

Corso di Ingegneria del Software

Prof. Ing. Andrea Calvagna

A.A. 2023/2024



Obiettivi del corso

- Fondamenti di programmazione orientata agli oggetti in Java
- Notazioni standard per la progettazione object oriented: Unified Modelling Language
- Principi di progettazione riusabile: Design Patterns
- Tecniche di Refactoring
- Ciclo di vita dei progetti software
- Metodologie per il Testing del software
- Misurazione della qualità del software

Lezioni

- Coprono tutto il programma del corso
- Partecipazione fortemente consigliata: si impara di più, e si ascolta da un esperto, è possibile fare domande ed ottenere risposte
- Ricevimento: mandare un messaggio su Teams per un appuntamento oppure provare a passare in studio
- Per rendere efficace lo studio: esercitarsi con il codice, usare i concetti spiegati e i tool consigliati, partecipare alle lezioni, non limitarsi alle slides

Lezioni

	Aula 24				
	Lun	Mar	Mer	Gio	Ven
8:00 - 9:00	Func. Conc. Progr Barbanera	Tutorato	Func. Conc. Progr Barbanera	Tutorato	Inglese Cacciola
9:00 - 10:00					
10:00 - 11:00		Ingegneria del SW M-Z Calvagna		Ingegneria del SW M-Z Calvagna	
11:00 - 12:00	Tutorato		Tutorato		Big Data Pulvirenti
12:00 - 13:00					
13:00 - 14:00					
14:00 - 15:00					
15:00 - 16:00	Reti di Calcolatori M-Z Riccobene/GuarneraL	Sistemi Operativi M-Z Pavone	Reti di Calcolatori M-Z Riccobene/GuarneraL	Sistemi Operativi M-Z Pavone	Tutorato
16:00 - 17:00					
17:00 - 18:00					
18:00 - 19:00					

Informazioni utili

- Materiale, slides e altro su gruppo teams di **Ingegneria del Software**
- Codice mostrato a lezione su <https://github.com/amcalvagna>
- Gruppo su Teams e su Telegram per avvisi sul corso
- Condiviso col prof. E. Tramontana e il dott. Scardace (Tutor)

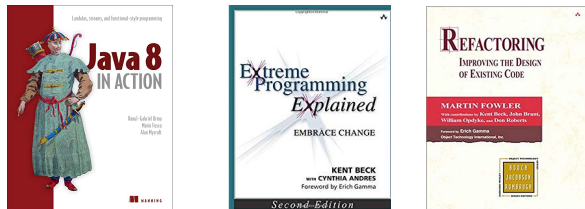
Libri consigliati

- Le slide non bastano :-)
- Sethi, R. (2022). Software Engineering: Basic Principles and Best Practices. Cambridge: Cambridge University Press.
- Fowler. UML Distilled. Pearson
- Gamma, Helm, Johnson, Vlissides. Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley
- Sommerville. Ingegneria del Software. Pearson
- Pressman. Principi di Ingegneria del Software. McGraw-Hill



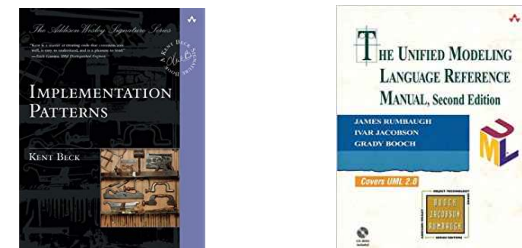
Libri Consigliati

- Urma, Fusco, Mycroft. Java 8 in Action. Manning
- Beck. Extreme Programming Explained. Addison-Wesley
- Fowler. Refactoring: Improving the design of existing code. Addison-Wesley



Libri per Approfondimenti

- Beck. Implementation Patterns. Addison-Wesley
- Rumbaugh, Jacobson, Booch. The Unified Modeling Language Reference Manual. Addison-Wesley



Modalità Esami

- Test a risposte multiple, test a risposta aperta (implementare codice, disegnare alcuni diagrammi UML), orale
- Progetto opzionale, da concordare (a partire da maggio)

What is Software Engineering?

