

GROUPE	40 GABRINI, Philippe	gabrini.philippe@uqam.ca	(514) 987-3000 1728	PK-4940
	Jeudi, de 17h30 à 20h30			

DESCRIPTION	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.
-------------	---

OBJECTIF	Acquérir une bonne compréhension et un point de vue critique du développement de logiciels et de ses nombreux aspects, ainsi que des différences existant entre le développement et la maintenance de logiciels.
----------	--

ÉVALUATION	Description sommaire	Date	Pondération
	Projet de session: Adéquation du code d'un petit logiciel industriel aux principes exprimés dans Code Complete 2; une étude critique.		50%
	Présentation orale sur un aspect de la maintenance, sujet à choisir avec l'accord du professeur.		20%
	Présentation orale du projet		20%
	Participation aux discussions en classe		10%

#### Critères

- Organisation du travail (plan)
- Références additionnelles
- Définition et compréhension des notions abordées
- Transition entre les différentes parties
- Profondeur des réflexions
- Bien fondé des critiques
- Synthèse
- Argumentation et présentation
- Clarté, qualité du français

CONTENU	<p><b>Description des sujets abordés</b></p> <p><u>Développement</u></p> <p>Processus de développement. Amélioration du produit. Équipe : objectifs, priorités, approche systématique, stratégies, feed-back, vision en avant, besoins réels, savoir dire non, on ne peut plaire à tout le monde (y compris à ses supérieurs). Tout a un coût. Caractéristiques désirables. Rapports lus. Réunions utiles. Dates de livraison raisonnables et arbitraires. Diviser pour régner. Pas de stagnation. Apprentissage continu. Réaction rapide. Le mythe du code sans bogues. Trop de travail. Pas de livraison complète. Code partageable. Longues heures et longs week-ends. Travailler fort # de penser fort.</p> <p><u>Construction</u></p> <p>Construction du logiciel. Réaction aux changements des spécifications durant la construction. Résistance aux erreurs. Langage de développement des programmes (pseudocode). Promotion de la réutilisation de code. Cohésion forte et couplage lâche. Programmation défensive. Modularité réelle et simulée. Dissimulation de l'information. Conception = processus heuristique. Importance des noms et des types. Variables globales. Types de données abstraits. Imbrications dangereusement profondes. Puissance de la programmation structurée. Structures de contrôle et complexité. Techniques de formatage. Code auto documenté. Style de programmation comme aide à la documentation. Outils : de conception, pour code source. Effet de la taille des projets sur les erreurs et la productivité. Gestion de la configuration. Traitement des programmeurs. Techniques pour augmenter la qualité du logiciel. Assurance de qualité. Revues. Vérifications unitaires. Mise au point. Intégration du système : par étapes., incrémentielle. Mise au point du code. Évolution du logiciel.</p> <p><u>Écriture de code</u></p> <p>Assertions et hypothèses. Programmation défensive. Algorithmes en double. Nettoyage. Forçage des erreurs. Journal des bogues. Chasse aux bogues. Conception des tests. Suivre le code à la trace. Flux des données. Plein feux sur les erreurs. Défauts des interfaces. Fonctions à idée fixe. Fonctions sans échecs. Code lisible. Commentaires avertisseurs. Redondance du code. Idioms risqués. Mélanges oiseux. Mémoire dangereuse. Fonctions parasites. Code pour programmeur moyen. Réparation d'erreurs immédiate. Nettoyage de code, danger! Tranches de code. Vérification d'abord, livraison ensuite. Équipe de vérification.</p>
---------	---

Maintenance

Particularités. Performance. Métriques. Qualité : composants et coûts. Productivité. Motivation : attitude, reconnaissance, attentes et mission des départements.

## CALENDRIER

Période	Contenu	Lecture et laboratoire
1	Présentation du cours; construction de logiciels.	
2	Métaphores du développement de logiciel; préalables de la construction; décisions vitales.	
3	Conception; classes de travail; Sous-programmes de qualité.	
4	Programmation défensive; pseudocode; utilisation des variables.	
5	Types de données fondamentaux; types de données inhabituels; organisation du code.	
6	Conditionnels; récursivité; tables; structures de contrôle.	
7	Présentations.	
8	Présentations.	
9	Qualité du logiciel; collaborations; tests.	
10	Mise au point; restructuration; amélioration de la performance.	
11	Taille du logiciel; gestion de la construction; Intégration.	
12	Outils; disposition et style; code auto-documenté.	
13	Caractéristiques personnelles; thèmes en artisanat logiciel.	
14	Présentations.	
15	Présentations.	

## RÉFÉRENCES

- VO Steve McConnell – *Code Complete 2* – Microsoft Press, 2004, 914 pages (voir sa bibliographie).  
OU SA TRADUCTION
- VO Steve McConnell – *Tout sur le code; pour concevoir du logiciel de qualité dans tous les langages* – ISBN 2-10-048753-1 Dunod 2005.
- VC Penny Grubb, Armstrong A. Takang – *Software Maintenance: Concepts and Practice* – 2nd edition World Scientific Publishing ; ISBN: 981238426X, 2003, 372 pages.
- VC *IEEE Computer Society Guide to the Software Engineering Body of Knowledge* – 2004 edition. – <http://www.swebok.org/>  
Chapter 4 Software Construction, Chapter 6 Software Maintenance.
- VC Andrew Hunt, David Thomas – *The pragmatic programmer* – Addison-Wesley, 2000, 321 pages.
- SC *IEEE Std 1219-1998 IEEE Standard for Software Maintenance*, 47 pages.
- VC Jerome B. Landsbaum, Robert L. Glass – *Measuring and Motivating Maintenance Programmers* – Prentice-Hall 1992, 96 pages.
- VC Steve Maguire – *Writing Solid Code* – Microsoft Press 1993, 256 pages.
- VC Steve Maguire – *Debugging the Development Process* – Microsoft Press 1994, 183 pages.

A : article – C : comptes rendus – L : logiciel – N : notes – R : revue –  
S : standard – U : uri – V : volume

C : complémentaire – O : obligatoire – R : recommandé