

## Diplôme d'études supérieures spécialisées en génie logiciel

**Téléphone :** 514 987-0437  
**Courriel :** cyclessupgenielogiciel@uqam.ca

Code	Titre	Grade	Crédits
1622	Diplôme d'études supérieures spécialisées en génie logiciel	Diplôme d'études supérieures spécialisées, DESS	30

<b>Contingent</b>	Programme non contingenté
<b>Régime et durée des études</b>	Temps complet : trois trimestres Temps partiel : neuf trimestres ou moins
<b>Campus</b>	Campus de Montréal

### OBJECTIFS

Le programme de DESS en génie logiciel a pour objectif général de former et de spécialiser des personnes en génie logiciel. Par l'intermédiaire de cours et d'un projet ou d'un stage professionnel en entreprise, les étudiants pourront acquérir des connaissances théoriques et pratiques dans les domaines de pointe du génie logiciel. Les diplômés du programme pourront agir comme agents de changement et d'amélioration dans le milieu professionnel du génie logiciel. Les diplômés du programme pourront poursuivre leur formation au sein de la maîtrise en génie logiciel.

Les objectifs spécifiques du DESS en génie logiciel sont :

- L'approfondissement de connaissances : l'étudiant pourra acquérir et approfondir les connaissances essentielles sur les différents aspects du processus de développement de logiciel, plus particulièrement les exigences, la conception, la réalisation, la maintenance et l'assurance qualité de logiciels.
- La spécialisation : l'étudiant pourra acquérir des connaissances de pointe, entre autres, en développement d'applications mobiles, en base de données, en processus Agile, en processus de tests, en sécurité des systèmes et en mesures.
- L'intégration : l'étudiant sera en mesure d'intégrer en pratique les connaissances acquises par la réalisation d'un projet technique ou d'un stage professionnel en entreprise.
- Le développement d'habiletés personnelles : l'étudiant développera des habiletés d'acquisition d'information, d'analyse, de réflexion et de synthèse sur des situations concrètes dans une conduite d'intégrité et d'éthique; il développera aussi des habiletés au travail d'équipe.

### PARTENARIAT

Ce programme est offert en partenariat avec l'École de technologie supérieure.

### CONDITIONS D'ADMISSION

Le candidat doit :

être titulaire d'un baccalauréat, ou l'équivalent, en informatique, en génie dans un domaine approprié, ou en sciences appliquées avec une composante en informatique, obtenu avec une moyenne cumulative d'au moins 3,0 sur 4,3 ou l'équivalent;

OU être titulaire d'un baccalauréat ou l'équivalent, dans un autre

domaine, obtenu avec une moyenne d'au moins 3,0 sur 4,3 et posséder les connaissances ou l'expérience jugées suffisantes en informatique, en développement de logiciels ou en technologie de l'information;

Tout dossier de candidature avec une moyenne inférieure à 3,0 mais supérieure à 2,7 sur 4,3 sera étudié par le sous-comité d'admission et d'évaluation du programme et pourrait, dans certains cas, faire l'objet d'une recommandation d'admission;

OU posséder les connaissances nécessaires, une formation appropriée et une expérience jugée pertinente d'au moins de cinq ans.

Les dossiers de candidats détenteurs d'un tel baccalauréat obtenu avec une moyenne inférieure à 2,7 sur 4,3, mais égale ou supérieure à 2,5 sur 4,3 (ou l'équivalent) seront étudiés par le sous-comité d'admission et d'évaluation, à la condition de posséder une formation additionnelle et appropriée d'au moins 15 crédits universitaires (ou l'équivalent) complétés avec une moyenne cumulative d'au moins 3,2 sur 4,3 (ou l'équivalent).

Le candidat dont la préparation n'est pas jugée suffisante pourrait se voir imposer des cours d'appoint ou une propédeutique.

#### Connaissance du français

Le candidat doit posséder une connaissance suffisante de la langue française orale et écrite.

Sur étude de dossier par le sous-comité d'admission et d'évaluation, le candidat pourrait se voir imposer de passer un test de classement en français.

S'il n'a pas acquis le seuil établi par le programme, le candidat devra réussir une formation en français dès sa première inscription au programme. Un ou des cours pourront être suivis parallèlement à la scolarité régulière du programme selon les possibilités. L'étudiant devra avoir démontré une maîtrise suffisante du français avant la fin de sa première année d'inscription au programme.

#### Connaissance de l'anglais

Le candidat doit posséder la capacité de lire des textes scientifiques et techniques rédigés en anglais.

#### Méthode et critères de sélection

Évaluation du dossier académique, de la lettre de motivation du candidat et des lettres de recommandation.

Les candidats présentant une demande sur la base de l'expérience pertinente doivent détailler leurs expériences en précisant les rôles et responsabilités exacts qu'ils ont cumulés. Un curriculum vitae complet

doit accompagner la demande d'admission. Les candidats peuvent être convoqués à une entrevue.

