

Ingegneria del Software

UniVR - Dipartimento di Informatica

Fabio Irimie

2° Semestre 2024/2025

Indice

1	Introduzione	2
1.1	Cos'è l'ingegneria del software?	2
1.1.1	Economia	2
1.1.2	Fallimento dei progetti	2
1.2	Prodotti software	2
1.2.1	Prodotti generici	2
1.2.2	Prodotti personalizzati	2
1.3	Caratteristiche di un buon software	2
1.4	Problemi che influenzano il software	3
1.5	Tipi di applicazione	3
1.6	Principi fondamentali	4
2	Processi del software	4
2.1	Modelli di processo	5
2.1.1	Modello a cascata	5

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.

1.1.1 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.

1.1.2 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.

1.2 Prodotti software

1.2.1 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.

1.2.2 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.

1.3 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.

1.4 Problemi che influenzano il software

- **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.
- **Cambiamento sociale o del business:** Il software deve essere in grado di adattarsi a cambiamenti sociali o organizzativi nelle aziende.
- **Sicurezza e fiducia:** Siccome il software fa parte della vita di tutti i giorni è essenziale che ci sia fiducia nel software.
- **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.

1.5 Tipi di applicazione

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. I principali tipi di applicazione sono:

- **Applicazioni stand-alone:** Sono applicazioni che si eseguono su un singolo computer locale e includono le funzionalità necessarie per l'utente.
- **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.
- **Sistemi embedded:** Sono applicazioni che controllano e gestiscono dispositivi hardware. Questi sistemi sono i più numerosi.
- **Sistemi batch:** Questi sistemi elaborano grandi quantità di dati per produrre un output.
- **Sistemi di intrattenimento:** Questi sistemi sono per uso personale e servono ad intrattenere l'utente.
- **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.
- **Sistemi di collezione di dati (o IOT):** Questi sistemi raccolgono dati da sensori e li inviano ad un sistema centrale per l'elaborazione.

- **Sistemi di sistemi:** Questi sistemi sono composti da più sistemi software che collaborano tra di loro.

1.6 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.

2 Processi del software

I **processi** del software sono un insieme di attività strutturate che sono necessarie per sviluppare un sistema software. Ci sono diversi tipi di processi, ma tutti coinvolgono le seguenti attività:

- **Specificazione**
- **Sviluppo**
- **Design e implementazione**
- **Validazione**
- **Evoluzione**

Un modello di processo software è una rappresentazione astratta di un processo da un punto di vista particolare.

Per descrivere e discutere i processi di solito si parla delle attività all'interno di questi processi e dell'ordine in cui queste attività vengono svolte. Alcune descrizioni possono anche includere:

- **Prodotti:** cioè i risultati di un'attività.
- **Ruoli:** cioè le responsabilità di un individuo o di un gruppo.
- **Pre e post condizioni:** cioè le condizioni che devono essere soddisfatte prima e dopo un'attività.

Alcuni esempi di processi sono:

- **Processi plan-driven:** sono processi dove tutte le attività sono pianificate in anticipo e tutti i progressi sono misurati rispetto al piano.
- **Processi agili:** sono processi in cui la pianificazione è incrementale ed è più facile adattarsi ai cambiamenti.

Nel pratico si utilizza un approccio ibrido tra i due.

2.1 Modelli di processo

- **Modello a cascata:** è un modello plan-driven con fasi di specificazione e di sviluppo distinte e separate
- **Modello incrementale:** la specifica, lo sviluppo e la validazione sono intercalate. Può essere plan-driven o agile.
- **Integrazione e configurazione:** i sistemi sono sviluppati da componenti già esistenti e configurabili.

Nel pratico, sistemi molto grandi sono sviluppati utilizzando un processo che incorpora elementi di tutti e tre i modelli.

2.1.1 Modello a cascata

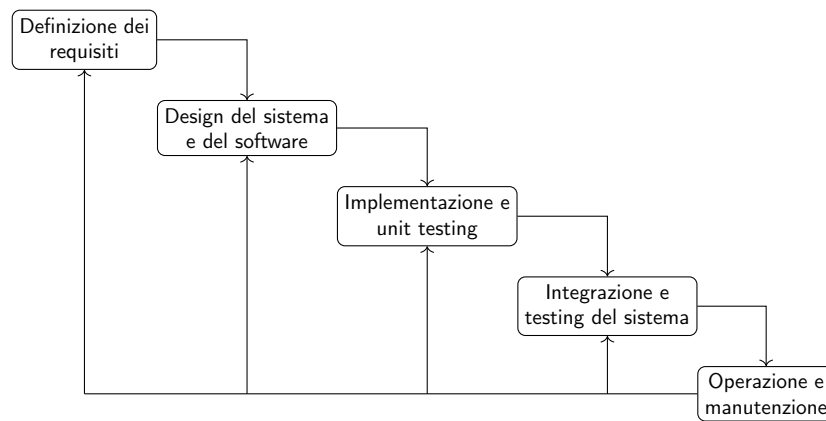


Figura 1: Modello a cascata