MD-2045, CHIŞINĂU, STR. STUDENȚILOR, 9/7, TEL: 022 50-99-01 | FAX: 022 50-99-05, www.utm.md

TEST ET VERIFICATION DES LOGICIELS

1. Informations sur l'unité de cours / module

Faculté	Ordinateurs, Informatique et Microélectronique				
Chaire/département	Filière Francophone Informatique, dép. Génie Logiciel et Automatique				
Cycle d'études	Études supérieures, Licence - cycle I				
Programme d'études	526.2 Technologies de l'information				
Année d'étude	**Semestre**	**Type d'évaluation**	**Catégorie formative**	**Catégorie d'option**	**Crédites ECTS**
III (enseignement à plein temps);	5	E	S – Unité de cours de spécialité	O - Unité de cours obligatoire	3

2. Estimation du temps total

Nombre total d'heures dans le programme	Dont				
	Heures dans la salle de cours		Travail individuel		
	Cours	Travaux pratique/dirigés	Projet d'année	Étude du matériel théorique	Préparation de l'application
120	30	30/-		30	30

3. Prérequis pour l'accès à l'unité de cours/module

Selon le programme d'études	Modélisation des systèmes logiciels (UML); Architecture des systèmes logiciels; Réseaux informatiques; POO, technologies Web, diverses langages de programmation: C ++, C #, Java; Bases de données.
Selon les compétences	Connaissances et compétences de conception et de développement d'applications en différentes langages de programmation.

4. Conditions de déploiement le processus éducatif pour

Cours	Pour présenter le matériel théorique en classe, il faut un tableau noir, un projecteur et un ordinateur.
Travaux pratique/dirigés	Les étudiants rédigeront des rapports selon les conditions formulées dans les indications méthodiques. La durée du soutien d'un travail pratique est une semaine après l'achèvement. La soumission tardive du document est pénalisée : -1 point pour une semaine de retard.

5. Compétences spécifiques accumulées

Compétences professionnelles	✓ Utilisation de critères et méthodes d'évaluation du processus de développement du système en termes de qualité et de performance. ✓ Développement et mise en œuvre de logiciels pour des problèmes concrets dans divers domaines.

Compétences transversales	CT1. Appliquer les principes, les normes et les valeurs de l'éthique professionnelle.
	CT2. Identifier, décrire et gérer les activités organisées en équipe ; développement des capacités de communication et de collaboration, ainsi que d'assumer les différents rôles (exécution et leadership)

	CT3. Faire preuve de l'esprit d'initiative et d'action pour mettre à jour les connaissances professionnelles, économique et de la culture organisationnelle

6. Objectifs de l'unité de cours / module

Objectif général	**Fournir des connaissances sur les techniques de vérification des programmes, méthodes de validation et vérification, coût des bugs : coûts économique, humains ; nécessité d'assurer la qualité des logiciels.**
Objectifs spécifiques	**Développer des stratégies de test, choisir des méthodes d'essai (test) complémentaires ayant comme objectif le développement des logiciels fiables.**

