

5.00 crédits

30.0 h + 15.0 h

Q1

Enseignants	Mens Kim ;
Langue d'enseignement	Anglais > Facilités pour suivre le cours en français
Lieu du cours	Louvain-la-Neuve
Thèmes abordés	<p>Alors que de nombreux cours de génie logiciel se concentrent sur la création de nouveaux systèmes logiciels à partir de zéro, dans la pratique industrielle, les développeurs de logiciels sont souvent confrontés à des systèmes logiciels déjà existants qui doivent être maintenus, réutilisés ou évolués. Cela nécessite des compétences spécifiques pour comprendre la conception et la mise en œuvre d'un système existant et quelles composantes doivent être modifiées, pour construire des systèmes logiciels plus faciles à maintenir et pour concevoir des systèmes, dès le départ, tenant compte de leur réutilisation et leur évolution.</p> <p>Ce cours étudiera une variété de techniques, d'outils et de méthodes pour aider à construire des systèmes logiciels plus faciles à maintenir, réutiliser et faire évoluer, tels que :</p> <p>Préliminaires et définitions :</p> <ul style="list-style-type: none"> - raisons et problèmes de la maintenance et l'évolution logicielle - définitions, différences et types de maintenance et d'évolution logicielle - dette technique - lois de l'évolution logicielle <p>Modélisation du domaine :</p> <ul style="list-style-type: none"> - lignes de produits logicielles - analyse de domaine - modélisation de features - points communs et variabilités - diagrammes de features <p>Réutilisation du logiciel :</p> <ul style="list-style-type: none"> - définition et raisons de la réutilisation logicielle - techniques de réutilisation et conception pour la réutilisation - techniques orientées objet pour la réutilisation et la maintenabilité - frameworks applicatifs orientés objet <p>Maintenance et évolution logicielle :</p> <ul style="list-style-type: none"> - Meilleures pratiques de programmation et normes de codage - Refactoring et reengineering du code - Les mauvais codes - Les motifs de conception - Principes et heuristiques de conception <p>Une étude de cas industrielle</p>
Acquis d'apprentissage	<p>A la fin de cette unité d'enseignement, l'étudiant est capable de :</p> <p>Eu égard au référentiel AA du programme « Master ingénieur civil en informatique », ce cours contribue au développement, à l'acquisition et à l'évaluation des acquis d'apprentissage suivants :</p> <ul style="list-style-type: none"> • INFO1.1 , INFO1.3 • INFO2.5 • INFO5.5 <p>Eu égard au référentiel AA du programme « Master [120] en sciences informatiques », ce cours contribue au développement, à l'acquisition et à l'évaluation des acquis d'apprentissage suivants :</p> <p>1</p> <ul style="list-style-type: none"> • SINF1.M3 • SINF2.5 • SINF5.5 <p>Les étudiants ayant suivi avec fruit ce cours seront capables de</p> <ul style="list-style-type: none"> • Comprendre les difficultés de développement de code dans un contexte de changement, par opposition au développement à partir de rien • Évaluer l'impact d'une demande de modification d'un produit existant de taille moyenne.

