UCL Université catholique de Louvain

# LINGI2252

*2013-2014*

# Software engineering : measures and maintenance

| 5.0 crédits | 30.0 h + 15.0 h | 1q |
|---|---|---|

| | |
|---|---|
| Enseignants: | Mens Kim ; |
| Langue d'enseignement: | Anglais |
| Lieu du cours | Louvain-la-Neuve |
| Ressources en ligne: | > http://icampus.uclouvain.be/claroline/course/index.php?cid=INGI2252 |
| Préalables : | -- programmation orienté-object et expérience avec des codes de taille moyenne (p.e. LSINF1121 : Algorithmique et structure de données)<br>-- avoir en parallèle ou précédemment une expérience de développement d'un logiciel de plus grande taille (p.e. LINGI2251 et LSINF2255/LINGI2255 : cours et projet de génie logiciel) |
| Thèmes abordés : | -- Les "bonnes pratiques" de la programmation orienté-objet;<br>-- Les techniques de réutilisation, application frameworks;<br>-- Les métriques et mesures de logiciels;<br>-- La gestion de versions: variantes, révisions, configurations;<br>-- La compréhension et rétro-ingénierie de logiciels;<br>-- La ré-ingénierie et la restructuration de logiciels;<br>-- L'utilisation de certains outils qui supportent les taches mentionnées ci-dessus. |
| Acquis d'apprentissage | Les étudiants ayant suivi avec fruit ce cours seront capables de<br>-- expliquer comment mesurer les qualités d'un logiciel (et notamment la qualité de la maintenabilité);<br>-- utiliser divers outils pour mesurer les qualités d'un logiciel et analyser de façon critique les résultats obtenus en les comparant avec ceux provenant d'autres outils et en inspectant le logiciel manuellement pour confirmer ou contredire les résultats obtenus;<br>-- expliquer la nature des problèmes rencontrés dans la maintenance de logiciels complexes et les processus impliqués;<br>-- mettre en oeuvre quelques approches-type pour favoriser la réutilisabilité et la maintenabilité;<br>-- programmer des applications maintenables et réutilisables en Smalltalk, un langage orienté-objet pur.<br>Les étudiants auront développé des compétences méthodologiques et opérationnelles. En particulier, ils ont développé leur capacité à<br>-- avoir un regard critique sur les résultats d'un outil automatisé;<br>-- réaliser une présentation convaincante;<br>-- rédiger un rapport de synthèse.<br><br>*La contribution de cette UE au développement et à la maîtrise des compétences et acquis du (des) programme(s) est accessible à la fin de cette fiche, dans la partie « Programmes/formations proposant cette unité d'enseignement (UE) ».* |
| Modes d'évaluation des acquis des étudiants : | L'évaluation du cours est basée sur les rapports et la présentation sur l'analyse d'uneétude de cas.<br>Chaque étudiant devra réaliser, par groupe de 2, un petit projet au cours du quadrimestre, avec des rapports intermédiaires et un rapport final à produire. La note finale se basera sur les résultats du projet, la qualité des rapports, et la présentation finale.<br>Conditions pour pouvoir participer à l'examen (= présentation finale) de janvier / juin : avoir soumis le(s) rapport(s) dans l'échéance indiquée par le professeur.<br>Pour l'examen de juin ou de septembre la note de janvier n'interviendra plus. Or, l'étudiant devra finaliser,individuellement, un nouveau projet de même ampleur que pour la session de janvier. |
| Méthodes d'enseignement : | Les aspects théoriques sont introduits dans les séances de théorie.<br>Ils sont mis en pratique pendant les travaux pratiques sur machine dans l'une des salles informatiques ou sur les ordinateurs portables des étudiants. Un seul logiciel est développé tout au long des différentes sessions pratiques et les techniques enseignées dans le cours théorique seront illustrées sur ce logiciel.<br>Une étude de cas sera réalisée sur un logiciel à analyser en utilisant ces techniques, par groupe de 2. Les étudiants sont invités à analyser les qualités de cette application, et notamment sa maintenabilité, et à suggérer des améliorations possibles . Les résultats de leur travail seront expliqués dans des rapports intermédiaires et final et lors d'une présentation.<br>Les étudiants sont invités à rédiger les rapports et à présenter leurs travaux en anglais. |
| Contenu : | La théorie introduite dans les exposés magistraux sera mise en pratique pendant les travaux pratiques qui se dérouleront sur ordinateur. Un logiciel sera développé au cours des différentes séances des travaux pratiques, et les différentes techniques apprises seront illustrées sur ce logiciel. L'évaluation de ce cours est un projet ou les étudiants doivent appliquer les techniques apprises sur un autre logiciel au choix, afin d'analyser les qualités de ce logiciel et notamment sa maintenabilité. |

| | |
|---|---|
| Bibliographie : | La théorie du cours se base sur plusieurs livres :<br>--<br>A. Black, S. Ducasse, O. Nierstrasz, "Pharo By Example", 2009.<br>--<br>K.Beck, "Smalltalk Best Practice Patterns", Prentice Hall, 1997.<br>--<br>M. Fowler, "Refactoring: Improving the Design of Existing Code", Addison-Wesley, 2001.<br>--<br>A. Riel, "Object-Oriented Design Heuristics", Addison-Wesley, 1996.<br>--<br>N.E. Fenton & mp; S.L. Pfleeger, "Software Metrics: A Rigorous and Practical Approach", 2nd edition, Thomson Computer Press,1996.<br>Les transparents des cours magistraux ainsi que les énoncés des séances pratiques et d'autres informations pratiques relatives au cours seront accessibles en ligne. |
| Autres infos : | |
| Cycle et année d'étude: : | > Master [120] : ingénieur civil en informatique<br>> Master [120] en sciences informatiques |
| Faculté ou entité en charge: | INFO |