7. Contenu de l'unité de cours / module

Thématique des activités didactiques	Nombre d'heures	
	enseignement à temps plein	enseignement à temps partiel
Thème des cours		
T1. Introduction aux tests et à la vérification des produits du programme. Le rôle des tests. Présentation des erreurs logicielles à travers l'histoire.	2	
T2. Processus de test de logiciel. Organisations: CTFL, ISTQB, REQB. Certifications.	2	
T3. Méthodes d'essai. Types de tests. Essais fonctionnels et structurels.	2	
T4. Gestion des exceptions et des erreurs. Stratégies globales de test. Outils de gestion des tests. Documentation des tests.	2	
T5. Test de logiciel orienté objet. Méthode d'essai hiérarchique.	2	
T6. Stratégies de test pour les environnements modernes: cloud computing, applications Web et mobiles.	2	
T7. Automatisation des tests. Le processus d'automatisation. Exemples de logiciels libres et autorisés disponibles pour l'automatisation des tests. Des exemples de fonctions qui peuvent être testées automatiquement. Exemple d'automatisation d'un test.	2	
T8. Test de logiciel avec Selenium. Tests fonctionnels. Test de bout en bout. Formulation de scénarios de tests.	2	
T9. Les tests fondés sur des modèles (automates finis, UML, chaînes de Markov). Critères de génération: couverture satisfaisante du modèle tous les états / transitions; combinaisons de transitions k consécutives.	2	
T10. Tests de bout en bout. Test de la fonctionnalité critique de l'application: communication avec d'autres systèmes, interfaces, bases de données, réseaux et autres applications.	2	
T11. Sécurité du logiciel. Tests de sécurité pendant les étapes de développement d'un produit logiciel.	2	
T12. Introduction aux méthodes formelles. Vue d'ensemble - pourquoi les méthodes formelles sont-elles utilisées et comment? analyse abstraite - démonstration automatique des propriétés sur le code réel. Modèle de vérification - Vérification des propriétés temporelles des distributeurs automatiques. Hoare Logic - Démonstration de propriétés génériques sur un programme complet. Démonstration interactive de théorèmes. Conclusions.	2	
T13. Intégration des méthodes formelles. Méthodes formelles pour les systèmes critiques. Problème général d'intégration des méthodes formelles. Le bon fonctionnement des systèmes. Importance et difficulté.	2	

T14. Assurer et contrôler la qualité du logiciel. Normes ISO 9000, ISO 9001 et CMM.	2	
T15. Récapitulatif de l'information sur l'utilisation de critères et méthodes pour évaluer le processus de conception des systèmes en termes de qualité et de performance.	2	
Total des cours:	30	

Thématique des activités didactiques	Nombre d'heures	
	enseignement à temps plein	enseignement à temps partiel
Thèmes des travaux pratiques		
TP1 Spécification des exigences logicielles. Création des spécifications - spécification de ce que le système doit faire et les contraintes pour le développement.	2/2	
TP 2. Définition des cas de test. Elaboration des scenarii de test.	2/2	
TP 3. Vérification et Validation (Test fonctionnel). Génération du rapport d'essai.	2/2	
TP 4. Tests de logiciel avant de la livraison (tests non fonctionnels): les tests de compatibilité; test d'endurance (test de trempage); test de charge; test de localisation; test de performance; tests de sécurité; les tests de régression et ainsi de suite.	2/2	
TP 5. Automatisation des tests (avec Selenium).	2/2	
TP6. Utilitaire de résolution des problèmes liés à la maintenance des logiciels ou progiciels.	2/2	
TP7. Soutenance du TP6 et d'autres travaux qui n'ont pas été soumis à temps.	3/3	
Total des travaux pratiques:	15/15	

8. Références bibliographiques

Principales	1. Paul Ammann, Jeff Offutt, Introduction to Software Testing, Cambridge University Press, 346 p., 2008. 2. Anne Mette Jonassen Hass, Guide to Advanced Software Testing, Artech House, 427 p., 2008. 3. Roger S. Pressman, Bruce Maxim, Software Engineering: A Practitioner's Approach , 8th Edition, McGraw-Hill Education, 976 p., 2014
Supplémentaires	- http://www.istqb.org/certification-path-root/foundation-level/foundation-level-content.html - https://www.astqb.org/get-certified/istqb-syllabi-the-istqb-software-tester-certification-body-of-knowledge/

Evaluation

Actuelle		Projet d'année	Examen final
Attestation 1	Attestation 2		
30%	30%		40%
Normes de rendement minimum			
Présence et activité aux cours et travaux pratiques; Obtenez le score minimal de "5" pour chacune des attestations et des travaux pratiques; Démonstration de l'assimilation des informations fournies pendant le cours et des compétences pour dessiner les diagrammes nécessaires à la conception d'un produit de programme à l'examen final.			