



UNIVERSITÀ  
DEGLI STUDI DI MILANO-BICOCCA

## COURSE SYLLABUS

### Evolution of Software Systems and Reverse Engineering

2122-2-F1801Q158

---

#### Obiettivi

Lo studente acquisirà competenze relative alle problematiche principali dell'evoluzione del software e della reverse engineering, sarà in grado di effettuare analisi e utilizzare diversi strumenti di reverse engineering e di supporto alla comprensione, evoluzione e manutenzione del software.

#### Contenuti sintetici

Introduzione alle principali problematiche di reverse engineering, software evolution e program comprehension. Approfondimenti di alcuni argomenti e sperimentazioni di diversi tool.

#### Programma esteso

1 Introduzione alla Software evolution, Reverse Engineering, Sistemi Legacy. Comprensione e manutenzione del software.

2 Tecniche e tools per la Reverse Engineering e l'Integrazione di sistemi. Introduzione e applicazione della model-driven reverse engineering.

3 Pattern object-oriented per la reverse engineering;

4 Metriche di Qualità del Software , software quality assessment. Application Portfolio Management: problemi, tools, tecniche e metriche.

5. Modernizzazione dei sistemi legacy: Migrazione dei sistemi legacy verso micro services.

Approccio unificato alla reverse engineering dei dati e del software.

6. Tool e tecniche per la design pattern detection nel codice. Tecniche di data mining per la design pattern detection. Tools per la software architecture reconstruction.

7. Riconoscimento di antipattern, code smell e architectural smells nel codice, il loro refactoring. Impatto delle tecniche di refactoring sulla qualità del codice.

8. Esempi di diverse analisi empiriche: Analisi di correlazione fra diverse metriche di qualità del codice e violazioni del codice e design; Analisi della prediction di problemi del software attraverso tecniche di machine Learning e data mining..

9. Tecniche di hacking, decompiling and code obfuscation. Analisi statica e dinamica per la reverse engineering .

10. Tecniche e strumenti per l'identificazione e gestione del Technical Debt.

## **Prerequisiti**

Conoscenza approfondita del linguaggio Java,

Conoscenza dei design patterns.

## **Modalità didattica**

Nel periodo di emergenza Covid-19 le lezioni si svolgeranno molto probabilmente in modalità mista: parziale presenza e lezioni videoregistrate asincrone/sincrone.

Lezioni frontali ed esercitazioni, approfondimenti di alcuni contenuti e sperimentazione di alcuni strumenti in aula attraverso presentazione da parte degli studenti.

Il corso viene erogato in italiano, ma se uno studente straniero è presente, il corso può essere erogato in inglese.

## **Materiale didattico**

Slides del docente, articoli, survey e tutorial forniti dal docente, tesi di laurea e di dottorato svolte presso il laboratorio di ricerca Essere e presso altre Università straniere con cui il laboratorio collabora.

Testo

Ingegneria del Software, Sommerville, solo 3 capitoli.

Object Oriented Reengineering patterns, Oscar Nierstrasz -Disponibile online

La maggior parte del materiale su cui dovrete studiare sarà disponibile on line.

## **Periodo di erogazione dell'insegnamento**

I semestre

## **Modalità di verifica del profitto e valutazione**

Due task assegnati durante il corso che prevedono una presentazione in aula. Valutazione 0-3 punti.

Un progetto finale singolo o al massimo in due studenti relativo alla sperimentazione di alcuni tool di reverse engineering e/o analisi empiriche. Valutazione 0-22 punti

Discussione orale sul progetto finale. Valutazione 0-8 punti.

## **Orario di ricevimento**

Per appuntamento.

---