

## Réalisation et maintenance de logiciels

Groupe 40

Jeudi, de 18h00 à 21h00 Voir local au: <https://portail.tudiant.uqam.ca/> (cours)

---

### Responsable(s) du cours

**Nom du coordonnateur :** TREMBLAY, Guy**Nom de l'enseignant :** BERGER, Jacques**Local :** PK-4115**Téléphone :** (514) 987-3000 #3699**Courriel :** [berger.jacques@uqam.ca](mailto:berger.jacques@uqam.ca)**Site Web :** [www.jberger.org](http://www.jberger.org)

---

### Description du cours

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.

---

### Objectifs du cours

- **Objectif général:** Se familiariser avec les concepts, pratiques, techniques et outils visant à développer et maintenir des logiciels de façon professionnelle.
- **Objectifs spécifiques:**
  - Comprendre la problématique de la réalisation et de la maintenance de logiciels.
  - Situer les activités de réalisation et de maintenance dans le cycle de vie du logiciel.
  - Connaître les approches contemporaines de réalisation et de maintenance.
  - Savoir utiliser des outils de contrôle du code source.
  - Comprendre l'importance des activités d'assemblage et de déploiement des logiciels.
  - Connaître les principaux outils et techniques de test et pouvoir les utiliser (tests unitaires et tests d'acceptation).
  - Connaître et utiliser des outils d'analyse de code pour en évaluer la qualité et la maintenabilité.
  - Comprendre la notion de "dette technique" et le rôle du «refactoring»
  - Se familiariser avec de nouvelles approches dans le domaine et de nouveaux langages de programmation.

---

### Contenu du cours

- Mesures de maintenabilité d'un logiciel
- Évaluation de maintenabilité
- Systèmes patrimoniaux
- Test-Driven Development
- Langages de programmation dynamique

- Performance et optimisation
- Programmation polyglotte
- Méthodes agiles
- Estimer et planifier l'effort de maintenance
- Pratiques industrielles de la réalisation et de la maintenance
- Logiciels libres
- Générateurs de code
- Normes
- Environnements de développements, de tests, de maintenance

## Formules pédagogiques

Le cours se base sur une approche par projet. L'approche par projet s'inscrit dans l'esprit de la formation par compétences. Elle permet la mobilisation des ressources de l'étudiant-e dans la réalisation d'une tâche complexe. Les étudiant-e-s devront travailler en équipe (4 personnes).

Plusieurs des périodes de cours seront divisées en deux parties (pas nécessairement égales). La première partie abordera divers thèmes, utiles pour la réalisation des projets. La deuxième partie consistera soit en une activité pratique en laboratoire, soit servira à réviser avec chaque équipe le déroulement des travaux entrepris dans le but d'éviter des dérives éventuelles.

## Modalités d'évaluation

Description sommaire	Date (sujet à changement)	Pondération
Projet 1		15%
Projet 2		15%
Projet 3		15%
Participation en classe		5%
Examen intra		25%
Examen final		25%

Un travail remis en retard reçoit la note zéro à moins d'avoir fait l'objet d'une entente préalable avec le professeur.

Le détail des conditions de réalisation de chaque travail est précisé avec la description du travail.

La qualité du français sera prise en considération, tant dans les examens que dans les travaux pratiques (jusqu'à 10 % de pénalité).

La note de passage du cours est de 60% pour l'ensemble des évaluations et de 50% pour l'examen final.

### Politique d'absence aux examens

**L'autorisation de reprendre un examen en cas d'absence est de caractère exceptionnel. Pour obtenir un tel privilège, l'étudiant-e doit avoir des motifs sérieux et bien justifiés.**

Il est de la responsabilité de l'étudiant-e de ne pas s'inscrire à des cours qui sont en conflit d'horaire, tant en ce qui concerne les séances de cours ou d'exercices que les examens. **De tels conflits d'horaire ne constituent pas un motif justifiant une demande d'examen de reprise.**

Dans le cas d'une absence pour raison médicale, l'étudiant-e doit joindre un certificat médical original et signé par le médecin décrivant la raison de l'absence à l'examen. Les dates d'invalidité doivent être clairement indiquées sur le certificat. Une vérification de la validité du certificat pourrait être faite. Dans le cas d'une absence pour une raison non médicale, l'étudiant-e doit fournir les documents originaux expliquant et justifiant l'absence à l'examen &ndash; par exemple, lettre de la Cour en cas de participation à un jury, copie du certificat de décès en cas de décès d'un proche, etc. Toute demande incomplète sera refusée. Si la direction du programme d'études de l'étudiant-e constate qu'un étudiant a un comportement récurrent d'absence aux examens, l'étudiant-e peut se voir refuser une reprise d'examen.

