y trouve des définitions et des renseignements détaillés concernant le génie logiciel, des aspects de la formation et plusieurs recommandations à l'intention de l'Ordre. Pour les besoins du présent rapport, le tableau 3 qui figure aux pages 15 et 16 du rapport de l'OIQ s'avère extrêmement intéressant: on y présente les sujets qui devraient être à l'étude dans le cadre d'un programme de génie logiciel, regroupés selon les catégories établies par le BCAPL.

Les membres du groupe de travail ont aussi examiné les cours des premiers programmes en génie logiciel offerts dans les universités canadiennes et accrédités par le BCAPL en 2001, soit ceux de l'Université McMaster [9], de l'Université d'Ottawa [10] et de l'Université Western Ontario [11]. Ces trois programmes sont offerts dans les facultés de génie en place, dont les programmes de génie sont déjà accrédités. Comme les étudiants inscrits dans ces premiers baccalauréats en génie logiciel finissaient en 2001, les programmes ont été examinés par des équipes du BCAPL à l'automne 2000 et ont été accrédités par ce dernier en juin 2001.

Le document intitulé «IEEE SWEBOOK» [12] constitue le dernier compte rendu d'un projet réalisé par l'entremise de la Société de l'informatique de l'IEEE et géré par l'Université du Québec à Montréal. Ce projet a été financé par plusieurs compagnies et organismes, dont le CCI. Il n'avait pas pour objet de réglementer l'exercice de la

profession d'ingénieur, mais plutôt d'examiner dix domaines de connaissances propres au processus de développement et de maintenance des logiciels, à savoir les exigences, la conception, la création, la mise à l'essai, la maintenance, la gestion de la configuration ainsi que les processus, les outils et la qualité du génie logiciel. Ces sujets ont tous été étudiés en détail dans le document, mais celui-ci ne traite pas de la grande question des domaines de connaissances requis aux fins de la réglementation de la profession. Il nous a toutefois été utile pour rédiger la section 3.3 intitulée «Développement de logiciels», présentée ci-après.

Les trois derniers éléments examinés [6,13,14] ont trait à l'évaluation de l'expérience des ingénieurs dans le domaine du génie logiciel.

## 3.0 Bloc de connaissances de base (BDCB) en génie logiciel

### 3.1 Mathématiques discrètes

L'analyse et la conception approfondies de tout logiciel nécessitent le recours à des outils mathématiques autres que le calcul et l'algèbre linéaire classiques, c'est-à-dire les «mathématiques discrètes» ou «structures discrètes». Ces matières font partie intégrante de nombreux programmes d'informatique, car elles constituent les bases mathématiques requises dans le domaine des structures de données et des algorithmes.



Les sujets d'étude sont les suivants: fonctions, relations, ensembles, arborescence, graphes, logique, algèbre booléenne, méthodes combinatoires, modèles à état, techniques de preuve et méthodes élémentaires de dénombrement.

### 3.2 Structures de données et algorithmes

Les structures de données et les algorithmes constituent les fondements de l'informatique et du développement de logiciels. Dans le cas du génie logiciel, il est essentiel de comprendre les divers algorithmes et structures de base ainsi que leurs caractéristiques. Ces sujets sont eux aussi à l'étude dans les programmes d'informatique et nécessitent la connaissance des mathématiques discrètes.

Les sujets d'étude sont les suivants: files d'attente, piles, listes, tas, arborescence, graphes, abstraction de données, tri, recherche, analyse, appariement de formes, techniques diviser-pour-régner, méthodes gloutonnes, complexité des algorithmes et critères de sélection.

### 3.3 Développement de logiciels

Ce domaine largement défini forme le «cœur» de la discipline unique en son genre qu'est le génie logiciel. Cette discipline consiste à mettre en application les fondements théoriques susmentionnés afin de produire des logiciels qui respectent les spécifications, dans un cadre de développement réel. Elle comprend notamment le cycle de vie du logiciel, soit les exigences, la conception, la création, la mise à l'essai et la maintenance. De plus, elle englobe la gestion de la configuration et le processus même du génie logiciel. Les concepts théoriques plus traditionnels ainsi que les aspects plus pratiques, axés sur l'expérience, font partie des connaissances à acquérir en développement de logiciels.

