

# Ingegneria del Software

UniVR - Dipartimento di Informatica

**Fabio Irimie**

2° Semestre 2024/2025

# Indice

<b>1</b>	<b>Introduzione</b>	<b>2</b>
1.1	Cos'è l'ingegneria del software?	2
1.2	Economia	2
1.3	Fallimento dei progetti	2
1.4	Prodotti software	2
1.4.1	Prodotti generici	2
1.4.2	Prodotti personalizzati	2
1.5	Caratteristiche di un buon software	2
1.6	Problemi che influenzano il software	3
1.6.1	Eterogeneità	3
1.6.2	Cambiamento sociale o del business	3
1.6.3	Sicurezza e fiducia	3
1.6.4	Scalabilità	3
1.7	Diversità di ingegneria del software	3
1.8	Tipi di applicazione	3
1.8.1	Applicazioni stand-alone	3
1.8.2	Applicazioni interattive transaction-based	3
1.8.3	Sistemi embedded	4
1.8.4	Sistemi batch	4
1.8.5	Sistemi di intrattenimento	4
1.8.6	Sistemi di modellazione e simulazione	4
1.8.7	Sistemi di collezione di dati (o IOT)	4
1.8.8	Sistemi di sistemi	4
1.9	Principi fondamentali	4

# 1 Introduzione

## 1.1 Cos'è l'ingegneria del software?

L'ingegneria del software è un insieme di metodologie, teorie, metodi e strumenti, che guidano nello sviluppo di software **professionale** in modo che esso fornisca le funzionalità richieste, sia performante e mantenibile, affidabile e usabile. Il **software** non è solo il programma e l'eseguibile, ma anche **la documentazione associata**. Le attività principale dell'ingegneria del software sono:

- **Specifica del software:** Il cliente e l'ingegnere del software definiscono le funzionalità e i vincoli del software da produrre.
- **Sviluppo:** Il software viene progettato e implementato.
- **Validazione:** Il software viene verificato per assicurarsi che soddisfi i requisiti forniti dal cliente.
- **Evoluzione:** Il software viene modificato per adattarlo a nuovi requisiti del cliente o del mercato.

## 1.2 Economia

Il software spesso costa più dell'hardware, e il costo è più legato alla manutenzione che allo sviluppo. Quindi l'ingegneria del software è importante per ridurre i costi di manutenzione.

## 1.3 Fallimento dei progetti

I progetti spesso falliscono per la **crescente complessità del sistema**: nuove tecniche di sviluppo e nuove tecnologie rendono i sistemi sempre più complessi, e quindi più difficili da mantenere.

## 1.4 Prodotti software

### 1.4.1 Prodotti generici

Sono prodotti che vengono pubblicizzati e venduti a qualsiasi cliente che ne faccia richiesta. La specifica è di proprietà del produttore e le decisioni sulle modifiche sono prese dai produttori.

### 1.4.2 Prodotti personalizzati

Sono prodotti che vengono commissionati da clienti specifici per soddisfare le loro necessità. La specifica è di proprietà del cliente dopo una contrattazione con il produttore e le decisioni sulle modifiche sono prese dal cliente.

## 1.5 Caratteristiche di un buon software

- **Mantenibilità:** la facilità con cui il software può essere modificato per correggere difetti, migliorare le prestazioni o adattarlo a cambiamenti.

- **Affidabilità e sicurezza:** la capacità del software di svolgere le sue funzioni in modo corretto. I malfunzionamenti non devono causare danni fisici o economici. Utenti malintenzionati non devono poter violare la sicurezza del sistema.
- **Efficienza:** il software non deve sprecare risorse (CPU, memoria, ecc).
- **Accettabilità:** il software deve essere accettato dagli utenti per i quali è stato progettato.

## 1.6 Problemi che influenzano il software

### 1.6.1 Eterogeneità

I sistemi devono sempre di più operare in modo distribuito, e quindi devono essere in grado di comunicare con sistemi diversi. Oppure bisogna garantire che il software funzioni su piattaforme diverse.

### 1.6.2 Cambiamento sociale o del business

Il software deve essere in grado di adattarsi a cambiamenti sociali o organizzativi nelle aziende.

### 1.6.3 Sicurezza e fiducia

Siccome il software fa parte della vita di tutti i giorni è essenziale che ci sia fiducia nel software.

### 1.6.4 Scalabilità

Il software deve essere sviluppato a più scale, cioè soluzione che funziona in piccolo deve adattarsi anche a grandi scale senza rischiare di fallire.

## 1.7 Diversità di ingegneria del software

Ci sono diversi tipi di sistemi software e non c'è un insieme universale di tecniche applicabili a tutti i sistemi. Quindi i metodi e gli strumenti utilizzati dipendono dal tipo di applicazione che si deve sviluppare, dalle richieste dei clienti e dalle competenze degli sviluppatori.

## 1.8 Tipi di applicazione

### 1.8.1 Applicazioni stand-alone

Sono applicazioni che si eseguono su un singolo computer locale e includono le funzionalità necessarie per l'utente.

### 1.8.2 Applicazioni interattive transaction-based

Sono applicazioni che sono eseguite su un computer in remoto e sono accessibili dagli utenti dai propri computer. Queste includono le applicazioni web come gli e-commerce.

### **1.8.3 Sistemi embedded**

Sono applicazioni che controllano e gestiscono dispositivi hardware. Questi sistemi sono i più numerosi.

### **1.8.4 Sistemi batch**

Questi sistemi elaborano grandi quantità di dati per produrre un output.

### **1.8.5 Sistemi di intrattenimento**

Questi sistemi sono per uso personale e servono ad intrattenere l'utente.

### **1.8.6 Sistemi di modellazione e simulazione**

Questi sistemi sono sviluppati da scienziati per modellare processi fisici o situazioni che includono tanti oggetti separati che interagiscono tra di loro.

### **1.8.7 Sistemi di collezione di dati (o IOT)**

Questi sistemi raccolgono dati da sensori e li inviano ad un sistema centrale per l'elaborazione.

### **1.8.8 Sistemi di sistemi**

Questi sistemi sono composti da più sistemi software che collaborano tra di loro.

## **1.9 Principi fondamentali**

- I sistemi devono essere sviluppati utilizzando un processo strutturato e ben pensato.
- L'affidabilità e la performance sono importanti per ogni tipo di software.
- Capire e gestire i requisiti del software è essenziale.
- Dove appropriato si dovrebbe riutilizzare software che è già stato sviluppato al posto di svilupparne uno da zero.