

Sigle : INF1583 Gr. 01

Titre : Développement des systèmes informatiques

Session : Automne 2019 Horaire et local

Professeur : Iglewski, Michal et Levasseur, Valérie

1. Description du cours paraissant à l'annuaire :

Objectifs

Introduire l'étudiant à l'approche professionnelle utilisée dans les différentes étapes du développement de systèmes informatiques (y compris des systèmes à temps réel). Lui présenter les différentes techniques formelles existantes dans le domaine et montrer comment les appliquer dans un projet informatique. Lui apprendre à documenter un système informatique. Introduire l'étudiant au travail en équipe.

Contenu

Principaux types de documents dans un système informatique et définitions de leurs contenus. Rôle des spécifications et méthodes formelles. Introduction à certaines notations formelles pour décrire les exigences et spécifications de composants et systèmes logiciels : modélisation abstraite, spécifications algébriques des types abstraits et/ou automates et systèmes de transition. Méthode rigoureuse de développement. Techniques de validation, tests, inspection. Travail en équipe, gestion de l'équipe, productivité, métriques. Outils de programmation supportant le développement des logiciels.

Descriptif – Annuaire

2. Objectifs spécifiques du cours :

Apprentissage par problèmes et par projets.

3. Stratégies pédagogiques :

Au cours de cette activité, diverses formules pédagogiques seront utilisées, notamment : cours magistraux, devoirs, exercices en classe et lectures personnelles. Les attentes sont que les étudiant(e)s investissent au moins 90 heures de travail personnel en plus des 45 heures de cours.

Les devoirs seront effectués en utilisant Angular, Firebase et GitHub comme outils de gestion de projets.

La page Moodle sera utilisée pour communiquer d'autres informations au cours de la session.

Huit séances de travaux dirigés (2 h/semaine) viendront appuyer les séances de cours. Ces séances se font en présence d'un assistant. La présence des étudiants à ces laboratoires est obligatoire.

4. Heures de disponibilité ou modalités pour rendez-vous :

Consultations : mardi : 11 h 30 à 12 h 30 ou sur rendez-vous.

Les étudiant(e)s sont prié(e)s de respecter les heures des consultations. En dehors des heures de consultation, les foires aux questions de Moodle constituent la voie principale de communication entre les étudiant(e)s et le professeur.

5. Plan détaillé du cours sur 15 semaines :

Semaine	Thèmes	Dates
1	<ul style="list-style-type: none"> • Présentation du cours <ul style="list-style-type: none"> ○ Présentation du plan de cours ○ Présentation du projet de session • Gestion d'un système en développement 	03 sept. 2019

	<ul style="list-style-type: none"> ○ Comment savoir que l'on développe le « bon » système : introduction à la vérification et la validation de systèmes ○ Gestion des versions <p>TD – le 05 septembre 2019</p> <ul style="list-style-type: none"> • Utilisation du système de gestion de versions Git 	
2	<p>Outils de développement</p> <ul style="list-style-type: none"> • Les assertions <ul style="list-style-type: none"> ○ Définition et utilisation ○ Exemples en Java • Les tests <ul style="list-style-type: none"> ○ Les types de tests ○ L'utilisation des assertions dans les tests ○ Exemples de tests unitaires avec JUnit • Introduction à la qualité du code <ul style="list-style-type: none"> ○ Qu'est-ce que du code de qualité? ○ Comment évaluer la qualité du code? <p>À remettre avant le début de la séance : Compte GitHub utilisé pour le cours</p> <p>TD – le 12 septembre 2019</p> <ul style="list-style-type: none"> • Atelier de mise à niveau sur les langages et cadres d'application utilisés dans le projet <ul style="list-style-type: none"> ○ Mini application Angular ○ Firestore : requêtes et manipulation d'une base de données NoSQL • Gestion des dépendances (versionnage sémantique) 	10 sept. 2019
3	<p>Qualité du code (suite)</p> <ul style="list-style-type: none"> • Inspection <p>Étapes de production d'un livrable : les outils de build</p> <ul style="list-style-type: none"> • Outils make, ant, Maven • Gestion des dépendances dans Java et Javascript <p>Les métriques logicielles</p> <ul style="list-style-type: none"> • SonarQube <p>TD – le 19 septembre 2019</p> <ul style="list-style-type: none"> • Accès à SonarQube • Écriture de tests et calcul de couverture • Inspection de code 	17 sept. 2019
4	Utilité des méthodes formelles (notions de génie logiciel)	24 sept. 2019

	<ul style="list-style-type: none"> • Vérification formelle <p>Outils mathématiques</p> <ul style="list-style-type: none"> • Paradoxes, ensembles, formules logiques, variables libres • Construction « <i>where</i> » • Fonctions et relations • Prédicats • Relation d'équivalence • Représentations tabulaires • Automates à états <p>TD – le 26 septembre 2019</p> <ul style="list-style-type: none"> • Atelier de mise à niveau sur l'utilisation des outils mathématiques 	
5	<p>Spécifications des programmes</p> <ul style="list-style-type: none"> • Quelques concepts <ul style="list-style-type: none"> ○ Programmes vs modules ○ Déterminisme vs non-déterminisme • Spécification des programmes <ul style="list-style-type: none"> ○ Qu'est-ce qu'une spécification et qu'est-ce qu'elle doit contenir? ○ Relation entre le code et sa spécification ○ Spécification de programmes déterministes sous forme de fonctions • Spécification de programmes non déterministes <ul style="list-style-type: none"> ○ Comment faire la spécification de programme non déterministe? ○ Introduction aux LD-relations <p>TD – le 03 octobre 2019</p> <ul style="list-style-type: none"> • Exercices sur la spécification de programmes 	01 oct. 2019
6	<p>Examen de mi-session (2 heures)</p> <p>Présentations du devoir #2.</p>	08 oct. 2019
7	<p>Semaine d'études</p>	15 oct. 2019
8	<p>Aperçu de la documentation lors du développement des systèmes informatiques</p> <p>Spécifications des programmes (suite)</p> <ul style="list-style-type: none"> • programmes déterministes et non déterministes • spécifications des programmes sous forme de « fonctions de programme » (LD-relations) • tables décrivant les fonctions de programme • exemples de spécifications • documentation de programmes avec « displays » • étude de cas 	22 oct. 2019
9	<p>Modules</p> <ul style="list-style-type: none"> • Conception modulaire <ul style="list-style-type: none"> ○ Définition des modules 	29 oct. 2019