#### 3.3.1. Processus du génie logiciel

On appelle processus du génie logiciel l'approche systématique qui débute par l'étape de la conception (idée originale) et se termine par la production d'un logiciel fiable et maintenable sur le plan opérationnel. Le processus comprend l'aspect organisationnel du développement du logiciel, de la vérification de la qualité, de la mise à l'essai et de la gestion de la configuration. Le processus du génie logiciel est semblable à celui du génie des systèmes et comporte des méthodes et des outils applicables au génie logiciel.

Les sujets d'étude sont les suivants: concepts, infrastructure, mesure, définition, analyse qualitative, mise en œuvre et modifications du processus du génie logiciel.

#### 3.3.2. Exigences

Ce domaine d'étude concerne les activités de vérification réalisées à chaque étape subséquente, soit la conception, la création, la mise à l'essai, le fonctionnement, la maintenance, etc., qui visent à faire en sorte que les exigences soient toujours respectées. Il comprend également l'analyse de l'architecture du système auquel la conception est destinée ainsi que d'autres architectures dans lesquelles la conception doit s'insérer.

Les sujets d'étude sont les suivants: élicitation, analyse, exigences et spécifications, exigences fonctionnelles et non fonctionnelles, prototypage, techniques de spécification formelles et validation.

#### 3.3.3. Conception

Ce domaine d'étude concerne les activités associées à la conception du logiciel. Des éléments clés sont élaborés aux fins de la testabilité, de la maintenabilité et de la qualité.

Les sujets d'étude sont les suivants: architecture et structure du logiciel, analyse et conception orientées objet, conception au niveau du composant, modèles répartis, conception en vue de la réutilisation, analyse et évaluation de la qualité, notations et validation.

#### 3.3.4. Élaboration

Les sujets d'étude de ce domaine sont les suivants: réduction de la complexité, anticipation de la diversité, méthodes linguistiques, méthodes

formelles, méthodes visuelles, validation, évaluation et sélection du langage.

#### 3.3.5 Mise à l'essai

Les sujets d'étude de ce domaine sont les suivants: concepts, définitions, niveaux (exigences, conception, création, intégration, fonctionnement), techniques et mesures.

#### 3.3.6 Maintenance

Ce domaine d'étude comprend tout ce qui concerne la maintenance du logiciel, c'est-à-dire des activités qui sont essentielles pour s'assurer que celui-ci continue de satisfaire aux exigences fonctionnelles et qualitatives dans un cadre opérationnel qui évolue.

Les sujets d'étude sont les suivants: documentation, exigences et spécifications, exigences fonctionnelles et non fonctionnelles, prototypage, techniques de spécifications formelles, conception structurée, analyse et conception orientées objet, conception au niveau du composant, modèles répartis, conception en vue de la réutilisation, maintenance, outils et environnements.

#### 3.3.7 Gestion de la configuration

Ce domaine d'étude comprend les sujets suivants: gestion du processus de configuration du logiciel, identification de la configuration, contrôle de la configuration, rapport sur l'état, vérification, gestion de la diffusion et livraison du logiciel.

### 3.4 Fiabilité et sécurité du système

Ce domaine d'étude porte sur la fiabilité et la sécurité de tout le système. Bien que la qualité et la fiabilité aient leur importance à toutes les étapes du cycle de vie de développement du logiciel, la fiabilité et la sécurité de l'ensemble du système, sur le plan matériel ou autre, est primordiale. Cet aspect a une incidence directe sur l'obligation qu'ont les ingénieurs d'assurer la sécurité et le bien-être du public en concevant des logiciels vérifiables et fiables, où une défaillance pourrait compromettre la sécurité. Nombre des sujets d'étude ont été mentionnés dans les paragraphes précédents, mais ceux-ci sont maintenant présentés séparément afin de faire ressortir l'obligation qu'ont les ingénieurs de produire des systèmes fiables et sécuritaires.