### Procédure

Le formulaire de demande d'admission complété doit être transmis au registrariat, accompagné d'une lettre de motivation précisant les objectifs du candidat face au programme, de ses relevés de notes officiels pour toutes les études de niveau universitaire (sauf celles réalisées à l'UQAM), d'un curriculum vitae et d'attestations d'expérience s'il y a lieu.

### Régime et durée des études

Temps complet : trois trimestres (un an)

Temps partiel : neuf trimestres ou moins (trois ans ou moins)

## COURS À SUIVRE

(Sauf indication contraire, les cours comportent 3 crédits. Certains cours ont des préalables. Consultez la description des cours pour les connaître.)

Le DESS en génie logiciel est composé de d'un bloc de cinq cours obligatoires (15 crédits), de trois cours de spécialisation (9 crédits) et d'une activité de synthèse (6 crédits), pour un total de 30 crédits.

### Bloc obligatoire (15 crédits)

#### Les cinq cours obligatoires suivants (15 crédits) :

- MGL7315 Gestion de projet en génie logiciel
- MGL7260 Exigences et spécifications de systèmes logiciels
- MGL7361 Principes et applications de la conception de logiciels
- MGL7460 Réalisation et maintenance de logiciels
- MGL7560 Vérification et assurance qualité de logiciels

#### Cours de spécialisation (9 crédits)

#### Les trois cours de spécialisation suivants (9 crédits) :

- 2 cours de spécialisation choisis dans la liste A, parmi les cours au choix de l'UQAM en informatique et génie logiciel (6 crédits)

- 1 cours choisi parmi les cours de spécialisation offerts par l'ÉTS, voir Liste B (3 crédits). Le directeur du programme doit approuver au préalable le choix du cours à suivre à l'ÉTS.

### Activités de synthèse (6 crédits)

#### L'activité de synthèse suivante :

- MGL8705 Stages industriels et rapport technique
- MGL8706 Rapport technique II

ou

MGL8707 Projet technique (6 cr.)

Liste A, cours au choix offerts à l'UQAM :

- INF7210 Nouvelles perspectives en bases de données
- INF7235 Programmation parallèle haute performance
- INF7270 Écosystème du logiciel libre
- INF8750 Sécurité des systèmes informatiques
- MGL7130 Développement d'applications mobiles
- MGL7230 Tests logiciels
- MGL7240 Mesures et développement de logiciel
- MGL7250 Processus de développement Agile
- MGL7760 Qualité et productivité des outils logiciels
- MGL7810 Sujets spéciaux en génie logiciel I
- MGL7811 Sujets spéciaux en génie logiciel II
- MGL7815 Lecture dirigée I

L'étudiant peut remplacer un (1) cours au choix par un autre, choisi parmi les cours de 2e cycle offerts par le Département d'informatique de l'UQAM.

Liste B, cours offerts par l'ÉTS :

- MGL810 Programmation temps réel sur des architectures parallèles
- MGL825 Télématique et réseaux
- MGL835 Interaction humain-machine

- MGL841 La mesure : concept clé en ingénierie du logiciel
- MGL842 L'ingénierie de la qualité du logiciel
- MGL843 Sujets avancés en conception logicielle
- MGL844 Architecture logicielle
- MGL845 Ingénierie dirigée par les modèles
- MGL846 Concepts et pratique des tests logiciels
- MGL847 Amélioration des processus logiciels : intervention dans une organisation
- MGL848 Validation et vérification de modèles en génie logiciel
- MGL849 Modélisation, analyse et programmation des systèmes temps réel
- MGR850 Sécurité de l'Internet
- MTI825 Gestion des services TI
- SYS869 Sujets spéciaux I : génie logiciel
- SYS870 Sujets spéciaux II : génie logiciel

Remarque : Le candidat voudra bien prendre note que les activités au choix dans ce programme et énumérées ci-dessus ne peuvent pas être offertes à chacun des trimestres (automne, hiver ou été). Par conséquent, elles sont réparties sur plusieurs trimestres et sont donc offertes en alternance d'un trimestre, voire d'une année à l'autre. De plus, la majorité des cours sont offerts en soirée.

## RÈGLEMENTS PÉDAGOGIQUES PARTICULIERS

Pour s'inscrire au projet technique l'étudiant doit avoir complété les cours du bloc obligatoire du programme.

Pour s'inscrire au stage, l'étudiant doit avoir complété les cours obligatoires, avoir une moyenne d'au moins 3,2 sur 4,3 et être citoyen canadien, résident permanent ou détenteur d'un permis de travail valide au Canada. L'étudiant doit aviser la direction du programme de son intention d'effectuer un stage au moins quatre mois avant le trimestre visé pour la réalisation du stage. L'UQAM ne s'engage pas à trouver un lieu de stage à l'étudiant, mais favorise les contacts avec des employeurs potentiels par l'entremise du service des stages du Département d'informatique.

