

PLAN DE COURS

Cours	INF 721 – Mesures et indicateurs du génie logiciel
Trimestre	Été 2018
Enseignant	Évariste Valéry BÉVO WANDJI

1. MISE EN CONTEXTE

Depuis plusieurs décennies maintenant on parle de « crise du logiciel », caractérisée par l'inaptitude à produire des logiciels « corrects » et fiables, dans le respect des budgets alloués et du temps imparti. Parmi les explications avancées, il y a non seulement la complexité inhérente aux processus de développement logiciel, mais également la façon dont les projets de développement de logiciels sont gérés. Il a été démontré dans la littérature que la gestion de ces projets pourrait être améliorée grâce au développement et l'exploitation efficiente de mesures et d'indicateurs adéquats. En effet, les mesures permettent de mieux maîtriser les projets de développement logiciels, grâce à l'identification, la quantification et le contrôle des principaux paramètres qui affectent le développement logiciel (effort, coût, qualité, productivité, etc.).

Considérées il y a quelques années maintenant comme une curiosité de quelques universitaires et compagnies dans l'industrie, les métriques dans le domaine du logiciel sont aujourd'hui une discipline bien établie, avec un nombre sans cesse grandissant de praticiens.

Le cours se propose de contribuer à l'élargissement de la communauté des praticiens en apportant une aide et un bagage de base aux étudiants qui en sont à leurs débuts dans la quantification du développement logiciel.

2. PLACE DU COURS DANS LE PROGRAMME

Qu'il s'agisse du programme de deuxième cycle en génie logiciel ou encore du programme de maîtrise en génie logiciel incluant un cheminement de type cours en technologies de l'information, le cours permettra à l'étudiant ou l'étudiante d'approfondir ses connaissances en matière de gestion des projets informatiques (contrôle des projets au moyen de mesures et d'indicateurs). Le cours est conçu de façon à intégrer des étudiants ou des étudiantes ayant différents niveaux de connaissances et d'expérience en matière de mesures des logiciels, en fournissant un cadre de référence uniforme pour l'étude des mesures des logiciels. Il est toutefois supposé que les étudiants ou les étudiantes sont familiers avec le processus de développement logiciel.

3. DESCRIPTEUR DU COURS

(Selon l'annuaire des activités pédagogique disponible sur le site Web de l'université)

Objectifs :

Décrire, classer, comparer et savoir utiliser les mesures et les indicateurs classiques du génie logiciel.

Justifier et planifier l'utilisation d'indicateurs et pouvoir anticiper les principaux problèmes potentiels reliés à leur exploitation.

Contenu :

Mesures et indicateurs dans le cycle de vie des systèmes informatiques; mesures et indicateurs de développement, de conception et d'analyse. Mesures et modèles de fiabilité. Cadre expérimental. Micro et macromodèles. Évaluation de modèles. Automatisation et exploitation des mesures : estimation et contrôle des projets, assurance qualité, mesure de la productivité.

4. OBJECTIFS SPÉCIFIQUES

À l'issue du cours l'étudiant ou l'étudiante devrait pouvoir :

- utiliser à bon escient la terminologie du domaine des métriques de logiciels dans ses échanges avec les experts du domaine (échanges verbaux ou écrits);
- analyser un texte scientifique portant sur les métriques de logiciels;
- connaître et appliquer les principales méthodes d'identification des aspects à mesurer, telles que le paradigme Goal-Question-Metrics;
- décrire les principaux types de modèles de mesure;
- connaître les éléments pertinents de la théorie de la mesure;
- mesurer la taille et la complexité du logiciel;
- réaliser des estimations de coût et d'effort pour des projets de développement de logiciels;
- décrire et utiliser les principales métriques de défauts;
- mesurer et prédire la fiabilité d'un système logiciel;
- mesurer le temps réponse et la disponibilité d'un système logiciel;
- mesurer la progression d'un projet de développement de logiciels;
- présenter correctement et efficacement les diverses métriques d'un produit, d'un projet ou d'un processus logiciel à la direction d'une entreprise.