Les sujets d'étude sont les suivants : méthodes et spécifications formelles, préconditions et postconditions, assertions, vérification formelle, exigences approximatives et changeantes, vérification et validation, élaboration de plans d'essai et génération de cas d'essai, techniques d'essai de la boîte noire et de la boîte blanche, contraintes financières et contraintes en temps réel, redondance et tolérance aux pannes, modes de défaillance, méthodes d'analyse probabiliste et modélisation des mesures de la qualité.

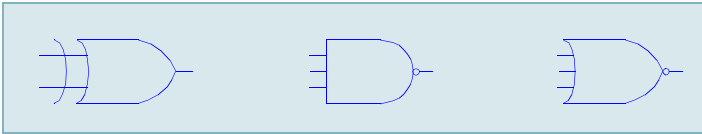
### 3.5 Systèmes numériques

Les ingénieurs en logiciel doivent connaître les systèmes numériques pour comprendre que les ordinateurs ne sont pas de simples «boîtes noires» destinées à faire fonctionner des logiciels. Il s'agit d'un domaine d'étude obligatoire également en génie électrique et en génie informatique (p. ex. le cours intitulé «98-Comp-A2»), ayant ses bases dans l'étude de la logique qui fait partie du domaine des mathématiques discrètes.

Les sujets d'étude sont les suivants: conception de circuits combinatoires et séquentiels, mise en œuvre à l'aide de passerelles, composants du système logique, matrices de portes et dispositifs logiques programmables, langages de définition évolués, conception de petits systèmes, flux de données, signaux, relation temporelle, organisation et interfaçage de microprocesseurs.

### 3.6 Architecture d'ordinateur

Ce domaine repose sur la connaissance des systèmes numériques décrits précédemment et fait lui aussi partie des programmes de génie informatique (p. ex. le cours intitulé «98-Comp-A3» du programme d'examen



du BCCAG), mais la matière n'y est pas examinée aussi en profondeur (simplement au niveau de l'abstraction logique et non pas au niveau du dispositif). Dans le contexte de la science informatique, l'aspect matériel de l'architecture d'ordinateur n'est habituellement pas étudié aussi en profondeur que le permet la connaissance préalable des systèmes numériques des étudiants en génie logiciel.

Les sujets d'étude sont les suivants: structure de l'ordinateur et architecture du microprocesseur, UCT et organisation de la mémoire, bus, interfaçage, E/S parallèle et en série, unités de stockage, jeu d'instructions, modes d'adressage, registres, interruptions et E/S, processeurs spécialisés et systèmes intégrés.

### 3.7 Systèmes d'exploitation

Comme la plupart des logiciels fonctionnent sous la surveillance et le contrôle des ressources des systèmes d'exploitation, il est essentiel de comprendre les caractéristiques et les limites de ces systèmes ainsi que leurs répercussions sur l'exécution des programmes, en particulier dans le cas des systèmes de commande et des systèmes en temps réel. Certains aspects des systèmes d'exploitation sont également étudiés en génie informatique (p. ex. le cours «98-Comp- A5» du programme d'examens du BCCAG) et science informatique. D'ordinaire, le génie informatique traite de l'utilisation et des caractéristiques des systèmes d'exploitation, tandis que la science informatique s'intéresse davantage à leur conception. La connaissance de certains concepts en organisation du matériel (p. ex. les ressources telles que la mémoire) est requise.

Les sujets d'étude sont les suivants: communication interprocessus, synchronisation, ordonnancement, affectation des ressources, gestion de la mémoire, fonctionnement multitâche et multiprocesseur, performance et mesure, exigences relatives au soutien en temps réel, impasse et caractéristiques des systèmes d'exploitation modernes.

### 3.8 Fichiers et bases de données

La plupart des systèmes logiciels utilisent des fichiers et des bases de données. Par ailleurs, le traitement d'importants volumes de données peut s'avérer nécessaire dans le cas de systèmes essentiels pour la sécurité. Ce domaine d'étude fait souvent partie intégrante des programmes de science informatique de base.