## PASSERELLES

Un étudiant désirant continuer sa formation à la maîtrise en génie logiciel se verra reconnaître à la maîtrise tous les cours réussis dans le cadre du DESS en génie logiciel.

Un étudiant qui abandonne le DESS en génie logiciel pourra recevoir l'attestation de diplomation du programme court de 2e cycle en génie logiciel s'il a rencontré toutes les exigences de ce programme court.

## DESCRIPTION DES COURS

### INF7210 Nouvelles perspectives en bases de données

Concepts avancés des bases de données. Gestion de transactions. Contrôle et optimisation des performances. Bases de données parallèles et réparties. Développement d'applications de bases de données Web et multi-tiers. Bases de données objet et objet-relationnel. Gestion de données semi-structurées et multimédia. Entrepôts de données et analyse de données (OLAP). Fouille de données (data mining). Bases de données déductives. Repérage de l'information.

### INF7235 Programmation parallèle haute performance

Modèles d'architectures à haute performance. Paradigmes de programmation parallèle et stratégies de conception de programmes parallèles. Métriques de performances et principales sources des surcoûts. Langages et bibliothèques de programmation parallèle. Problèmes typiques en programmation scientifique haute performance: calculs de grilles, de particules, de matrices.

### INF7270 Écosystème du logiciel libre

Objectifs

Ce cours vise à présenter aux étudiants la problématique du logiciel libre dans les organisations. Au terme du cours, les étudiants devraient

être capables d'évaluer le potentiel des logiciels libres dans la stratégie des TI ; d'en évaluer les avantages et les inconvénients ; d'identifier les applications où le logiciel libre peut représenter des bénéfices importants ; de comprendre la dynamique des communautés de développeurs de logiciel libre ; de concevoir des approches pour effectuer une veille technologique du secteur ; de s'initier à la mise sur pied d'un projet de logiciel libre ; de saisir les principaux enjeux relatifs aux licences libres ; d'établir une stratégie adaptée à chaque organisation relativement au logiciel libre y compris la sensibilisation de la haute direction à ce mouvement.

#### Sommaire du contenu

Définition du logiciel libre selon la Free Software Foundation (FSF) et l'Open Software Initiative (OSI) ; identification des principales licences libres ; problématiques particulières des logiciels applicatifs ; principaux domaines où le logiciel libre fait des percées importantes ; études de cas (Apache, Mozilla, Eclipse) ; les forges et leur fonctionnement ; le logiciel libre comme modèle d'affaires : Alfresco, MongoDB, etc. ; la dynamique des projets libres ; l'étude de faisabilité d'un projet libre ; problématique particulière des logiciels applicatifs ; motivation des participants à des projets libres : revue des principales études sur ce sujet (ex. FLOSS de la Commission européenne) ; le logiciel libre comme outil de développement économique (ex. rapport de l'OCDE) ; élaboration d'un plan stratégique incluant le libre et plan de communication destiné à la direction ; enfin principaux inconvénients du libre.

#### Préalables académiques

INF7115 Bases de données ou INF7215 Analyse et conception des systèmes d'information de l'entreprise ou l'équivalent

#### INF8750 Sécurité des systèmes informatiques

Principes et concepts fondamentaux de la sécurité des systèmes informatiques. Principaux services: confidentialité, intégrité, disponibilité, authentification, non répudiation, contrôle d'accès. Typologie des attaques: fuites, modifications d'information, privations de service. Mécanismes sécuritaires modernes: systèmes de chiffrement symétriques et asymétriques; fonctions de hachage; génération pseudo-aléatoire. Protocoles sécuritaires: authentification, signature, échange et gestion de clés. Sécurité des systèmes centralisés et des systèmes répartis: politiques et modèles de sécurité; contrôle d'accès; rôles et privilèges. Sécurité des programmes: virus, chevaux de Troie. Contre-mesures: journalisation, audits; détection d'intrusion; filtrage; mécanismes de recouvrement. Analyse de risque. Éducation des usagers. Considérations légales, politiques et éthiques.

#### MGL7130 Développement d'applications mobiles

##### Objectifs

À l'issue de ce cours, l'étudiant sera en mesure de développer des applications mobiles dans des environnements modernes.

##### Sommaire du contenu

Architecture et fonctionnalités de base des plate-formes mobiles; environnement de développement des plate-formes mobiles; structure et composants fondamentaux des applications mobiles ; construction de l'interface utilisateur ; utilisation des ressources : XML, images, fichiers, etc. ; persistance des données ; intégration ; signature d'une application ; déploiement et contraintes particulières des applications mobiles. API pour le développement des applications mobiles dépendantes de leur contexte. Interactions avec les outils de captage de données à travers les appareils et les réseaux mobile. Développement d'applications de commerce électronique mobile. Tendances et perspectives futures.