5. PLANIFICATION HEBDOMADAIRE

SEMAINE	CONTENU DU COURS	LECTURES Laird & Brennan 2006	LECTURES Fenton et al. 2015	TRAVAUX	POIDS
1	Présentation du contenu du cours et entente d'évaluation Introduction aux mesures de logiciels La mesure (1ère partie) : • ISO 15939	chap. 1, 2	chap. 1		
2	La mesure (2ème partie) : • Théorie de la mesure	chap. 3	chap. 2	<i>Synthèse/analyse (proposition d'un article)</i>	
3	La mesure de la taille (1ère partie): • Mesures physiques • Mesures de la fonctionnalité	chap. 4	chap. 8		

4	La mesure de la taille (2ème partie) <ul style="list-style-type: none"> Suite de la mesure de la fonctionnalité 	chap. 4	chap. 8	<i>Synthèse/analyse no 1 (remise)</i>	15%
5	La mesure de la complexité L'estimation de l'effort : <ul style="list-style-type: none"> Estimation par l'expert « Benchmarks » Estimation par analogie Estimation par points d'approximation (points de fonction, etc.) 	chap. 5, 6	chap. 9	<i>Exercice de mesure (énoncé)</i>	
6	L'estimation de l'effort (suite): <ul style="list-style-type: none"> Estimation par points d'approximation (points de fonction, etc.) (suite) Modèles algorithmiques 	chap. 6			
7	Les mesures de défauts : <ul style="list-style-type: none"> Fautes et défaillances Dynamique des défauts Techniques et modèles de projection Aspects économiques de l'enlèvement des défauts 	chap. 7	chap. 7	<i>Exercice de mesure (remise)</i>	15%
8	La mesure et la prédiction de la fiabilité : <ul style="list-style-type: none"> Classes de sévérité des défaillances Intensité des défaillances Aspects économiques de la fiabilité Théorie de la fiabilité Modèles de la fiabilité Taux d'émergence des défaillances La fiabilité et l'architecture des systèmes 	chap. 8	chap. 11		
9	Temps de réponse et disponibilité	chap. 9		<i>Travail de session (évaluation formative, non notée) Rapport individuel sur le travail de session (évaluation formative, non notée)</i>	
10	Les mesures de projet : <ul style="list-style-type: none"> Jalons Patrons d'intégration du code Mesure de l'avancement des tests Mesure de l'identification des défauts 	chap. 10			
11	La mesure des processus : <ul style="list-style-type: none"> Mesure des activités Mesure de l'effort Mesure des activités cognitives 				
12	Implantation d'un programme de mesure		chap. 3, 5		
13	Aspects économiques et financiers de la mesure <ul style="list-style-type: none"> Coûts Bénéfices Mesures pour l'analyse de rentabilité	chap. 12			

14	Exposés (travail de session)			<i>Travail de session (exposés et remise des rapports finaux)</i>	35%
15	Examen final			<i>Examen final</i>	35%

6. APPROCHE PÉDAGOGIQUE PRÉCONISÉE

L'approche pédagogique qui permettra d'atteindre les objectifs visés par le cours est la suivante : Un enseignement hebdomadaire de deux cours magistraux, avec exemples et discussions en classe pour une période de trois heures chacun. L'étudiant ou l'étudiante devra compléter sa formation par des travaux individuels proposés, un projet de session en équipe et des lectures personnelles suggérées. Le cours pourra être adapté selon les besoins des étudiants et étudiantes, si nécessaire.

7. ÉVALUATION DE L'APPRENTISSAGE

DESCRIPTION SOMMAIRE	PONDÉRATION	PONDÉRATION INDIVIDUELLE	PONDÉRATION DE GROUPE
Synthèse et analyse d'article : <ul style="list-style-type: none"> <i>travail</i> individuel résumé d'un article scientifique fourni 	15%	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Exercice de mesure : <ul style="list-style-type: none"> <i>travail</i> individuel Utilisation d'une méthode de mesure 	15%	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Examen final : <ul style="list-style-type: none"> <i>travail</i> individuel <i>questions</i> de cours et exercices 	35%	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Travail de session : <ul style="list-style-type: none"> <i>par</i> équipe de 2 à 4 personnes <i>rapport</i> de synthèse d'environ 20 pages par équipe <i>présentation</i> orale de 10 à 15 minutes par équipe 	35%	<input type="checkbox"/>	<input checked="" type="checkbox"/>

NOTE : Pour la synthèse, l'étudiant(e) aura la possibilité de soumettre une seconde version visant à corriger les lacunes et faiblesses de la première, le cas échéant. L'étudiant(e) n'est pas obligé(e) de soumettre une seconde version pour le travail couvert. Cependant, s'il ou elle le fait alors les règles suivantes s'appliquent.