What happens?



Arianne 5



On June 4, 1996, the Arianne 5 took off on its maiden flight.

Space Shuttle Software

- Cost: \$10 Billion, millions of dollars more than planned
- Time: 3 years late
- Quality: First launch of Columbia was cancelled because of a synchronization problem with the Shuttle's 5 onboard computers.
 - Error was traced back to a change made 2 years earlier when a programmer changed a delay factor in an interrupt handler from 50 to 80 milliseconds.
- Substantial errors still exist.
 - Astronauts are supplied with a book of known software problems "Program Notes and Waivers".

USS Yorktown



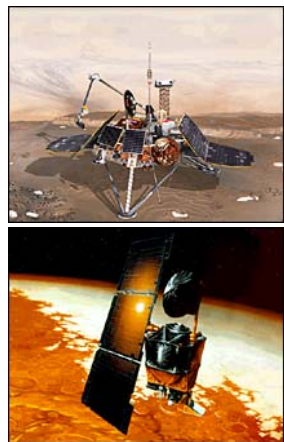
“After a crew member mistakenly entered a zero into the data field of an application, the computer system proceeded to divide another quantity by that zero. *The operation caused a buffer overflow, in which data leaked from a temporary storage space in memory, and the error eventually brought down the ship's propulsion system.* The result: *the Yorktown was dead in the water for more than two hours.*”

“Better, Faster, Cheaper”

• In Sept.'99, NASA lost both the Mars Polar Lander and the Climate Orbiter.

• Later investigations determined software errors were to blame.

- Orbiter: Component reuse error.
- Lander: Precondition violation.



Automotive Analogy

- “If the automobile had followed the same development as the computer, a Rolls-Royce would today cost \$100, get a million miles per gallon, and ...

Automotive Analogy

- "If the automobile had followed the same development as the computer, a Rolls-Royce would today cost \$100, get a million miles per gallon, and explode once a year killing everyone inside."
- - Robert Cringely

Warrantees?

LIMITED WARRANTY. Microsoft warrants that (a) the SOFTWARE PRODUCT will perform substantially in accordance with the accompanying written materials for a period of ninety (90) days from the date of receipt, ...

LIMITATION OF LIABILITY. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, *IN NO EVENT SHALL MICROSOFT OR ITS SUPPLIERS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING, ...) ARISING OUT OF THE USE OF ... THE SOFTWARE PRODUCT... MICROSOFT'S ENTIRE LIABILITY ... SHALL BE LIMITED TO THE GREATER OF THE AMOUNT ACTUALLY PAID BY YOU FOR THE SOFTWARE PRODUCT OR U.S. \$5.00; PROVIDED...*

Group Activity: What is Software Engineering?

- Come up with a group answer to the question:
 - How would you define it or characterize it?
- Choose someone to report the group answer(s)

Early Use of the term "software engineering"

Margaret Hamilton got the Presidential Medal of Freedom, November 22, 2016



"Software during the early days of [the Apollo space missions] was treated like a stepchild and not taken as seriously as [hardware] ... I fought to bring the software legitimacy so that it (and those building it) would be given its due respect"

– Margaret Heafield Hamilton

From a 2014 interview

What is Software Engineering?

a raw and incomplete definition

- The discipline that studies:
 - The **process** of building a software **product**.
- Some questions to put SE in perspective:
 - What are the sizes of some typical software products?
 - How many people would it take to build these in 1 year? 2?
 - What would you do if a bug could cost lives and \$2 billion?
 - What would you do if a delay could cost \$100's or millions?