##### Conditions d'accès

Connaissance d'un langage de programmation orienté-objet des technologies du Web.

#### MGL7230 Tests logiciels

##### Objectifs

À la fin du cours, l'étudiant sera en mesure de réaliser la planification, la conception et le suivi de toutes les activités de tests d'un projet; de

développer et accomplir efficacement les différentes phases de tests;

##### Sommaire du contenu

Tests logiciels dans le cycle de vie du logiciel. Normes relatives aux tests. Processus des tests logiciels. Les différents niveaux de tests. Conception des cas de tests. Efficacité des tests. Stratégies de tests pour les environnements modernes : infonuagique, applications Web, applications mobiles, applications multi-couches. Tests et méthodologies agiles. Principaux cadres pour les tests. Automatisation des tests.

#### MGL7240 Mesures et développement de logiciel

##### Objectifs

Ce cours vise à développer les connaissances et habiletés de l'étudiant à définir, à déployer et à faire évoluer un plan de mesure afin de soutenir les processus de décision et de gestion du logiciel (estimation, planification, suivi, déploiement, exploitation et mise au rancart). À la fin du cours, l'étudiant sera en mesure de décrire les concepts de base de la mesure et d'élaborer et de déployer les divers éléments d'un plan de mesure adapté au contexte d'une entreprise afin de répondre aux besoins en information du personnel technique et des gestionnaires. Il connaîtra aussi les principales normes applicables.

##### Sommaire du contenu

Concepts de base de la mesure. Normes, modèles et méthodes liés aux principes, aux processus et aux pratiques de la mesure en génie logiciel. Processus type de la mesure. Conception et application d'un plan de mesure. Présentation des résultats pour faciliter la prise de décision. Mesures liées aux projets, aux processus, aux applications et à la gouvernance des TI. Mesure de la taille fonctionnelle des logiciels et ses usages: productivité des processus de développement et de maintenance, estimation de projet, étalonnage (benchmarking), gouvernance du portfolio applicatif et amélioration de la qualité et de la productivité du processus. Aspects économiques de la mesure. Rôle de la mesure dans les différents modèles d'affaires de sous-traitance du développement logiciel. Défis organisationnels liés à la mesure. Outils de mesure.

#### MGL7250 Processus de développement Agile

##### Objectifs

Ce cours vise à développer les connaissances et habiletés de l'étudiant à comprendre et appliquer les différents principes et méthodes dites Agile. À la fin du cours, l'étudiant sera apte à mettre en place un processus de développement agile au sein d'un groupe de développement logiciel

##### Sommaire du contenu

Valeurs et principes de l'Agilité. Méthodes : Scrum, TDD, XP, Crystal, DDD, DSDM. Démarrage d'un projet Agile. Architecture et livraison incrémentale. Gestion de projet; déploiement. Impact de l'Agilité sur les équipes d'infrastructure technologique. Entretien et évolution de logiciel. Culture organisationnelle et gouvernance. Évolution du processus logiciel. Soutien au développement. Agilité et documentation. Mesures de performance organisationnelle liées à l'adoption de l'Agilité. Modèles de bonnes pratiques du développement logiciel. Impact de l'adoption de l'Agilité sur les individus.

#### MGL7260 Exigences et spécifications de systèmes logiciels

Introduction à l'ingénierie des systèmes. - Modèles de processus des exigences logicielles. - Intervenants dans le processus des exigences logicielles. - Support et gestion du processus des exigences logicielles. - Qualité et amélioration du processus des exigences logicielles. - Sources des exigences logicielles. - Techniques d'explicitation des exigences logicielles. - Classification des exigences logicielles. - Modélisation conceptuelle. - Conception architecturale et allocation des exigences logicielles. - Négociation des exigences logicielles. - Document de définition des exigences logicielles. - Document de spécification des exigences logicielles. - Structure et normes de documentation des exigences logicielles. - Qualité de la documentation des exigences logicielles. - Revue des exigences logicielles. - Prototypage. - Validation des modèles. - Tests d'acceptation. - Gestion des changements des exigences logicielles. - Attributs des exigences

logicielles. - Trace des exigences logicielles. - Sujets avancés en exigences logicielles.

### **MGL7315 Gestion de projet en génie logiciel**

Principes et gestion de projet de génie logiciel. Gestion de l'ingénierie des exigences, gestion de l'ingénierie du design, de l'ingénierie de la construction du code, des stratégies d'essais, de la maintenance et de l'évolution des logiciels. Principes et techniques de gestion spécifiques au développement de projets de génie logiciel, incluant la mesure et l'estimation, l'amélioration des processus, l'ingénierie de la qualité, les outils de soutien au développement et la gestion de configuration. Application des normes d'ingénierie du logiciel (incluant les normes ISO, IEEE et les normes industrielles) pour la planification, l'encadrement et la réalisation de projets de génie logiciel.

