

 UNIVERSITE AMADOU MAHTAR MBOW DE DAKAR Plus qu'une formation, un avenir à construire	UNIVERSITE AMADOU MAHTAR MBOW	
	Nom de la structure :	Polytech Diamniadio
	Filière	Master en Ingénierie des Systèmes d'Information et des Données (MISID)
PLAN DE COURS		

Domaine : Sciences et Technologies			Mention :		Spécialité : Informatique	
Intitulé de l'UE : Développement d'applications 1					Code : DAP2200	
Statut : Obligatoire					Crédit : 6	
Intitulé de l'EC : Qualité logicielle					Code : 3DAP2200	
Modalité : En Présence				Niveau : Master 1		
Année académique	Horaire				Semestre	Coefficient
2023 – 2024	CM	TD	TP	TPE	2	2
	8	0	12	10		
Auteur/Contacts :					Date de la Version : 2024	

I) Justification de l'EC

La qualité logicielle est un domaine crucial dans le développement moderne de logiciels, notamment avec l'augmentation des attentes des utilisateurs et la complexité croissante des systèmes. Ce cours de qualité logicielle permet aux étudiants de comprendre les principes de base et les pratiques modernes pour améliorer la qualité du code, des processus de développement et des produits finaux. La mise en place de processus de qualité et l'utilisation des outils adéquats sont des compétences essentielles pour réussir dans l'industrie, surtout dans des contextes Agile, DevOps ou de développement continu.

L'objectif de cet enseignement est de permettre aux étudiants de comprendre les enjeux liés à la qualité des logiciels et de les former aux bonnes pratiques nécessaires à l'amélioration de la performance, de la sécurité et de la maintenabilité des logiciels.

II) Conditions requises

Prérequis :

- 1 Connaissances de base en génie logiciel : cycles de développement, gestion de projets.
 - 2 Maîtrise des langages de programmation courants (Java, Python, C++, etc.).
 - 3 Connaissances de base en tests logiciels et développement orienté objet.
- 3.1 Outils didactiques : Syllabus ; Supports de cours, Vidéo projecteur

III) Intentions pédagogiques

Compétences :

1	1. OG 1 : Comprendre les concepts fondamentaux de la qualité logicielle	
	OS 1.1 : Maîtriser les principes et modèles de qualité logicielle.	
	OS 1.2 : Appréhender les processus et outils de gestion de la qualité dans le développement logiciel.	
	OS 1.3 : Être capable de définir et de mesurer des critères de qualité d'un logiciel	
2	2. OG 2 : Maîtriser les outils et techniques pour assurer la qualité logicielle dans les projets	
	OS 2.1 : Utiliser des outils d'analyse statique et dynamique du code pour identifier les défauts et mesurer la qualité	
	OS 2.2 : Automatiser les tests logiciels et intégrer les pratiques d'Assurance Qualité dans un processus de développement agile.	
	OS 2.3 : Mettre en place des pipelines d'intégration continue pour assurer la qualité sur le long terme.	
3	3. OG 3 : Intégrer la qualité dans le cycle de vie complet du développement logiciel	
	OS 3.1 : Appliquer des pratiques de qualité à chaque étape du cycle de vie du développement logiciel.	
	OS 3.2 : Assurer la sécurité, la performance et la fiabilité des logiciels dans des environnements complexes.	
	OS 1.3 : Réaliser un audit complet de la qualité logicielle et proposer des améliorations.	

	OS 1.4 : Contribuer à l'amélioration continue des processus de développement dans un contexte agile ou DevOps.
--	--

IV) Contenu l'EC

Résumé de l'EC :

Ce cours de **qualité logicielle** a pour objectif d'initier les étudiants à l'importance de la qualité dans le cycle de vie d'un logiciel. L'accent sera mis sur l'analyse de la qualité du code, la mise en place d'outils de test automatisés, et l'utilisation des bonnes pratiques en matière de gestion des défauts. Les étudiants apprendront à intégrer des pratiques de qualité dans des environnements de développement Agile, DevOps, ou en utilisant des méthodologies classiques. Ils seront également formés à l'utilisation d'outils de mesure de la qualité, à la gestion des risques de sécurité et à la gestion de la performance des logiciels.

Le cours comprendra une combinaison de théorie et de pratique, avec des exercices réguliers en laboratoire et un projet final impliquant l'audit de la qualité d'un projet logiciel réel. Ce cours permettra aux étudiants de se familiariser avec les outils et techniques utilisés dans l'industrie pour améliorer la qualité des produits logiciels et les préparer à des défis concrets dans des environnements professionnels.