Software Engineering: Definition

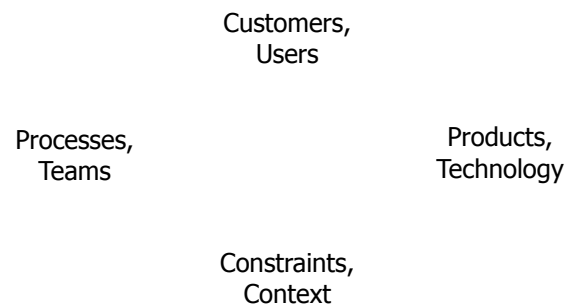
Software Engineering is a collection of techniques, methodologies and tools that help with the production of

- a high quality software system
- with a given budget
- before a given deadline

while change occurs.

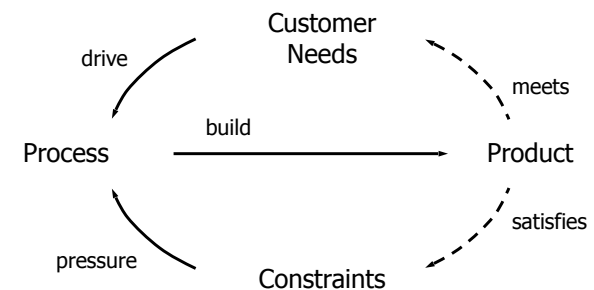
Drivers (Forces) in the Definition

A significant change related to any one of them affects all aspects of a project



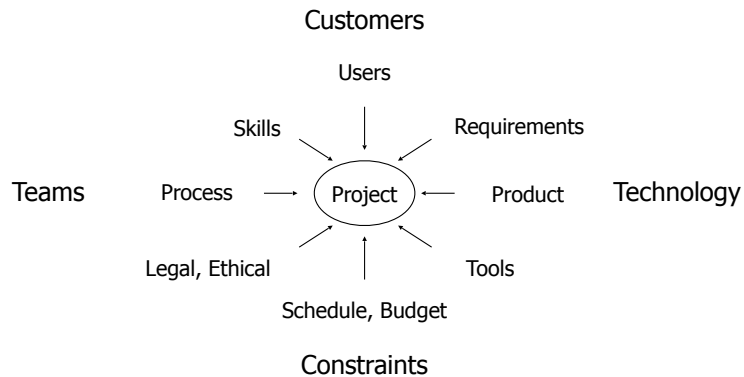
Software Engineering

Developers can control the Process (how they work) and Product (what they build)



Representative Sample of Driving Forces

Drivers are related; e.g., teams use processes, products embody technology



Role of software engineer

- Programming skill not enough
- Software engineering involves "programming-in-the -large"
 - understand requirements and write specifications
 - derive models and reason about them
 - master software
 - operate at various abstraction levels
 - member of a team
 - communication skills
 - management skills

Qualità del software

- Le tecniche dell'ingegneria cercano di produrre sistemi software entro i costi e i tempi preventivati e con **qualità** accettabile
- Criteri operativi per valutare la qualità
 - **Correttezza**: il sistema software raggiunge lo scopo ed è conforme alle specifiche
 - Il sistema software fa quello che il cliente vuole?
 - Il sistema software soddisfa le specifiche?
 - **Efficienza, manutenibilità, dependability** (sicurezza e affidabilità), **usabilità**

Qualità del software

- **Manutenibilità**: un sistema software è intrinsecamente modificabile, poiché non ha parti fisiche
- Se un sistema software è di successo sarà necessario cambiarlo
 - Per adattarlo ad una realtà che cambia (mutate o nuove esigenze)
- Le richieste di estensione aumentano al crescere del successo
- Poiché di successo, il sistema software sopravvive all'hardware per cui era stato sviluppato inizialmente, generando una nuova esigenza di adattamento alla nuova piattaforma