### **MGL7361 Principes et applications de la conception de logiciels**

Rôle de la conception dans le cycle de vie du logiciel. Apprentissage des principales méthodes de conception. Évaluation de nouvelles méthodes de conception. Sélection et utilisation d'une méthode propre à un système logiciel donné. Évaluation de la conception: choix de la méthode, qualité de la conception, vérification formelle, respect des exigences, etc. Outils de conception.

### **MGL7460 Réalisation et maintenance de logiciels**

Rôle de la réalisation et de la maintenance dans le cycle de vie du logiciel. Évolution et maintenance du logiciel. Méthodes propres à étendre la durée de vie. Sélection de la méthode appropriée de réalisation. Prototypage. Mise au point. Gestion de la maintenance. Réutilisation et rétro-ingénierie des logiciels. L'interaction entre réalisation et maintenance sera traitée tout au long du cours.

### **MGL7560 Vérification et assurance qualité de logiciels**

But et concepts de qualité des logiciels. Facteurs qualité (efficacité, exactitude, performance, facilité d'entretien). Normes d'assurance qualité et de vérification et validation (ISO, IEEE). Plans d'assurance qualité et de vérification et validation (coût, activités, ressources). Méthodes d'assurance qualité et de vérification et validation (revues, inspections, audits). Les tests: principes, méthodes, processus et plan de tests. Outils logiciels facilitant la mise en oeuvre de l'assurance qualité, de la vérification et validation de logiciels et des tests.

### **MGL7760 Qualité et productivité des outils logiciels**

Aperçu des outils pour assister au développement et à l'entretien des logiciels. Plates-formes d'intégration des outils. Environnements de développement. Outils pour la rétro-ingénierie des logiciels. Critères et stratégies d'évaluation des outils.

### **MGL7810 Sujets spéciaux en génie logiciel I**

#### Objectifs

Présenter et diffuser aux étudiants des sujets de pointe ou en émergence dans le domaine du génie logiciel.

#### Sommaire du contenu

Présentation de sujets d'intérêt majeur dans le domaine du génie logiciel et familiarisation avec les derniers développements technologiques dans un ou plusieurs domaines de pointe ou en émergence. Les sujets spéciaux seront déterminés avant les inscriptions de chaque trimestre.

### **MGL7811 Sujets spéciaux en génie logiciel II**

#### Objectifs

Présenter et diffuser aux étudiants des sujets de pointe ou en émergence dans le domaine du génie logiciel.

#### Sommaire du contenu

Présentation de sujets d'intérêt majeur dans le domaine du génie logiciel et familiarisation avec les derniers développements technologiques dans un ou plusieurs domaines de pointe ou en émergence. Les sujets spéciaux seront déterminés avant les inscriptions de chaque trimestre.

### **MGL7815 Lecture dirigée I**

#### Objectifs

Cette activité a pour objectif de permettre à l'étudiant d'approfondir un thème relatif au génie logiciel. Elle s'inscrit dans une démarche personnalisée de formation à l'intérieur du programme de maîtrise en génie logiciel.

#### Sommaire du contenu

Lectures approfondies sous la supervision d'un professeur dans un sujet relié à son domaine de spécialisation. Compléter la formation en approfondissant ou en diversifiant ses connaissances tout en développant son sens critique et son esprit de synthèse. Cette activité requiert une grande autonomie de la part de l'étudiant. Le choix des lectures et le suivi de l'étudiant sont sous la responsabilité du professeur. Des rencontres périodiques ont lieu tout au long de cette activité. Un rapport dont l'objectif, la nature et l'ampleur sont définis par le professeur doit être rédigé à la fin de l'activité. Le rapport doit être remis à la fin du trimestre de l'inscription.

#### Conditions d'accès

L'inscription à ce cours requiert l'autorisation préalable du directeur du programme.

### **MGL810 Programmation temps réel sur des architectures parallèles**

Présenter à l'étudiant les différentes architectures monoprocesseurs et celles sur les architectures parallèles. L'informer sur les différents environnements de programmation parallèle. Lui donner les méthodes essentielles à la conception de logiciels performants sur des architectures parallèles. Définition, description sommaire et identification des niveaux et des environnements de programmation parallèle. Classification des architectures parallèles SISD, SIMD, MISD et MIMD. Présentation du matériel, de l'environnement et des progiciels disponibles. Environnements de programmation et outils de support à la programmation parallèle pmake, multithread, PVM et MPI. Application des phases du génie logiciel à la conception d'algorithmes numériques adaptés à une architecture parallèle. Ajouts de considérations propres aux systèmes numériques, aux architectures parallèles et au temps réel (synchronisation, événements asynchrones, communications, opérations multiples). Application sur l'architecture disponible.

