

INTITULE DE LA MATIERE : **ANALYSE ET CONCEPTION LOGICIELLE**

CODE : UE-82

SEMESTRE : 8

NOM DE L'ENSEIGNANT RESPONSABLE : P. BELLOT

NOMBRE D'HEURES ENCADREES CM : 12 TD : 10 TP : 18

NOMBRE D'HEURES DE TRAVAIL PERSONNEL DE L'ELEVE – estimation : 25

OBJECTIF GENERAL (pas plus de 5 lignes) :

L'objectif de cet enseignement est avant tout méthodologique puisqu'il consiste à présenter une méthode de modélisation de logiciels et de processus fondée sur une approche itérative et incrémentale incluant les phases d'analyse, de conception (logicielle) et de test, en exploitant le langage UML (Unified Modeling Language).

CONTENU – PLAN DU COURS (pas plus de 15 lignes) :

- 1- Présentation du cycle de vie du logiciel via une approche itérative et incrémentale (Unified Process, méthodes agiles)
- 2- Rappels de concepts liés au développement et à la conception orientée objets
- 3- Le langage de modélisation UML, de l'analyse à la conception : cas d'utilisation, acteurs, diagrammes d'activités, diagrammes de classes et modèles d'éléments, diagrammes de séquences, d'interaction et d'états, déploiement (composants, génération de code) et représentation de l'environnement
- 4- UML pour la modélisation de processus (Business Process modeling)
- 5- Panorama des environnements logiciels UML (+ TP)
- 6- Techniques de test : le test dans le cycle de vie du logiciel (Test case et Use case)
- 7- L'automatisation du test logiciel : scripts et robots, tests unitaires
- 8- Les métriques du test logiciel
- 9- L'analyse structurelle pour le test et les tests de couverture
- 10- Les environnements de test logiciels

FORME DE L'EVALUATION :

Contrôle continu Contrôle terminal Mémoire/rapport Soutenance
préciser si nécessaire, nombre d'épreuves : et type oral / écrit

ACQUIS DE LA FORMATION ATTENDUS, ET QUI SONT EVALUES (5 à 10 items)

A l'issue de cet enseignement, l'élève est capable de :

Utilisez les termes comprendre, analyser, exploiter, interpréter, calculer, utiliser, mettre en œuvre, choisir, communiquer, rédiger,...

- Utiliser le langage de modélisation UML pour l'analyse et la conception de logiciels
- Comprendre et mettre en œuvre les méthodes agiles de développement logiciel
- Rédiger les cas d'utilisation et les scénarios de test d'un logiciel
- Choisir et utiliser un logiciel et un environnement de modélisation et de test en fonction des objectifs
- Calculer la couverture de tests logiciels et les valeurs des métriques de qualité
- Mettre en œuvre un plan de test et un suivi de modifications et de demandes de changements dans un projet logiciel.

PREREQUIS (pas plus de 5 lignes): Concepts de développement objet, langage Java

REFERENCES, BIBLIOGRAPHIE (pas plus de 5 lignes) :

- Analyse des besoins pour le développement logiciel : Recueil et spécification, démarches itératives et agiles, J. Lonchamp, Eyrolles, 2015
- UML 2 en action : De l'analyse des besoins à la conception, P. Roques, F. Valée, Eyrolles, 2007

- UML 2.5 par la pratique: Etudes de cas et exercices corrigés, P. Roques, Eyrolles, 2018.
- Pratique des tests logiciels 3è éd. : Concevoir et mettre en oeuvre une stratégie de tests Préparation à la certification ISTQB, J.-F. Pradat-Peyre, J. Printz, Dunod, 2017.
- Industrialiser le test fonctionnel - 2e édition: Pour maîtriser les risques métier et accroître l'efficacité du test, B. Legeard, F. Bouquet, N. Pickaert, Dunod, 2011.