Les sujets d'étude sont les suivants: modèles de données, rapports entre entités, dispositifs de stockage de masse, structure des fichiers, types de base de données (relationnelle, hiérarchique, en réseau), organisation des fichiers (séquentiels, indexés, à accès direct, dispersés), méthodes et langages d'interrogation, sécurité et intégrité, traitement transactionnel.

### 3.9 Systèmes et commandes

Ce domaine d'étude traite des principes fondamentaux liés à l'analyse de la stabilité des systèmes de commande en boucle fermée. Comme le besoin de réglementation en cette matière découle directement de la nécessité d'assurer la protection du public, l'analyse de la stabilité est une étape essentielle de la conception de tout système de contrôle informatique servant d'interface avec le monde réel. Il s'agit d'une matière obligatoire dans la plupart des disciplines relatives au génie électrique et au génie informatique.

Les sujets d'étude sont les suivants: modèles, fonctions de transfert et réponse du système, analyse et conception du lieu des racines, réaction et stabilité, diagrammes de Bode, critère de Nyquist, conception de domaine fréquentiel, représentation des variables d'état, régulateur PID, contrôle numérique, transformée Z, interfaçage et algorithmes de commande.

### 4.0 Correspondance entre le programme d'étude et le programme d'examen du BCAP

Le tableau 1 présente les domaines de connaissance ou d'étude communs à la plupart des programmes. Les numéros d'examen qui figurent dans les colonnes intitulées BCCAG (programme original de 1998 du BCAP), PEO (programme révisé de l'Ontario) et C.-B. (programme révisé de la Colombie-Britannique) correspondent aux domaines d'étude généraux présentés dans la colonne de gauche. Les numéros indiqués dans les colonnes intitulées MAC (Université McMaster), Udo

Tableau 1: Aperçu des sujets fondamentaux en génie logiciel

Domaine général	BCCAG[3]	PEO[4]	C.-B. [5]	OIQ[8]	MAC[9]	Udo[10]	UWO[11]
Mathématiques discrètes	-	BS17	-	Oui	2E03	MAT2343	SE251
Structures de données et algorithmes	A1, A4	A4	BS(A1)	Oui	2C04	CSI2114, CSI3105	CS027, CS210, CS340
Développement de logiciels	A5	A4	A3	Oui	2A03, 2B04	SEG2100, SEG2101, SEG3100	SE351, SE452, SE453
Mise à l'essai, fiabilité	-	A7	-	Oui	2F03	SEG4111	SE453
Systèmes numériques	A2	A2	BS(A2)	Oui	2D04	ELG1100	ECE339
Architecture d'ordinateur	A2	A3	BS(A2)	Oui	3G03, 3F03	ELG2181, CEG3391	ECE375
Systèmes d'exploitation	A3	A5	A1	Oui	3B04	CSI 3310	SE201, CS305
Fichiers et bases de données	A6	-	A4	Oui	3H03	CSI 3317	-
Systèmes et commandes	-	A1	-	-	3L03, 4A03	-	-

(Université d'Ottawa) et UWO (Université Western Ontario) sont ceux des cours offerts dans le cadre des trois programmes accrédités. Les «Oui» indiqués dans la colonne OIQ (Ordre des ingénieurs du Québec) signifient simplement que les cours en question figurent dans le tableau 3 de cet organisme. Comme le document SWEBOK ne traite qu'une partie de ces sujets dans le domaine du développement de logiciels, bien qu'il soit très détaillé, il n'a pas été inclus dans le tableau. Des renvois aux examens ou aux cours particuliers qui traitent des sujets généraux sont indiqués.