15 min break

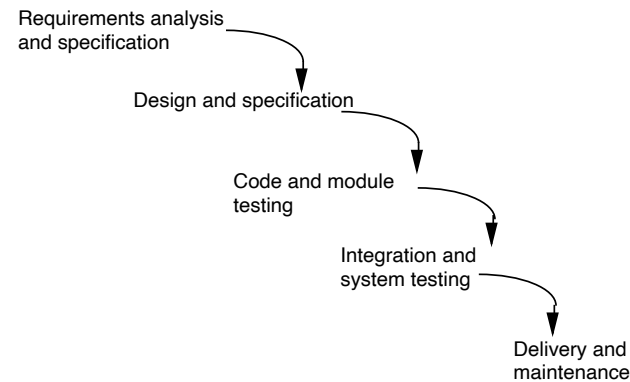


Software Lifecycle Definition

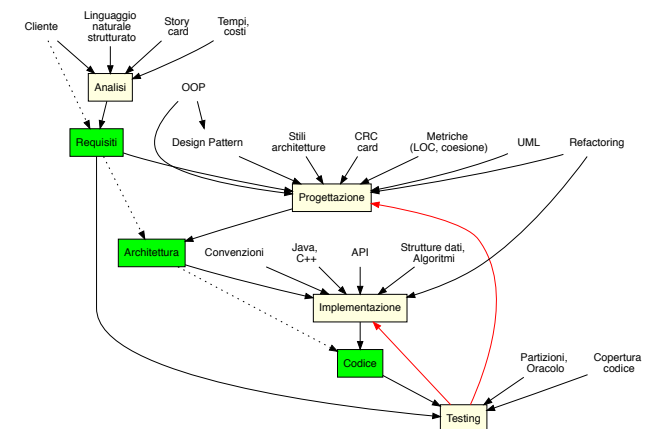
- Set of activities and their relationships to each other to support the development of a software system
- Typical Lifecycle questions:
 - Which activities should I select for the software project?
 - What are the dependencies between activities?
 - How should I schedule the activities?

The software lifecycle
(a preview)

Waterfall model

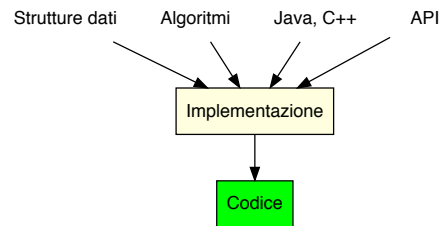


The software lifecycle
(enhanced model)



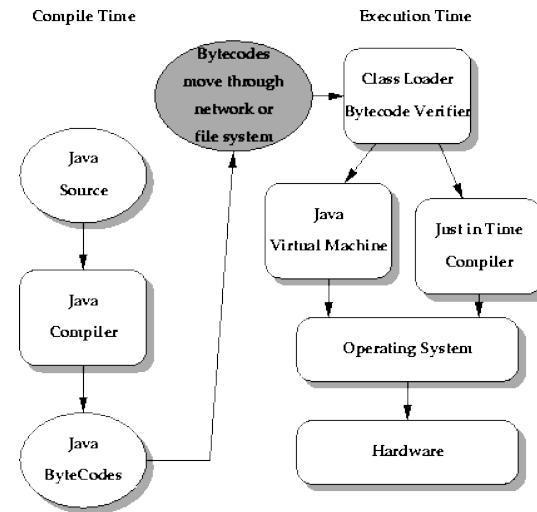
giallo = attività
verde = artefatto

Svilupperemo in Java...



giallo = attività
verde = artefatto

Java Program Environment



HelloWorld.java

```
import java.time.LocalDate;

/**
 * Classe che stampa sullo schermo un messaggio e la data corrente
 */
public class HelloWorld { // definizione classe
    // dichiarazione e assegnazione campi
    private static final String msg = "Lezione di Ingegneria del Software";
    private static final LocalDate d = LocalDate.now();

    /**
     * Metodo da cui inizia l'esecuzione del programma
     *
     * @param args parametri passati al metodo all'avvio della classe
     */
    public static void main(String[] args) {
        System.out.println("Hello World");
        System.out.println(msg);
        System.out.println(d);
    }
}
```

Output:
Hello World
Lezione di Ingegneria del Software
2020-03-03