### **MGL825 Télématique et réseaux**

Approfondir sa compréhension du développement d'applications en télécommunication, en se fondant sur les couches supérieures du modèle OSI. Analyser progressivement les couches transport, session, présentation et application afin d'acquérir une compréhension avancée des services et protocoles impliqués. La conception de modèles ainsi que le développement de systèmes sont requis. Utilisation d'une méthode et d'un outil orientés objets afin de mieux maîtriser les différents concepts. Conception de systèmes télématiques à l'aide de l'outil. Ces systèmes sont exclusivement de la couche application comme, par exemple, les protocoles MHS (messagerie électronique), FTAM (transfert de fichier) ou autres. Ce cours utilise des outils de développement de méthodes semi-formelles UML, ainsi que de méthodes formelles SDL.

#### Préalables académiques

Des connaissances en programmation orientée objet sont requises.

### **MGL835 Interaction humain-machine**

À la suite de ce cours, l'étudiant sera en mesure : - de concevoir des interfaces utilisateurs en appliquant une démarche centrée sur l'utilisateur; - d'incorporer des techniques récentes et des fonctionnalités interactives novatrices à la conception d'un système; - d'appliquer des méthodes d'évaluation pour valider les prototypes et guider leur modification. Étapes de spécification, de conception, de développement, et d'évaluation des interfaces utilisateurs. Conception itérative et centrée sur l'utilisateur. Analyse des tâches. Directives de conception. Programmation événementielle. Styles et techniques d'interaction (interaction gestuelle, haptique, tridimensionnelle, oculaire, etc.). Périphériques d'entrée et de sortie. Visualisation en 2D et 3D. Loi de Fitts et techniques de modélisation prédictive. Méthodes qualitatives et quantitatives d'évaluation des interfaces. Récents développements

technologiques et axes de recherche.

#### **MGL841 La mesure : concept clé en ingénierie du logiciel**

À la fin du cours, l'étudiant sera en mesure : d'appliquer les concepts fondamentaux requis pour une utilisation professionnelle des mesures et des modèles quantitatifs en génie logiciel; de collecter et analyser des données; de construire des modèles de productivité et d'estimation du logiciel. Concepts clés en métrologie. Mesure de la taille fonctionnelle des logiciels, ISO 19761. Processus de validation et de vérification du design des mesures. Modèles de productivité et d'étalonnage. Modèles d'estimation. Programme corporatif de mesure. Utilisation d'un référentiel international de données de projets de développement de logiciels.

#### **MGL842 L'ingénierie de la qualité du logiciel**

À la fin du cours, l'étudiant sera en mesure : d'analyser les exigences de haut niveau (d'affaires); d'en extraire les exigences qualité; de les transformer en cibles quantitatives de qualité et de les intégrer à l'intérieur d'un processus d'implantation de la qualité du logiciel. Concepts et méthodes d'ingénierie de la qualité du logiciel. Modèles et processus permettant d'identifier, définir et formaliser les exigences qualité, les processus de transposition des exigences haut niveau (d'affaires) aux mesures de qualité, de même que les méthodes de contrôle de traçabilité et la documentation. Méthode, modèle et processus d'implantation de la qualité avec une analyse comparative en utilisant les modèles de développement du logiciel reconnus dans l'industrie. Approche consolidée, utilisant la méthode de formalisation des exigences qualité et le modèle d'implantation de la qualité pour effectuer un processus complet d'ingénierie de la qualité du logiciel.

#### **MGL843 Sujets avancés en conception logicielle**

Sujets avancés en conception de logiciels. Pratiques à l'avant-garde de la conception de logiciels, la visualisation des éléments de la conception, les design patterns (motifs ou patrons de conception), la stabilité, la traçabilité des exigences non fonctionnelles, la fiabilité, l'agilité, la refactorisation, tout avec une perspective orientée-objet.

#### **MGL844 Architecture logicielle**

Ce cours met l'emphase sur les attributs de qualité comme pilotes des activités d'analyse, d'élaboration, d'évaluation et d'implémentation de l'architecture logicielle. Architecture et cycle de vie, attributs de qualité, tactiques architecturales, styles/patrons architecturaux avec emphase sur les styles modernes (infonuagique - "cloud computing", architectures orientées services - SOA), formalismes pour exprimer une architecture logicielle (notations informelles, UML, langages de description architecturale – ADL), rétro-ingénierie/redécouverte architecturale, méthodes de conception architecturale, évaluation architecturale, lignes de produits logiciels.

