

## Syllabus de formation

---

## Master 2

## Semestre 2 M2: Tronc Commun

### - Test Logiciel

<b>Matière N° 6</b>	
<b>Intitulé de la Matière</b>	Test Logiciel
<b>Enseignant de la matière</b>	Fomekong Evaris
<b>Code</b>	
<b>Volume Horaire</b>	20h
<b>Crédits</b>	
<b>Niveau</b>	M2
<b>Semestre</b>	2
<b>Prérequis</b>	Fondamentaux Programmation, Base Genie Logiciel, Bases de données, Architecture Logiciel
<b>Description du contenu de la matière</b>	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.
<b>Objectif général</b>	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.
<b>Objectifs spécifiques (Savoirs et savoir-faire envisagés)</b>	<ol style="list-style-type: none"> <li>1. <b>Comprendre les Fondamentaux du Test Logiciel</b> : 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.</li> <li>2. <b>Maîtriser les Techniques de Conception de Tests</b> : 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.</li> <li>3. <b>Utiliser des Outils de Test Modernes</b> : 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.</li> </ol>

## Syllabus des cours formation M2

	<p>4. <b>Développer des Stratégies de Test</b> : 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. <b>Appliquer les Connaissances dans des Projets Réels</b> : 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>
<b>Contenu du cours</b>	
<b>Lectures</b>	<b>Sujets Abordés</b>
<i>Lecture 1: Introduction au Test Logiciel</i>	<ul style="list-style-type: none"> <li>❖ Principes fondamentaux du test logiciel.</li> <li>❖ Types de tests : unitaires, d'intégration, de système, d'acceptation.</li> <li>❖ Cycle de vie du développement logiciel et intégration du test.</li> </ul>
<i>Lecture 2: Techniques de Conception de Tests</i>	<ul style="list-style-type: none"> <li>❖ Techniques de boîte noire, de boîte blanche, et basées sur l'expérience.</li> <li>❖ Gestion des cas de test et critères de couverture.</li> <li>❖ Utilisation de tests automatisés.</li> </ul>
<i>Lecture 3: Contrôle Continu</i>	<ul style="list-style-type: none"> <li>❖ Evaluation</li> <li>❖ Correction</li> </ul>
<i>Lecture 4: Outils et Environnements de Test</i>	<ul style="list-style-type: none"> <li>❖ Présentation des principaux outils de test (JUnit, Selenium, Postman, etc.).</li> <li>❖ Configuration d'un environnement de test.</li> <li>❖ Automatisation des tests et intégration continue.</li> </ul>
<i>Lecture 5: Stratégies de Test et Gestion des Défauts</i>	<ul style="list-style-type: none"> <li>❖ Élaboration de stratégies de test efficaces.</li> <li>❖ Suivi et gestion des défauts.</li> <li>❖ Évaluation de la qualité et métriques de test.</li> <li>❖ <b>Examen Final</b></li> </ul>
<b>Approches pédagogiques</b>	<p>Ce cours est basé sur une approche pédagogique <b>actives</b>. 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"> <li>❖ Faire par soi-même, expérimenter, faire</li> </ul>

## Syllabus des cours formation M2

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