De façon générale, il semble y avoir un important consensus entre les diverses sources en ce qui concerne les domaines de connaissance requis en matière de génie logiciel. Comme c'est un domaine qui évolue, on s'attend toutefois à ce que certaines différences se manifestent. Le programme du BCCAG ne semble pas exiger l'apprentissage des mathématiques discrètes ainsi que de certains éléments liés à la mise à l'essai et à la fiabilité des logiciels. La PEO et l'Université McMaster, quant à elles, mettent obligatoirement au programme des cours sur les systèmes et les commandes à l'instar d'autres établissements d'enseignement qui désirent faire accréditer leurs programmes en génie logiciel, comme l'Université de Waterloo.

### 5.0 Bloc de connaissances complémentaires (BDCC) en génie logiciel

On présume que les ingénieurs devront acquérir d'autres connaissances en plus de leurs connaissances de base, c'est-à-dire un BDCC. Ils devront évidemment maîtriser les sujets du BDCB avant d'aborder ceux du BDCC. Certains domaines d'étude pourront être ajoutés ou supprimés au fil de l'évolution de la technologie et des techniques du génie.

Le BDCC contribue à l'enrichissement du bloc de connaissances de plusieurs façons.

#### 5.1 Approfondissent des sujets fondamentaux

Les ingénieurs peuvent approfondir davantage certaines domaines d'études du BDCB, tout comme s'ils suivaient des cours supérieurs leur permettant d'approfondir la matière étudiée dans les cours de base.

#### 5.2 Approfondissement des domaines de spécialisation

Il s'agit de domaines d'étude en génie logiciel qui viennent s'ajouter au BDCB. Ils permettent aux praticiens de mettre en application des compétences et des connaissances avancées dans des domaines du génie logiciel qui vont plus loin que les cours de base.

Voici quelques exemples:

- Conception de produits logiciels
- Évaluation et amélioration du processus logiciel
- Outils d'estimation et de planification de logiciels
- Conception simultanée de logiciels
- Conception de systèmes en temps réel
- Conception d'interfaces homme-machine
- Conception de systèmes d'essai évolués

#### 5.3 Élargissement des domaines d'application

Les ingénieurs en logiciel peuvent mettre leurs compétences en pratique dans des domaines d'application en dehors de leur discipline particulière. De fait, cette interrelation avec d'autres disciplines du génie revêt une importance particulière pour l'ingénieur en logiciel, de sorte que ce dernier doit posséder une bonne connaissance pratique de ces domaines d'application. Comme ceux-ci sont généralement liés à d'autres disciplines du génie, le praticien doit avoir un vaste éventail de connaissances dans des domaines d'application particuliers.

Voici quelques exemples:

- Contrôle des procédés industriels
- Logiciel de télécommunications (commutation)
- Logiciel d'avionique

- Logiciel pour réseau (protocole) et systèmes répartis
- Composants et systèmes médicaux

## 6.0 Sommaire et conclusions

Il n'y a pas longtemps que le génie logiciel est reconnu étant une discipline distincte, encadrée par des règles professionnelles et des règles d'octroi de permis. Les fondements de cette discipline remontent toutefois à beaucoup plus loin. Un consensus général a donc été établi à l'égard de ce qui constitue le bloc de connaissances de base requis dans ce domaine. Les sujets d'étude comprennent non seulement ceux qui portent sur le développement immédiat des logiciels et leur cycle de vie, mais aussi des sujets comme les systèmes numériques, l'architecture d'ordinateur et les systèmes de commande. Voilà qui reflète bien la philosophie du génie selon laquelle un ingénieur en logiciel doit posséder des connaissances dans tous les domaines liés aux systèmes et aux environnements dans lesquels fonctionnent les logiciels. Conjugée aux sciences et aux mathématiques fondamentales requises, cette exigence élargie permettra aux ingénieurs en logiciel d'acquérir une formation de base garantissant que les produits et processus logiciels qu'ils mettent au point respectent leur obligation professionnelle de protéger la sécurité du public.