	<ul style="list-style-type: none"> ○ Principes de conception ○ Études de cas • Aperçu de la documentation des modules <ul style="list-style-type: none"> ○ Guide des modules, spécification de l'interface, document de conception interne des modules • Guide des modules <p>TD – le 31 octobre 2019</p> <ul style="list-style-type: none"> • Spécification de programmes à l'aide des LD-relations • Documentation de programmes avec « <i>displays</i> » 	
10	<p>Spécifications d'interfaces de modules (suite)</p> <ul style="list-style-type: none"> • Exemples en Lotos et Z • Les spécifications <ul style="list-style-type: none"> ○ Rappels ○ Propriétés ○ Techniques de spécifications ○ Règles, structure et contenu • Spécification de modules vs spécification de programmes • Contenu de la spécification d'interfaces de modules 	05 nov. 2019
11	<p>Spécifications d'interfaces de modules (suite)</p> <ul style="list-style-type: none"> • Retour sur le contenu de la spécification d'interfaces de modules • Règles pour des spécifications formelles et abstraites • Méthode de traces • Exemples de spécifications de module 	12 nov. 2019
12	<p>Documentation interne de modules</p> <ul style="list-style-type: none"> • Choix de la structure de données et définition de la fonction d'abstraction • Exemples <p>TD – le 21 novembre 2019</p> <ul style="list-style-type: none"> • Spécification d'interfaces de module 	19 nov. 2019
13	<p>Processus de conception des systèmes informatiques</p> <ul style="list-style-type: none"> • Principes de conception des systèmes informatiques <ul style="list-style-type: none"> ○ Approche fonctionnelle ○ Documentations des besoins • Présentation du dernier devoir • Étude de cas : système de gestion d'examens <ul style="list-style-type: none"> ○ Définition et spécification des besoins ○ Spécification d'interfaces et conception interne de modules <p>TD – le 28 novembre 2019</p>	26 nov. 2019

	<ul style="list-style-type: none"> • Conception interne des modules • Exercices de révision 	
14	Retour sur les étapes de développement <ul style="list-style-type: none"> • Retour sur le contenu du cours • Exercices de révision 	03 déc. 2019
15	Examen final	10 déc. 2019

6. Évaluation du cours :

L'évaluation est l'appréciation du niveau d'apprentissage atteint par l'étudiant(e) par rapport aux objectifs des cours et des programmes.

L'attribution des notes se fera selon la répartition suivante :

- Examen de mi-session : 15 %
- Examen de fin de session : 25 %
- Devoirs : 60 %

Pour les devoirs d'équipe, les notes peuvent être attribuées d'une manière individuelle selon la contribution de l'étudiant(e).

Attention : La présence aux cours est fortement recommandée.

7. Politiques départementales et institutionnelles :

- Politique du département d'informatique et d'ingénierie relative à la tenue des examens
- Note sur le plagiat et sur la fraude
- Politique relative à la qualité de l'expression française écrite chez les étudiants et les étudiantes de premier cycle à l'UQO
- Absence aux examens : cadre de gestion, demande de reprise d'examen (formulaire)

• **SANS OUI**

C'EST NON!

Travaillons ensemble pour développer une culture du respect ! La communauté universitaire de l'UQO se mobilise et lance un message haut et fort de **tolérance zéro en matière de violence à caractère sexuel** (pour de plus amples renseignements, veuillez consulter la page Web : uqo.ca/sansouicestnon).

8. Principales références :

Volumes suggérés :

- Sommerville, I.; Software Engineering; Addison-Wesley, 2010, 9/e, ISBN: 9780321313799.
- Articles :
 - Articles : Parnas, D.L., Madey, J., "Functional Documents for Computer Systems", Science of Computer Programming, 25, 1, pp. 41-61.
 - Parnas, D.L., "On the Criteria to be Used in Decomposing Systems into Modules", Communications of the ACM, Vol. 15, No. 12, December 1972, pp. 1053-1058.
 - Parnas, D.L., Madey, J., Iglewski, M., "Formal Documentation of Well-Structured Programs", IEEE Transactions on Software Engineering, Vol. 20, No.12, 1994, pp.948-976.
 - Parnas, D.L., Clements, P.C., "A Rational Design Process: How any Why to Fake It", IEEE Transactions on Software Engineering, Vol. SE-12, No. 2, February 1986, pp. 251-257.

Volumes de références :

- Brooks, F.P., No Silver Bullet--Essence and Accidents of Software Engineering, Computer, 20(4), 1987 (aussi un chapitre dans The Mythical Man Month: Essays on Software Engineering, Anniversary Edition, Addison-Wesley, 1995).
- Ghezzi, C., Jazayeri, M., Mandrioli, D.; Fundamentals of Software Engineering; 2^e, Prentice-Hall, 2002, ISBN 0-13-305699-6.
- Pressman, R.; Software Engineering: A Practitioner's Approach; McGraw-Hill, 7/e, 2010, ISBN: 0073375977.

9. Page Web du cours :

<https://moodle.uqo.ca>