

UE Techniques avancées de validation



Niveau d'étude
Bac +5



ECTS
3 crédits



Crédits ECTS
Echange
3.0



Composante
UFR IM2AG
(informatique,
mathématiques
et
mathématiques
appliquées)



Période de
l'année
Automne (sept.
à dec./janv.)

- > **Langue(s) d'enseignement:** Français
- > **Méthodes d'enseignement:** En présence
- > **Forme d'enseignement :** Cours magistral
- > **Ouvert aux étudiants en échange:** Non
- > **Crédits ECTS Echange:** 3.0
- > **Code d'export Apogée:** GBGI9U13
- > **Temps de travail personnel pour l'étudiant:** 30

Présentation

Description

Tout développement logiciel est confronté au défi de vérifier et de valider le logiciel produit. Même si d'aucuns considèrent qu'il est tout à fait possible de vivre au milieu de programmes erronés, la responsabilité de tout ingénieur logiciel est de mettre en œuvre des moyens appropriés pour développer un logiciel de qualité. Le test des logiciels devient aujourd'hui un métier à part entière parmi les métiers du développement logiciel. De plus, le développement de grands logiciels fait largement appel à la sous-traitance, à la réutilisation, ou à l'intégration de composants. Dans ce contexte, il est essentiel de valider ces composants ainsi que leur assemblage.

L'objectif de ce cours est de donner aux étudiants les outils et techniques qui leur permettront de mettre en œuvre diverses activités de vérification ou de validation au cours d'un développement logiciel basé sur une spécification. Il s'agit d'exprimer une spécification qui puisse être vérifiée à l'exécution, comme oracle d'une activité de test. Dans un deuxième temps, il s'agit de mettre en œuvre des techniques de test pour générer, si possible automatiquement, des tests et coupler l'activité de test avec un oracle. Nous verrons comment ces oracles, utilisés avec des techniques de bouchonnage (mock), peuvent être utilisés en support d'un développement orienté par le test (TDD, BDD) et d'une intégration continue (Gitlab CI). Toujours en utilisant les concepts d'oracle

et d'intégration continue, nous nous intéresserons aux tests des programmes réactifs (i.e., qui interagissent en continu avec leur environnement).

Nous verrons également comment des spécifications formelles peuvent, dans certains cas, être vérifiées exhaustivement (model-checking, SMT, interprétation abstraite).

Un extrait des thématiques et outils qui seront abordés est le suivant :

- + Tests unitaires (JUnit)
- + Techniques de bouchonnage (mockito)
- + Tests combinatoires et aléatoires basés sur les propriétés (JUnit Theories, QuickCheck)
- + Développement dirigé par le test (TDD), par le comportement (Jbehave),
- + Intégration continue (Gitlab CI)
- + Tests de programmes réactifs
- + Model-checking, Sat-Modulo Théorie
- + Interprétation Abstraite

L'enseignement est organisé en des séances de cours, destinées à présenter diverses techniques applicables pour les diverses activités du test (spécification, génération, exécution, oracle et arrêt du test). Ces séances sont complétées par des séances de travaux dirigés, sur machine, qui permettront aux étudiants de se familiariser avec des outils de test. Le contrôle continu prendra la forme de mini-projets associés aux séances de travaux dirigés

Heures d'enseignement

CM	CM	15h
TD	TD	15h

Pré-requis recommandés

Notions de base du test logiciel, du langage de programmation Java et de la logique des prédicats.

Période : Semestre 9

Compétences visées

L'objectif de ce cours est de donner aux étudiants des outils et techniques qui leur permettront de mettre en oeuvre les diverses activités du test et de validation au cours d'un développement logiciel.

Infos pratiques

Contacts

Responsable pédagogique

Erwan Jahier

✉ Erwan.Jahier@grenoble-inp.fr, Erwan.Jahier@univ-grenoble-alpes.fr

Lieu(x) ville

> Grenoble

Campus

> Grenoble - Domaine universitaire