L'étudiant-e absent-e lors d'un examen doit, dans les cinq (5) jours ouvrables suivant la date de l'examen, présenter une demande de reprise en utilisant le formulaire prévu, disponible sur le site Web du département à l'adresse suivante : <http://info.uqam.ca/politiques/>

L'étudiant-e doit déposer le formulaire dûment complété au secrétariat de la direction de son programme d'études : PK-3150 pour les programmes de premier cycle, PK-4150 pour les programmes de cycles supérieurs. Pour plus de détails sur la politique d'absence aux examens du Département d'informatique, consultez le site web suivant : <http://info.uqam.ca/politiques>

## Intégrité académique

### PLAGIAT Règlement no 18 sur les infractions de nature académique. (extraits)

Tout acte de plagiat, fraude, copiage, tricherie ou falsification de document commis par une étudiante, un étudiant, de même que toute participation à ces actes ou tentative de les commettre, à l'occasion d'un examen ou d'un travail faisant l'objet d'une évaluation ou dans toute autre circonstance, constituent une infraction au sens de ce règlement.

La liste non limitative des infractions est définie comme suit :

- la substitution de personnes;
- l'utilisation totale ou partielle du texte d'autrui en la faisant passer pour sien ou sans indication de référence;
- la transmission d'un travail pour fins d'évaluation alors qu'il constitue essentiellement un travail qui a déjà été transmis pour fins d'évaluation académique à l'Université ou dans une autre institution d'enseignement, sauf avec l'accord préalable de l'enseignante, l'enseignant;
- l'obtention par vol, manœuvre ou corruption de questions ou de réponses d'examen ou de tout autre document ou matériel non autorisés, ou encore d'une évaluation non méritée;
- la possession ou l'utilisation, avant ou pendant un examen, de tout document non autorisé;
- l'utilisation pendant un examen de la copie d'examen d'une autre personne;
- l'obtention de toute aide non autorisée, qu'elle soit collective ou individuelle;
- la falsification d'un document, notamment d'un document transmis par l'Université ou d'un document de l'Université transmis ou non à une tierce personne, quelles que soient les circonstances;
- la falsification de données de recherche dans un travail, notamment une thèse, un mémoire, un mémoire-crédation, un rapport de stage ou un rapport de recherche;
- Les sanctions reliées à ces infractions sont précisées à l'article 3 du Règlement no 18.

Les règlements concernant le plagiat seront strictement appliqués. Pour plus de renseignements, veuillez consulter les sites suivants : <http://www.sciences.uqam.ca/etudiants/integrite-academique.html> et <http://www.bibliotheques.uqam.ca/recherche/plagiat/index.html>

## Médiagraphie

VO R. C. Martin **Clean Code**. Prentice-Hall, 2009

VR A. Hunt and D. Thomas. **The Pragmatic Programmer--From Journeyman to Master**. Addison-Wesley, 2000.

VC K. Beck. **Test-Driven Development--By Example**. Addison-Wesley, 2003.

VC J.A. Campbell and P.P. Papateou. **SonarQube In Action**. Manning, 2013.

VC J. Humble and D. Farley. **Continuous Delivery--Reliable Software Releases Through Build, Test, and Deployment Automation**. Addison-Wesley, 2011.

VC A. Hunt. **Pragmatic Thinking and Learning--Refactor Your Wetware**. The Pragmatic Bookshelf, 2008.

VC J. Rasmusson. **The Agile Samurai--How Agile Masters Deliver Great Software**. The Pragmatic Bookshelf, 2010.

VC J.F. Smart. **BDD In Action**. Manning, 2015.

VC V. Subramaniam and A. Hunt. **Practices of an Agile Developer--Working in the Real World**. The Pragmatic Bookshelf, 2006.

VC J. Visser. **Building Maintainable Software--Ten Guidelines for Future-Proof Code (Java Edition)**. O&rsquo;Reilly, 2016.