#### **MGL845 Ingénierie dirigée par les modèles**

Ce cours présente les principes de l'ingénierie logicielle dirigée par les modèles. En particulier, le cours aborde le processus de développement logiciel par transformations de modèles et les concepts de modèles indépendants des plateformes et modèles spécifiques aux plateformes. Ce cours couvre aussi les principes et les standards de modélisation et de méta-modélisation, les langages spécifiques aux domaines et l'architecture dirigée par les modèles MDA (Model-Driven Architecture) de l'OMG (Object management group).

#### **MGL846 Concepts et pratique des tests logiciels**

Fondements des tests logiciels : terminologie, questions clés des tests logiciels, relation des tests logiciels avec les autres activités du cycle de vie logiciel. Niveaux de tests : cibles des tests logiciels, objectifs des tests logiciels. Techniques de tests logiciels. Mesures des tests logiciels : évaluation des programmes testés, évaluation des tests effectués. Processus des tests logiciels : considérations pratiques des tests logiciels, activités des tests logiciels. Outils de tests logiciels. Sujets avancés en tests logiciels.

#### **MGL847 Amélioration des processus logiciels : intervention dans une organisation**

À la suite de ce cours, l'étudiant sera en mesure : d'effectuer, en

équipe, une intervention en industrie; de rédiger un cas d'affaires; de rédiger un plan de communication et un plan d'amélioration des processus; de définir un processus logiciel, d'identifier et de gérer les principaux risques associés à un projet d'amélioration; d'identifier les facteurs organisationnels pouvant nuire ou aider à l'amélioration de processus; de réaliser et d'exécuter un plan de mise en œuvre; de réaliser un bilan du projet d'intervention. Aperçu de l'ensemble de la problématique et de la démarche d'amélioration des performances des processus logiciels (productivité, cycle de développement et qualité). Approche d'amélioration du processus logiciel selon le modèle d'amélioration "IDEAL" du Software Engineering Institute. Enjeux organisationnels. Analyse de la culture de l'organisation. Évaluation des commanditaires d'un projet d'amélioration. Analyse du système de motivation et du niveau de stress. Évaluation des agents de changement. Analyse de la promptitude individuelle au changement. Cas d'affaires. Plan de communication. Modèle d'évolution des capacités (CMMI®) du Software Engineering Institute. Normes applicables (ISO et IEEE). Les normes ISO/IEC 29110 pour les très petits organismes. Les forces et faiblesses des normes et modèles. Diagnostic d'un processus logiciel. Définition et documentation d'un processus logiciel. Évaluation et mitigation des risques d'un projet d'amélioration. Plan d'amélioration du processus. Bilan du projet d'intervention. Projet d'intervention, en équipe de trois étudiants, dans une organisation.

#### **MGL848 Validation et vérification de modèles en génie logiciel**

Ce cours vise à procurer à l'étudiant une connaissance approfondie des méthodes formelles et semi-formelles pour la description et l'analyse de matériels ou de produits logiciels. Il vise également à faire comprendre les avantages et les limites de ces méthodes. À la fin du cours, l'étudiant sera en mesure de produire un modèle abstrait et formel d'un système, de l'utiliser pour démontrer certaines propriétés, de d'expliquer comment ces propriétés répondent aux spécifications essentielles/critiques du système. L'étudiant devra également être capable de lire et comprendre des spécifications formelles écrites par des professionnels et de produire des spécifications formelles de systèmes de complexité moyenne. Il possèdera une très bonne connaissance des méthodes existantes et saura discuter intelligemment des avantages et des inconvénients de l'utilisation de telles approches.

#### **MGL849 Modélisation, analyse et programmation des systèmes temps réel**

Les systèmes temps réel et embarqués sont omniprésents. Ces systèmes sont souvent caractérisés par des contraintes de temps sévères. En outre, ils sont naturellement concurrents, distribués et souvent critiques. La complexité de tels systèmes est continuellement en croissance. Par conséquent, la conception et l'implémentation de systèmes temps réels corrects et fiables sont des tâches cruciales et complexes. La modélisation de ces systèmes utilisant des méthodes et langages de modélisation standards, comme AADL ou UML MARTE, permet d'une part de maîtriser leur complexité, et d'autre part, d'utiliser des techniques d'analyse sophistiquées comme l'analyse d'ordonnement et d'estimation du temps de réponse au pire cas. Dans ce cours, on introduit les concepts, terminologies et problématiques relatives aux systèmes temps réel et embarqués; le paradigme de programmation concurrente et les problématiques associées à la concurrence; les notions de fiabilité et les techniques de tolérance aux fautes; les méthodes et langages standards de modélisation; et les techniques d'analyse de systèmes temps réel.

#### **MGL8705 Stages industriels et rapport technique**

Objectifs

L'objectif du stage est de faire appliquer les connaissances acquises en génie logiciel dans un milieu de travail afin de parfaire la formation.