	<ul style="list-style-type: none"> • Décrire les techniques, les idiomatiques de codage et d'autres mécanismes de mise en 'uvre des conceptions qui sont plus maintenables. • Comprendre comment les modèles de conception peuvent améliorer la conception d'un système de logiciel. • Refactoriser une implémentation logicielle existante pour améliorer certains aspects de sa conception. • Identifier les principaux problèmes liés à l'évolution d'un logiciel et expliquer leur impact sur le cycle de vie du logiciel. • Discuter des avantages et inconvénients des différents types de réutilisation de logiciels.
<p>Modes d'évaluation des acquis des étudiants</p>	<p>Dans le cadre de ce cours, les étudiant-es sont évalué-es par :</p> <ul style="list-style-type: none"> • une évaluation continue certificative du projet, qui se repose sur 3 missions différentes, réalisés en binôme, à délivrer pendant le quadrimestre; • une présentation en binôme, réalisé en session, où les missions réalisés pendant le quadrimestre sont mis en perspective avec les concepts vus dans le cours; • un examen individuel sur la matière du cours, réalisé en session. <p>Pour constituer la note finale, la pondération donnée à l'évaluation continue est :</p> <ul style="list-style-type: none"> • 10% de la note finale pour la participation active lors des séances pratiques • 50% pour les 3 missions réalisées en binôme pendant le quadrimestre <p>et la pondération de l'examen est :</p> <ul style="list-style-type: none"> • 20% pour l'examen écrit (individuel) • 20% pour la présentation (en binôme) <p>La note relative à l'évaluation continue peut être individualisée en fonction de l'implication de l'étudiant-e au sein de son binôme pendant le quadrimestre (présence aux activités, participation active aux missions intermédiaires et aux travaux réalisés et présentés). Les travaux donnant lieu à la note d'évaluation continue ne peuvent être refaits en seconde session; la note d'évaluation continue acquise en première session est conservée en cas de seconde session. Seulement l'examen écrit individuel et la présentation en binôme mettant en perspective les missions avec les concepts vus dans le cours pourront être représentés en seconde session.</p> <p>Comme le cours même, toutes les évaluations de ce cours seront en anglais.</p>
<p>Méthodes d'enseignement</p>	<ul style="list-style-type: none"> • Cours magistraux couvrant les différents sujets du cours. • Travaux pratiques pour appliquer les concepts de manière pratique. • Missions pour expérimenter les problèmes liés et les solutions pour développer et faire évoluer un système logiciel maintenable et réutilisable. • Présentation par une entreprise invitée pour illustrer certains des sujets de cours appliqués dans la pratique industrielle. • Optionnellement, des sessions sur un thème lié à la recherche dans le domaine de la maintenance, de la réutilisation et de l'évolution logicielle.
<p>Contenu</p>	<p>Le cours couvre une variété de techniques, outils et méthodes pour aider la construction de systèmes logiciels qui sont plus faciles à comprendre, maintenir, réutiliser et évoluer:</p> <ul style="list-style-type: none"> • Définitions et motivations préliminaires • La modélisation de domaine • La réutilisation logicielle • Le mauvais code • Le refactoring • Les patrons de conception • Les heuristiques de conception • Les frameworks applicatifs • Une étude de cas industrielle • Sessions supplémentaires liées à des sujets de recherche sélectionnés
<p>Ressources en ligne</p>	<p>Site du cours Moodle</p> <p>Les transparents de cours ainsi que d'autres informations pertinentes et pratiques relatives au cours seront accessibles sur Moodle. La même plate-forme sera également le moyen de communication entre l'enseignant et les étudiants.</p>
<p>Bibliographie</p>	<p>French</p> <p>Compte tenu de la variété des sujets abordés, ce cours ne suivra pas un seul livre de référence, mais sera basé sur du matériel provenant de nombreuses sources différentes. Les slides de cours seront le matériel de référence principale pour ce cours et des pointeurs vers des lectures supplémentaires seront fournis par la plate-forme de cours en ligne.</p> <p>English</p> <p>Given the variety of topics covered, this course will not follow a single textbook but is based on material from many different sources. As such, the course slides will be the main reference material for this course and pointers to additional reading material will be provided through the online course platform.</p>

Autres infos	<p>Même si un logiciel de bonne qualité peut être plus facile à maintenir et à faire évoluer, les techniques d'assurance de la qualité des logiciels ne seront pas abordées explicitement dans ce cours car ils font l'objet d'un cours distinct sur l'assurance qualité du logiciel [LINFO2251]</p> <p>Préalables:</p> <ul style="list-style-type: none"> • Avoir une bonne connaissance et expérience avec les concepts de la programmation orientée objet, les algorithmes et les structures de données. • Avoir une expérience préalable ou simultanée avec le développement d'un logiciel de taille moyenne à grande.
Faculté ou entité en charge:	INFO

Programmes / formations proposant cette unité d'enseignement (UE)				
Intitulé du programme	Sigle	Crédits	Prérequis	Acquis d'apprentissage
Master [120] : ingénieur civil en informatique	INFO2M	5		
Master [120] en sciences informatiques	SINF2M	5		