

Semestre 2 **M2: Tronc Commun**

- Test Logiciel

Matière N° 6	
Intitulé de la Matière	Test Logiciel
Enseignant de la matière	Fomekong Evaris
Code	
Volume Horaire	20h
Crédits	
Niveau	M2
Semestre	2
Prérequis	Fondamentaux Programmation, Base Genie Logiciel, Bases de données, Architecture Logiciel
Description du contenu de la matière	Ce cours de test logiciel est conçu pour les étudiants en Master 2 et se déroulera sur quatre séances de 4 heures chacune. Le programme couvrira les aspects théoriques et pratiques du test logiciel, en mettant l'accent sur les méthodologies modernes, les outils et les meilleures pratiques. Les étudiants apprendront à concevoir, à exécuter et à gérer des tests logiciels pour garantir la qualité et la fiabilité des applications.
Objectif général	L'objectif général de ce cours est d'équiper les étudiants avec les connaissances et compétences nécessaires pour mener à bien des activités de test logiciel dans un contexte professionnel. Les participants apprendront à appliquer des méthodologies de test systématiques pour identifier les défauts dans le logiciel, améliorer la qualité des produits et réduire les risques associés aux défaillances logicielles.
Objectifs spécifiques (Savoirs et savoir-faire envisagés)	<ol style="list-style-type: none"> 1. Comprendre les Fondamentaux du Test Logiciel : Les étudiants seront capables d'expliquer l'importance du test logiciel et de décrire les différents types de tests ainsi que leur rôle dans le cycle de vie du développement logiciel. 2. Maîtriser les Techniques de Conception de Tests : Les participants apprendront à concevoir des tests efficaces en utilisant diverses techniques et à gérer des cas de test pour maximiser la couverture du code. 3. Utiliser des Outils de Test Modernes : Les étudiants acquerront une expérience pratique avec des outils de test logiciel courants, leur permettant de mettre en place et d'exécuter des tests automatisés.

Syllabus des cours formation M2

	<p>4. Développer des Stratégies de Test : Les participants seront capables de développer des stratégies de test globales, de gérer le cycle de vie des défauts logiciels et d'utiliser des métriques pour évaluer l'efficacité des tests.</p> <p>5. Appliquer les Connaissances dans des Projets Réels : En fin de cours, les étudiants devraient être capables d'appliquer les techniques, outils et stratégies apprises pour tester des applications logicielles dans des environnements réels.</p>
Contenu du cours	
Lectures	Sujets Abordés
<i>Lecture 1: Introduction au Test Logiciel</i>	<ul style="list-style-type: none"> ❖ Principes fondamentaux du test logiciel. ❖ Types de tests : unitaires, d'intégration, de système, d'acceptation. ❖ Cycle de vie du développement logiciel et intégration du test.
<i>Lecture 2: Techniques de Conception de Tests</i>	<ul style="list-style-type: none"> ❖ Techniques de boîte noire, de boîte blanche, et basées sur l'expérience. ❖ Gestion des cas de test et critères de couverture. ❖ Utilisation de tests automatisés.
<i>Lecture 3: Contrôle Continu</i>	<ul style="list-style-type: none"> ❖ Evaluation ❖ Correction
<i>Lecture 4: Outils et Environnements de Test</i>	<ul style="list-style-type: none"> ❖ Présentation des principaux outils de test (JUnit, Selenium, Postman, etc.). ❖ Configuration d'un environnement de test. ❖ Automatisation des tests et intégration continue.
<i>Lecture 5: Stratégies de Test et Gestion des Défauts</i>	<ul style="list-style-type: none"> ❖ Élaboration de stratégies de test efficaces. ❖ Suivi et gestion des défauts. ❖ Évaluation de la qualité et métriques de test. ❖ Examen Final
Approches pédagogiques	<p>Ce cours est basé sur une approche pédagogique actives. Elle vise à rendre l'étudiant acteur des savoirs qu'il construit à travers des situations de recherche et d'expérimentations, seul ou en groupe. En ce sens, elles valorisent l'autonomie. Il s'agit globalement de :</p> <ul style="list-style-type: none"> ❖ Faire par soi-même, expérimenter, faire

Syllabus des cours formation M2

	<p>l'expérience directe.</p> <ul style="list-style-type: none"> ❖ Manipuler, réaliser, produire, « agir », apprendre en faisant (learning by doing). ❖ Pratiquer le « tâtonnement expérimental », tester, se tromper (droit à l'erreur), trouver des solutions singulières aux problèmes posés. ❖ Participer, s'impliquer, s'exprimer, donner son avis. ❖ Choisir, décider ou participer aux prises de décision.
Méthode d'évaluation	<ul style="list-style-type: none"> ❖ Contrôle(s) continu(s) (TD ou TP) - 50% de la note finale ❖ Examen final ou Projet Final – 50% de la note finale
Lectures recommandées	<ul style="list-style-type: none"> ❖ "Test Driven Development (TDD) : Développez avec les tests" par <i>James Grenning</i>. ❖ "Tests unitaires en Java : Le développement guidé par les tests" par <i>Jeff Langr</i>. ❖ "Automatisation des tests logiciels en pratique" par <i>Bruno Legeard et Fabrice Bouquet</i>. ❖ "Clean Code: A Handbook of Agile Software Craftsmanship" by <i>Robert C. Martin</i>. ❖ "Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation" by <i>Jez Humble and David Farley</i>. ❖ "Accelerate: The Science of Lean Software and DevOps: Building and Scaling High Performing Technology Organizations" by <i>Nicole Forsgren, Jez Humble, and Gene Kim</i>.