Sommaire du contenu

Stage réalisé en entreprise permettant à l'étudiant d'apporter une contribution significative à la solution d'un problème réel de génie logiciel dans le milieu technologique, avec ses contraintes économiques, techniques et autres. Le contenu du stage est en fonction du ou des mandats confiés au stagiaire par l'employeur. L'étudiant remettra un rapport technique suite au stage selon les

consignes données par la direction du programme

#### Modalité d'enseignement

Travail d'au moins quatre mois en entreprise dont la semaine comporte un minimum de 35 heures de travail, sous la supervision d'un professeur. La notation du cours est succès ou échec.

#### Conditions d'accès

Avoir complété les cours obligatoires du programme et avoir une moyenne cumulative de 3.2 sur 4.3. Être citoyen canadien, résident permanent ou étudiant étranger détenteur d'un permis de travail valide au Canada. L'étudiant doit aviser la direction du programme son intention d'effectuer un stage au moins 4 mois avant le trimestre prévu pour effectuer le stage.

#### **MGL8706 Rapport technique II**

##### Objectifs

L'objectif du cours est d'approfondir une problématique de génie logiciel vécue ou constatée lors du stage. Le sujet du rapport sera autorisé par la direction du programme.

##### Sommaire du contenu

Suite à la réalisation du stage, l'étudiant sera en mesure d'approfondir une problématique de génie logiciel vécue ou constatée lors du stage en entreprise. L'étudiant fera une analyse du problème et proposera une solution. L'étudiant aura à concevoir un rapport selon les directives du programme, comportant entre autres : la description du problème, son analyse et des pistes de solutions.

#### Modalité d'enseignement

Travail fait sous la supervision d'un professeur.

#### Préalables académiques

MGL8705 Stages industriels et rapport technique

#### **MGL8707 Projet technique**

##### Objectifs

Le projet technique permet à l'étudiant d'intégrer ses connaissances et de les utiliser concrètement dans le cadre d'un projet pratique.

##### Sommaire du contenu

Sous la supervision d'un professeur, l'étudiant réalisera un projet pratique de génie logiciel et la rédaction d'un rapport. Le projet doit se conformer aux directives fournies par la direction du programme. L'accomplissement du projet et la rédaction du rapport requièrent au moins 270 heures, réparties sur un ou deux trimestres. Le rapport fera l'objet d'une présentation devant le comité d'évaluation du programme.

#### Modalité d'enseignement

Le sujet du projet et son envergure doivent être approuvés par le directeur du programme avant d'entreprendre le projet.

#### Conditions d'accès

Avoir complété les cours obligatoires du programme

#### **MGR850 Sécurité de l'Internet**

Pertinence: L'importance de la sécurité de l'Internet est une nécessité. Dans la réalité moderne des réseaux interreliés et des applications critiques, la sécurité n'est plus considérée comme une valeur ajoutée mais bien un aspect englobant du plan de développement. Donner à l'étudiant des connaissances approfondies des moyens nécessaires pour rendre sécuritaires les échanges par Internet. Problématique de la sécurité. Terminologie. Notion de confiance. Identification des faiblesses de l'Internet. Types d'attaques possibles contre chacune des faiblesses. Analyse des risques. Enjeux d'éthique. Mécanismes de protection disponibles. Pratiques préventives. Contre-mesures. Techniques de cryptographie. Mécanismes de base.

#### **MTI825 Gestion des services TI**

À la suite de ce cours, l'étudiant sera en mesure : - d'analyser les besoins en services TI; - de spécifier les services TI requis; - d'élaborer un cadre de gestion de services TI. Fourniture des services des TI

(Service Delivery); soutien des services aux TI (Service Support); gestions des infrastructures TI; spécification et gestion des niveaux de services; gestion des services TI dans un mode d'impartition. Services TI et sécurité informatique. Introduction à l'amélioration des processus.

#### **SYS869 Sujets spéciaux I : génie logiciel**

Sujets d'intérêt majeur dans le domaine de la technologie des systèmes et familiarisation avec les derniers développements technologiques dans un ou plusieurs domaines de pointe. Sujets particuliers dans différentes spécialités du domaine de la technologie des systèmes

#### **SYS870 Sujets spéciaux II : génie logiciel**

Sujets d'intérêt majeur dans le domaine de la technologie des systèmes et familiarisation avec les derniers développements technologiques dans un ou plusieurs domaines de pointe. Sujets particuliers dans différentes spécialités du domaine de la technologie des systèmes.

N.B. : Le masculin désigne à la fois les hommes et les femmes sans aucune discrimination et dans le seul but d'alléger le texte.  
Cet imprimé est publié par le Registrariat. Basé sur les renseignements disponibles le 25/01/21, son contenu est sujet à changement sans préavis.  
Version Hiver 2022