

Enseignants:	Van Lamsweerde Axel ;
Langue d'enseignement:	Anglais
Lieu du cours	Louvain-la-Neuve
Préalables :	<ul style="list-style-type: none"> -- logique mathématique telle qu'enseignée dans le cours INGI1101 -- maîtrise de la programmation orienté-objet, de l'algorithmique et des structures de données telle que visée par le cours SIN1121 -- avoir participé à la réalisation d'un projet logiciel de taille réduite (par exemple SIN1124).
Thèmes abordés :	<ul style="list-style-type: none"> -- Cycle de vie d'un projet logiciel: aspects statiques (les produits) et dynamiques (les procédés). -- Introduction à l'analyse des besoins: modélisation semi-formelle, spécification et analyse des exigences auxquelles doit répondre le logiciel. -- Introduction à la conception d'architectures logicielles: hiérarchisation, modularisation; styles et patterns architecturaux. -- Spécification d'un module vu comme unité de travail. -- Validation d'un logiciel: conception de jeux de test: black-box, white-box et d'intégration. -- Documentation des décisions prises au cours des différentes étapes du cycle de vie.
Acquis d'apprentissage	<p>Les étudiants ayant suivi avec fruit ce cours seront capables de</p> <ul style="list-style-type: none"> -- développer un produit logiciel de grande taille en spécifiant les exigences logicielles, en exploitant une méthodologie de conception de logiciel -- créer, évaluer et mettre en oeuvre un plan de test pour un logiciel de grande taille -- réagir de manière adéquate quand des problèmes inhérents à de gros projets logiciels apparaissent -- utiliser avec rigueur des outils ou des concepts spécifiques pour gérer les étapes critiques d'un grand projet de développement logiciel en gardant à l'esprit l'objectif de qualité durant l'ensemble du processus de développement -- développer des spécifications en se basant sur une description de la situation, des résultats d'entrevues avec les clients et les futurs utilisateurs du logiciel (l'étape d'élucidation des exigences étant grandement simplifiée) -- concevoir une architecture logicielle -- concevoir de «bons» jeux de test -- documenter le processus de développement suivi. <p>Les étudiants auront développé des compétences méthodologiques et opérationnelles. En particulier, ils auront développé leur capacité à</p> <ul style="list-style-type: none"> -- travailler en équipe en démontrant, par la participation à un projet d'équipe, les éléments centraux de l'esprit d'équipe et la gestion d'équipe, -- rédiger des rapports montrant de manière claire l'état d'avancement d'un projet et justifiant les choix effectués en les fondant sur une argumentation rigoureuse. <p><i>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) ».</i></p>
Modes d'évaluation des acquis des étudiants :	<ul style="list-style-type: none"> -- quizz en cours de quadrimestre (30%) -- rapports intermédiaires (cahier de charges, architecture du logiciel, formalisation des spécifications et jeux de test) et rapport final, démonstration du logiciel réalisé (40%) -- participation individuelle aux réunions de groupe avec l'assistant (30%)
Méthodes d'enseignement :	<p>Le projet est fortement couplé au cours INGI2251. Il consiste à développer un logiciel de grande taille, par équipes, selon les techniques étudiées dans le cours INGI2251.</p> <p>Le cours INGI2251 est organisé de manière intensive, en début de quadrimestre, pour permettre un démarrage rapide du projet, puis de façon plus épisodique, en fonction des besoins des différentes étapes de développement.</p> <p>Le projet consiste à développer une application en vraie grandeur, typique d'un produit logiciel industrielle, dans des conditions de travail semi-professionnelles</p> <ul style="list-style-type: none"> -- Travail en équipe de 6-8 développeurs (nécessaire pour mener à bien un gros projet), supervisé par un chef de projet (chercheur) -- Gestion de la traçabilité entre étapes. -- Echange d'unités de travail entre développeurs (spécification d'un composant par A, design de ce composant par B, conception des jeux de test par C, implémentation par D). -- Gestion du parallélisme entre développeurs (à maximiser) et des interactions (à minimiser) -- Rencontre hebdomadaire avec le chef de projet (un chercheur de l'institut): présentation de l'état d'avancement et difficultés, évaluation d'options alternatives, proposition de répartition du travail au sein de l'équipe <p>Pour les phases principales du projet, travail préalable en équipe sur un mini-projet (application de taille réduite), suivi d'une correction avec les assistants.</p> <p>A différentes étapes, quizz individuel basé sur ce mini-projet, pour s'assurer que chaque étudiant possède le bagage nécessaire pour contribuer de manière effective à l'avancement du travail de son équipe.</p>

Contenu :	<p>Le projet consiste à développer une application en vraie grandeur, typique d'un produit logiciel industrielle, dans des conditions de travail semi-professionnelle</p> <p>Exemples de systèmes ainsi développés: système de vote électronique, dispatching d'ambulances, organisateur de réunions, gestion de ressources pour cybernomades, gestion hospitalière, gestion de bibliothèque, gestion des programmes d'étudiants, ...</p>
Bibliographie :	<p>-- transparents du cours INGI2251 en ligne</p> <p>-- énoncés et échéanciers du projet en ligne</p> <p>-- support partiel: « Requirements engineering, From System Goals to UML Models to Software Specification », A. van Lamsweerde, Wiley, 2009</p>
Autres infos :	<p>-- quizz en cours de quadrimestre</p> <p>-- rapports intermédiaires (cahier des charges, architecture du logiciel, formalisation des spécifications et jeux de test) et rapport final, démonstration du logiciel réalisé</p> <p>-- examen écrit en session</p>
Cycle et année d'étude :	<p>> Master [120] en sciences informatiques</p> <p>> Master [60] en sciences informatiques</p>
Faculté ou entité en charge:	<p>INFO</p>