

Génie logiciel / Software engineering



Composante
Polytech
Grenoble - INP,
UGA

- > **Langue(s) d'enseignement:** Français
- > **Ouvert aux étudiants en échange:** Oui
- > **Code d'export Apogée:** KARI9M03

Présentation

Description

Objectifs : sensibilisation à la conception logicielle (architecture) et la validation (tests)

Compétences visées :

- représenter une architecture logicielle cohérente (dans un formalisme donné)
- connaître, savoir utiliser (en justifiant) des styles architecturaux
- sensibilisation à la validation d'une architecture par rapport à des exigences
- resituer le test dans un cycle de développement par rapport aux exigences
- connaître et savoir des techniques élémentaires de constructions de tests (structurelles et fonctionnelles)

Le cours se découpe en 2 parties : architecture et test.

1. Architecture

- introduction : de l'importance de l'architecture dans la conception
- représentation : différentes vues pour capturer l'architecture
- conception : notion de style/patron et de techniques
- validation : lien exigences-architecture; cohérences de vues.

2. Test

- notion de faute, erreur, défaillance
- notion de données de test vs oracle
- méthodes de "test fonctionnel" (catégorie et partition)
- méthodes de "test structurel" (notion de couverture du graphe de contrôle, def-use, ...)

- outils de tests (couverture et drivers d'exécution)

Objectives: awareness of software design (architecture) and validation (tests)

Targeted skills :

- represent a coherent software architecture (in a given formalism)
- know, know how to use (justifying) architectural styles
- awareness of the validation of an architecture with respect to requirements
- put the test back into a development cycle in relation to the requirements
- know and know basic techniques of test construction (structural and functional)

It is composed of two parts concerning both architectural design and validation by test.

1. Architecture

- introduction
- representation
- design
- validation

2. Test

- fault/error/failure
- test data and oracle
- "black-box" testing methods
- "white-box" testing methods
- tools for testing

Heures d'enseignement

Génie logiciel / Software engineering - CMTD

Cours magistral - Travaux dirigés

38h

Pré-requis recommandés

Bon niveau en algorithmique-programmation. Premières expériences de projets de développement.

Good level in algorithmics and programming. First experiences in development projects.

Période : Semestre 9

Évaluation initiale / Session principale - Épreuves

Libellé	Nature de l'enseignement	Type d'évaluation	Nature de l'épreuve	Durée (en minutes)	Nombre d'épreuves	Coefficient de l'épreuve	Remarques
						40/100	

Bibliographie

- "Introduction to Software Testing" - Paul Ammann, Jeff Offutt
- "Software Architecture in Practice" - Len Bass, Paul Clements and Rick Kazman
- "Pattern-Oriented Software Architecture Volume 1: A System of Patterns" - Frank Buschmann, Regine Meunier, Hans Rohnert and Peter Sommerlad

Infos pratiques

Lieu(x) ville

- > Grenoble

Campus

- > Grenoble - Saint-Martin d'Hères