Les divers participants aux travaux du comité s'entendent tous à propos des sujets les plus importants. Même le projet SWEBOK de IEEE, bien qu'il ne vise pas expressément à déterminer les exigences liées à l'octroi des permis d'ingénieur au Canada, a contribué à la détermination des sujets d'études relatifs au développement de logiciels. Le bloc de connaissances de base devra évidemment être adapté aux changements dont fera l'objet la discipline. Il faudra également porter une attention particulière aux divers intervenants en jeu et à l'apport de leurs organismes respectifs.

Le bloc de connaissances complémentaires, quant à lui, n'est pas encore bien défini. De façon générale, un ingénieur en logiciel devrait posséder des connaissances autres que celles de sa discipline particulière, c'est-à-dire des connaissances approfondies sur un ou plusieurs sujets fondamentaux, dans au moins un domaine de spécialisation, ainsi qu'une connaissance élargie des domaines d'application. Il existe déjà plusieurs ingénieurs en logiciel qui possèdent des connaissances spécialisées ou qui exercent dans certains domaines d'application. Ce sujet n'a toutefois pas encore été abordé. Il devra être examiné ultérieurement, et tous les lecteurs du présent document sont invités à donner leur avis sur ce sujet. Comme dans le cas des sujets fondamentaux, les sujets complémentaires évolueront eux aussi au rythme de l'évolution de la discipline, de sorte que des modifications et des mises à jour à cet égard devront également être apportées au présent document.

La gestion du développement de logiciels et son cycle de vie complet n'a pas été explicitement inclus dans le présent rapport. Il est entendu que les techniques de gestion de projet mises en pratique dans toutes des disciplines du génie peuvent être appliquées au génie logiciel. Il est également entendu qu'il peut exister certaines situations propres au génie logiciel nécessitant l'adoption de méthodes différentes.

## 7.0 Remerciements

Les auteurs aimeraient remercier le personnel du Conseil canadien des ingénieurs de leur précieux concours, en particulier David Lapp, Marie Carter et Begonia Lojk. Ils désirent également remercier Phil Sunderland, directeur du projet pilote visant à établir un bloc de connaissances, de son soutien et de sa patience.

## 8.0 Références

- [1]. Conseil canadien des ingénieurs. *Core and Supplementary Bodies of Knowledge for Software Engineering*, Ottawa, Ontario, février 2002.
- [2]. Conseil canadien des ingénieurs. *Guide sur l'exercice de l'ingénierie au Canada*, Ottawa, Ontario, 2001.
- [3]. Conseil canadien des ingénieurs. *CCI 1998 Software Engineering*

Syllabus, Ottawa, Ontario, 1998.

- [4]. Professional Engineers Ontario. *PEO Revised Software Engineering Syllabus*, Toronto, Ontario, 1999.
- [5]. Association of Professional Engineers and Geoscientists of British Columbia. *APEGBC Revised Software Engineering Syllabus*, Burnaby, C.-B., 1999.
- [6]. Professional Engineers Ontario. *PEO EDTG Experience Requirements for Cross-Discipline Applicants Practicing in the Software Engineering Field*, Toronto, Ontario, 25 mars 1999.
- [7]. Williams, N. *Professional Engineers Ontario's Approach to Licensing Software Engineering Practitioners*, comptes rendus de la 14<sup>e</sup> Conference on Software Engineering Education and Training, (Charlotte, NC), pp. 77 et 78, février 2001.
- [8]. Ordre des ingénieurs du Québec. *Software Engineering Definitive Report*, Montréal, Québec, 17 novembre 2000.
- [9]. Université McMaster. Department of Computing and Software, *McMaster University Software Engineering Curriculum*, Hamilton, Ontario, (accrédité par le BCAP en 2001).
- [10]. Université d'Ottawa. School of Information Technology and Engineering, *University of Ottawa Software Engineering Curriculum*, Ottawa, Ontario, (accrédité par le BCAP en 2001).
- [11]. Université de Western Ontario. Department of Electrical and Computer Engineering, *University of Western Ontario Software Engineering Curriculum*, London, Ontario, (accrédité par le BCAP en 2001).
- [12]. IEEE. *IEEE SWEBOOK Stone Man Version (0.7)*, avril 2000.
- [13]. Conseil canadien des ingénieurs. *CEQB Interview and Assessment Guide for the Evaluation of Software Engineering Experience*, Ottawa, Ontario, 12 mars 2001.
- [14]. Conseil canadien des ingénieurs. *CEQB Guide for the Presentation and Evaluation of Software Engineering Experience*, Ottawa, Ontario, 13 juin 2001.

