UNIVERSITÀ
DEGLI STUDI DI MILANO-BICOCCA

COURSE SYLLABUS

Evolution of Software Systems and Reverse Engineering

2627-2-F1802Q121

Aims

The student will acquire skills related to the main issues of software evolution and reverse engineering, be able to perform empirical software analysis and use various tools for reverse engineering, supporting software understanding, evolution and maintenance, and tools for technical debt management.

Knowledge and Ability to Understand
By the end of the course, the student will have acquired a solid knowledge on maintenance and evolution of complex software projects and technical debt management.
Applied Knowledge and Ability to Understand
The student will be able to perform various software analyses, including correlation analysis between given software problems and prediction analysis. Each student will be expected to know and use tools well known in companies for code understanding, technical debt management and reverse engineering.
Autonomy of judgment
The course promotes the development of critical thinking through the comparative evaluation of tools for software analysis and maintenance, and through individual contribution in the examination project. Students will also have the opportunity to attend one or two company seminars on topics of great interest in the evolution of software systems and reverse engineering that will further stimulate judgment skills.
Communication Skills
The student will be able to communicate effectively through one or two individual classroom presentations, through the exam project related to the use of tools seen in class to perform empirical analysis of software projects, through the document produced on the exam project, and through the individual oral test.
Ability to learn
The course promotes the development of autonomy in study through an active teaching approach. At the end of the course, the student will be able to independently investigate advanced topics on software evolution and maintenance that can also result in a thesis, thus consolidating the basis for continuous learning beyond the curricular context. Thesis dissertation topics may concern, for example, evaluating the impact of refactoring of architectural problems on security, performance, energy consumption and other quality metrics, or on migrating to microservice architectures, managing technical debt or on using LLMs to recognize and refactor architectural problems in the code.

Contents

Introduction to the principal problematics of reverse engineering, software evolution, software quality assesment and program comprehension.
Deep study of some topics with different tools experimentations.
Software quality analysis: metrics and tools.
Object-oriented reeengineering patterns.

Detailed program

1 Introduction to Software evolution and Reverse Engineering. Legacy systems. Software comprehension and maintainability: principal problems.

2 Techniques and tools for Reverse Engineering. Introduction and application to model-driven reverse engineering.

3 Object-oriented patterns for reverse engineering and reengineering.

4 Software quality metrics and software quality assessment. Application Portfolio Management: problems, tools, techniques and metrics.

5 Modernization of legacy systems: Migration of legacy systems towards microservice architecture.

6 Tools and techniques for software architecture reconstruction.

7 Antipattern, code and architectural smell detection, their refactoring.
8. Impact of refactoring techniques on the code quality, on performance issues and energy consumption.

9. Empirical analysis of different kinds: Correlation analysis among code smells and metrics, Prediction analysis of code anomalies or different other problems through machine learning techniques.

10. Introduction to techniques of hacking, decompiling and code obfuscation to protect code. Static and dynamic analysis for reverse engineering.

11.. Tools and techniques for Managing Technical Debt.
12. Using LLM in revrse engineering and software quality assessment.

Prerequisites

Knowledge of Java Language.

Teaching form

The lessons will be given in presence.

Lessons, exercitations, students experimentatons of tools with an oral presentation.

The course can be offered in Italian or in English language, according to the students attending the course. and their preferences.

6 lessons of 2 hours in presence
15 lessons of 2 hours in presence with student interactions
4 lessons with exercise and students interactions in presence

Textbook and teaching resource

Slides, papers, online books, survey and tutorial, Master and PhD thesis available online.

Software Engineering, Sommerville, only 3 chapters

Object Oriented Reeengineering patters, Oscar Nierstrasz -available online

Most of the material necessary to prepare the exam will be available online.

Semester

I semester

Assessment method

Final exam with a project or experimentation of some tools of reverse engineering.

Project done alone or in maximum two students. Evaluation 0-22 points.

Oral discussion on the project. Evaluation 0-8 points.

One or two tasks assigned during the course with a presentation. Evaluation 0-3 points.

Office hours

On appointment.

Sustainable Development Goals