Intitulé du Chapitre	Contenu indicatif	Concepts clés
Chap. 1. Introduction à la qualité logicielle et aux modèles de qualité	Ce chapitre pose les bases de la qualité logicielle. Il introduit les concepts de qualité, les enjeux pour l'industrie, et les différents modèles de qualité. Les étudiants apprendront l'importance de définir des critères de qualité dès les premières étapes du développement et les standards qui encadrent la gestion de la qualité.	<ul style="list-style-type: none"> Définition de la qualité logicielle Modèles de qualité : ISO 9126, ISO 25010, CMMI Critères de qualité : efficacité, maintenabilité, performance, sécurité
Chap. 2. Métriques de qualité logicielle et analyse du code	Ce chapitre aborde l'évaluation de la qualité à travers des métriques de code. Les étudiants découvriront comment mesurer la qualité d'un logiciel à l'aide de différentes métriques (complexité cyclomatique, couverture de code, densité des défauts, etc.). Des outils d'analyse statique seront également introduits pour détecter les problèmes de code en amont.	<ul style="list-style-type: none"> Outils d'analyse statique : SonarQube, PMD, Checkstyle Interprétation des résultats de l'analyse statique Amélioration de la lisibilité et de la maintenabilité du code via le refactoring
Chap. 3. Tests logiciels : stratégies et automatisation	Dans ce chapitre, les étudiants apprendront les différentes stratégies de tests logiciels, avec un accent particulier sur les tests unitaires, d'intégration et de régression. L'automatisation des tests sera abordée, en mettant l'accent sur les outils et les bonnes pratiques pour garantir la couverture et la répétabilité des tests dans un processus continu.	<ul style="list-style-type: none"> Types de tests : tests unitaires, tests d'intégration, tests de régression Automatisation des tests avec des outils comme JUnit, TestNG, Selenium Intégration continue et pipelines CI/CD pour l'automatisation des tests
Chap. 4. Qualité et gestion des risques dans les environnements Agile et DevOps	Ce chapitre explore l'intégration de la qualité dans les processus Agile et DevOps. Les étudiants apprendront à travailler dans des équipes agiles où la qualité est constamment mesurée et améliorée tout au long du cycle de développement. Les concepts de DevOps, d'intégration continue et de déploiement continu seront également abordés.	<ul style="list-style-type: none"> Qualité dans un environnement Agile : tests dans les sprints, rétrospectives de qualité DevOps et automatisation des tests dans le pipeline Pratiques de gestion des risques : gestion des vulnérabilités, sécurité logicielle

V) Ressources complémentaires

1. Glossaire des principaux concepts

- Qualité logicielle** : Ensemble des caractéristiques d'un logiciel permettant de satisfaire les attentes des utilisateurs et de répondre aux critères de performance, sécurité, maintenabilité, etc.

- **Refactoring** : Processus de réorganisation du code source sans en modifier le comportement pour en améliorer la structure.
- **Analyse statique du code** : Technique permettant d'examiner le code sans l'exécuter afin de détecter des anomalies, erreurs potentielles, ou violations de bonnes pratiques.
- **Automatisation des tests** : Pratique qui consiste à écrire des scripts pour tester automatiquement les différentes fonctionnalités du logiciel à chaque modification du code.
- **CI/CD (Continuous Integration / Continuous Deployment)** : Pratiques qui permettent d'intégrer et de déployer le code de manière continue et automatique à chaque modification apportée.
- **Couverture de code** : Mesure du pourcentage de code exécuté pendant les tests. Plus la couverture est élevée, mieux on s'assure que le code est testé.
- **DevOps** : Méthodologie qui vise à intégrer étroitement les équipes de développement et d'opérations, permettant ainsi une livraison continue de logiciels de qualité.
- **Tests unitaires** : Tests réalisés pour vérifier le bon fonctionnement d'une unité de code, souvent une fonction ou une méthode.
- **Tests de régression** : Tests effectués après chaque modification du logiciel pour vérifier que les modifications n'ont pas introduit de nouveaux défauts.

2. Lectures

"Software Engineering: A Practitioner's Approach" (Roger S. Pressman)

"Foundations of Software Testing" (Aditya P. Mathur) Livre essentiel pour comprendre les bases des tests logiciels, des tests unitaires aux tests de performance.

"Test-Driven Development: By Example" (Kent Beck)

3. Ressources multimédia

3.1 Ressources multimédia obligatoires : Vidéo, sites web, etc.

3.2 Ressources multimédia suggérées : Vidéo, sites web, etc.

VI) Activités d'apprentissage

Résumé

Description

Évaluation formative

VII) Modalités d'évaluation

Contrôle Continu			Examen Final	
Nombre	Nature	Durée	Nature	Durée
1	Écrit/Oral	2h	Écrit/Oral	3h
Poids (%) : 40			Poids (%) : 60	