## À propos de l'auteurs

**Bob Dony** est titulaire d'un baccalauréat et d'une maîtrise en génie de la conception des systèmes qu'il a obtenus à l'Université de Waterloo en 1986 et en 1988 respectivement. Il a obtenu son doctorat en génie électrique et informatique à l'Université McMaster, en 1995. Depuis 1997, il est professeur adjoint dans le cadre du programme de systèmes et de calcul d'ingénierie de l'école d'ingénierie de l'Université de Guelph. Il est membre du Bureau canadien des conditions d'admission en génie (BCCAG) et président du sous-comité de génie logiciel chargé du projet pilote visant à déterminer un bloc de connaissances pour le BCCAG. Il est membre de Professional Engineers Ontario et membre du comité de vérification des diplômes, du groupe de travail sur l'évolution des admissions en génie ainsi que du groupe de travail sur les nouvelles disciplines de Professional Engineers Ontario (PEO). Il a par ailleurs coprésidé la conférence canadienne sur le génie électrique et informatique de l'IEEE en 2001 et fait partie du comité permanent de la conférence canadienne sur la formation en génie informatique et logiciel.



**Pieter Botman** est ingénieur agréé en Colombie-Britannique. Fort de plus de vingt ans d'expérience en génie logiciel et en gestion, il aide les entreprises en services informatiques à évaluer et à améliorer leur processus logiciel, et à gérer les projets, la qualité et les produits logiciels. M. Botman est membre de la Société de l'informatique de l'IEEE, de l'Association for Computing Machinery et de la division de la qualité logicielle de l'ASQ. On peut le joindre à [p.botman@ieee.org](mailto:p.botman@ieee.org)



**Bill Briggs** a obtenu un baccalauréat spécialisé en chimie et un certificat en génie de l'Université Mount Allison, puis a poursuivi des études de deuxième cycle en génie électrique à l'Université du Nouveau-Brunswick. Il a ensuite travaillé pour Énergie NB, d'abord au sein du groupe des opérations nucléaires, puis à la distribution. Il a occupé un poste de scientifique en recherche et développement chez Fiberglas Canada Inc. pendant cinq ans. Depuis dix ans, il travaille à son compte à titre d'expert-conseil dans divers domaines de haute technologie. Il est retourné depuis peu dans le monde universitaire en qualité de chargé de cours supérieur pour le département de génie électrique et informatique de l'Université du Nouveau-Brunswick.



**Ross Haggart** a fait ses études à l'Université Carleton, où il a obtenu en 1980 un baccalauréat en génie électrique avec spécialisation en systèmes informatiques. Il a consacré la première moitié de sa carrière à la conception, au développement et au déploiement de systèmes informatiques pour des applications militaires et industrielles. Il a par la suite travaillé au sein de l'industrie des services conseils dans le domaine des services administratifs où il a participé principalement aux activités de gestion de projets et de programmes. M. Haggart occupe actuellement le poste de vice-président de l'ingénierie chez Cerkits Corporation, société canadienne qui développe des robots de sécurité pour des environnements civils et militaires. Il est membre de Professional Engineers Ontario et fait partie du Comité de vérification des diplômes (ERC) de l'organisme ainsi que du groupe de travail sur les admissions en génie.



**Paul Taylor** est titulaire d'un baccalauréat et d'un doctorat de l'Université de Wales. Il est directeur du département de génie informatique et logiciel de l'Université McMaster, à Hamilton, où il enseigne également. Ses recherches portent sur l'application du contrôle informatique aux processus chimiques. Elles comportent trois domaines d'expertise, soit la théorie de commande, la dynamique et la modélisation des processus ainsi que les applications en temps réel.

