



UNIVERSITÀ DEGLI STUDI  
DI MILANO

# Ingegneria del software

Laurea Triennale in Informatica



UNIVERSITÀ DEGLI STUDI  
DI MILANO

# **Lezione 1:**

# **Introduzione al corso**

# Informazioni pratiche

## Orario lezioni

*Lun* 14:30-17:00 aula G10

*Mer* 13:00-15:30 aula 301

## Orario Laboratorio

*Gio* 14:30-18:30 due turni equivalenti 307 e 309

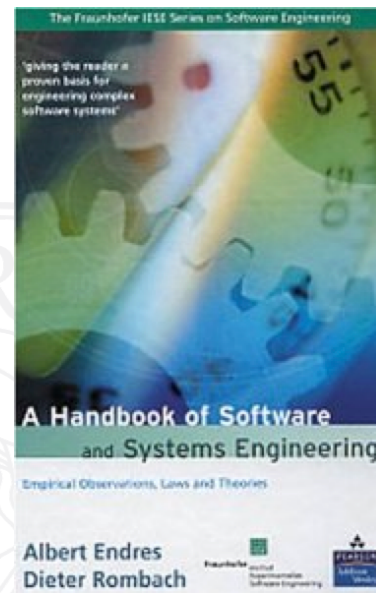
Ricevimento su appuntamento

*email:* [carlo.bellettini@unimi.it](mailto:carlo.bellettini@unimi.it)

# Libro del corso

## Appunti studenti

- Non c'è un vero "libro di testo"



- ma anche altri... o risorse di rete

<https://bellettini.di.unimi.it>

Gruppi Telegram

# Programma corso

- Processi di gestione del Software
- Progettazione del Software
- Verifica e convalida
- Specifiche del Software

# Modalità d'esame

## Esame di laboratorio

- prova pratica in laboratorio di 4 ore
- oppure durante l'anno ci saranno 2 laboratori valutati

## Esame orale per la teoria

# Laboratorio

Iniziano il 5 ottobre

- non serve prenotarsi (mai chiesto)... vedremo al momento come siamo messi
- in prima istanza la divisione in turni verrà fatta secondo parità della matricola
  - turno A: matricole pari, turno B: matricole dispari
- chi vuole usare il proprio computer (consigliato) deve avere installato:
  - git (tool di *versioning* v.2.42+)
  - gradle (tool di *automation building* v.8.1+)
  - IntelliJ (IDE di sviluppo v.2023.2.2+)
  - JDK (la 17+)

# Compitini

Laboratori valutati (compitini)

- bisogna essere frequentanti
  - ci sarà tolleranza per max **una** assenza a compitino
- non può partecipare chi non ha sostenuto con successo esame di Programmazione (1)
- chi non ha sostenuto con successo esame di programmazione 2... è fortemente sconsigliato... ma nel caso verranno messi in coppia tra di loro
- Non valgono più le prove sostenute quando tenevo il corso di ingegneria del software qualche anno fa...





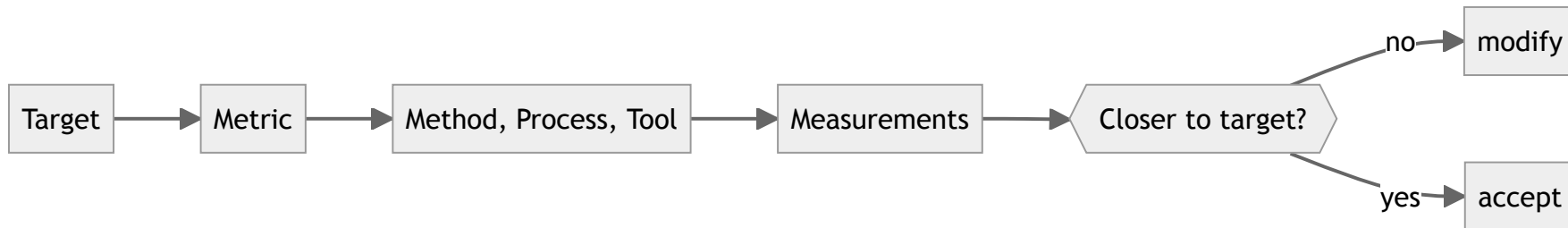
UNIVERSITÀ DEGLI STUDI  
DI MILANO

# Lezione 1: Inizio corso

# Processi di produzione del sw

- Negli anni '50 e '60 si è colta la necessità di superare metodi di produzione artigianale
- Studio quindi di tecniche e metodologie che potessero migliorare e "*assicurare*" software di qualità

# Approccio ingegneristico



Ma qual è il nostro Target?

# Principali problemi

- Numero e tipo **persone** coinvolte
  - il programmatore non è il cliente e questo crea problemi di comunicazione
- **Dimensioni** del software
  - milioni di linee di codice
  - migliaia di anni uomo
- **SOFT**ware
  - malleabilità porta a moltiplicarsi di versioni e evoluzioni

# Perché studiare un processo?

- Convinzione che un buon **processo** produca un **prodotto** di qualità
- Quali sono le *qualità* a cui miriamo nel software?
  - che **funzioni**
  - che sia **bello**
  - che mi faccia diventare **ricco**

# Cosa vuol dire *che funziona*?

- che fa quello che è stato chiesto

- **CORRETTEZZA**

... ma se mi è stato chiesto qualcosa di sbagliato o incompleto?

## **R.Glass' Law (L1):**

Requirements deficiencies are the *prime source* of projects failures

- che mi posso *fidare*

- **AFFIDABILITÀ**

- che non fa male

- **INNOCUITÀ (SAFETY) e/o ROBUSTEZZA**

# Cosa vuol dire *bello*?

- facile da usare
  - **USABILITÀ**

**Nielsen-Norman's Law (L26):**  
Usability is quantifiable

- veloce
  - **EFFICIENZA** nell'uso delle risorse
- pulito
  - **VERIFICABILITÀ**

# Come fa a *farmi diventare ricco*?

- non rifare qualcosa di già fatto
  - **RIUSABILITÀ** di componenti

## McIlroy's Law (L15):

Software reuse reduces cycle time and increases productivity and quality

- semplificare gli interventi post consegna
  - **MANUTENIBILITÀ**
    - correzione errori (RIPARABILITÀ)
    - estensione dei requisiti, nuove situazioni (EVOLVIBILITÀ)

## M. Lehman's Laws (L27 e L28):

A system that is used will be changed

An evolving system increases its complexity unless work is done to reduce it



# Come deve essere un processo?

... anche lui deve: funzionare, essere bello, farmi diventare ricco ...

- resistere agli imprevisti
  - **ROBUSTEZZA**
- essere veloce
  - **PRODUTTIVITÀ**
- cogliere l'attimo
  - **TEMPISMO**

# Volatilità dei *requirements*

<https://dilbert.com/strip/2002-02-27>



<https://dilbert.com>

<https://xkcd.com>

<https://phdcomics.com>

<https://geek-and-poke.com>