- Le professeur évaluera quantitativement une seconde version si celle-ci est soumise avant le début de la séance no 14. La note retenue sera la moyenne des notes initiale et courante.
- Le fait de soumettre une seconde version pour un travail couvert n'a aucune incidence sur la note attribuée aux autres travaux.
- L'examen final ne peut pas faire l'objet d'une seconde version.
- Une fois notés, le travail de session et le rapport individuel sur celui-ci ne peuvent pas faire l'objet d'une seconde version. Toutefois, le professeur pourrait fournir une évaluation formative pour les versions préliminaires de ces travaux lorsque remis avant le début de la séance no 9.

8. ADRESSE ÉLECTRONIQUE POUR LA REMISE DES TRAVAUX

Sauf indication contraire, les travaux doivent être envoyés à l'adresse suivante :
Evariste.Valery.Bevo.Wandji@USherbrooke.ca

9. BIBLIOGRAPHIE

- [1] Laird, L. M.; Brennan, M. Carol., *Software Measurement and Estimation: A Practical Approach*, IEEE Computer Society Press / Wiley-Interscience, 2006.
- [2] Université de Sherbrooke / CeFTI, *INF 721 – Mesures et indicateurs du génie logiciel – Trimestre: Été 2016*.
- [3] Fenton, N. E.; Bieman, J., *Software Metrics: A Rigorous and Practical Approach, Third Edition*, CRC Press (Taylor & Francis Group, LLC), Boca Raton, FL, 2015.
- [4] Abran, A., *Software Metrics and Software Metrology*, Hoboken, N.J. : Wiley; Los Alamitos, CA : IEEE Computer Society, 2010.
- [5] Fenton, N. E.; Pfleeger, S. L., *Software Metrics: A Rigorous Approach - 2nd ed.*, Boston, MA : PWS , c1997.
- [6] Kan, S., *Metrics and Models in Software Quality Engineering – 2nd ed.*, Montréal: Addison Wesley, 2003.
- [7] McConnell, S., *Code Complete – 2nd ed.*, Redmond, WA : Microsoft Press, 2004.
- [8] McConnell, S., *Software Estimation: Demystifying the Black Art*, Redmond, WA : Microsoft Press, 2006.
- [9] Robillard, P.N.; Kruchten, P., *Software Engineering Process with the UPEDU*, Montréal: Addison Wesley, 2003 (chap. 12).
- [10] Zuse, H., *Software Complexity - Measures and Methods*, De Gruyter Publisher, Berlin, New York, 1991.
- [11] Boehm, B. W. et al., *Software Cost Estimation with COCOMO II*, Prentice-Hall Publ., 2000.

Quelques articles

- [1] Boehm, B. W.; Fairley, R. E., *Software Estimation Perspectives*, IEEE Software, November/December 2000, pp. 22-26.
- [2] Jenner, M. S., "Automation of Counting of Functional Size Using COSMIC FFP in UML", *Proc. of the 12th International Workshop on Software Measurement*, October 7-9, 2002, Magdeburg, Shaker Publ., Aachen, pp. 43-51.
- [3] Anselmo, D.; Ledgard, H., *Measuring Productivity in the Software Industry*, Comm. of the ACM, 46(2003)11, pp. 121-125.

Quelques Sites Web

- ☞ <http://www.ukσμα.co.uk/>
- ☞ www.cosmicon.com
- ☞ www.ifpug.org
- ☞ <http://csse.usc.edu/tools/COCOMOII.php>
- ☞ <http://cmmiinstitute.com/>