

Introduzione

Contenuti da Cap. 1 Ghezzi et al.

Sommario

- Definizione
- Nascita dell' ingegneria del software
- Ruolo
- Relazione con altre discipline

Il software

- Il **software** è definito come:
i programmi, le procedure, e l'eventuale documentazione associata e i dati relativi all'operatività di un sistema di elaborazione.

Definizione di Ingegneria del software

- Nel glossario dell' IEEE (“IEEE Standard Glossary of Software Engineering”, 1990), l' ***ingegneria del software*** è definita come:

applicazione di un approccio sistematico, disciplinato e quantificabile allo sviluppo, all' operatività e alla manutenzione del software.

Campo di applicazione

- **Costruzione di sistemi software**
 - grandi e complessi
 - prodotti da squadre
 - esistenti in diverse versioni
 - in funzione per anni o decenni
 - che subiscono evoluzioni

Storia: anni Cinquanta

- Il software doveva risolvere problemi relativamente semplici e ben compresi
- L'utente e il programmatore erano la stessa persona
- Ingegneria del software = programmazione

Storia: anni Sessanta

- Calo dei costi dell' hardware, maggiore diffusione dei computer
- Software applicato a problemi vari, più complessi, non sempre ben compresi
- Utente != programmatore

Storia: anni Sessanta

- Problemi:
 - difficoltà di comunicazione fra utenti e programmatori, requisiti del software non chiari
 - tecniche di programmazione non adatte a grossi sistemi
 - parti dei sistemi software fortemente accoppiate:
 - necessità di coordinazione continua fra i programmatori
 - difficoltà nella sostituzione di chi abbandonava i progetti
 - la modifica di una parte influenzava l'intero sistema

Storia: anni Sessanta

- Esempio di progetto: sviluppo del sistema operativo OS 360 per i mainframe IBM della famiglia 360
- Viene coniata l'espressione "crisi del software"

Storia: anni Sessanta

- Risultato:
 - sistemi poco soddisfacenti, poco affidabili, che non rispettavano i tempi e i costi previsti
- Si comprende la necessità di un approccio sistematico alla produzione del software
- L'espressione "**ingegneria del software**" nasce in questo periodo

Storia: dagli anni Sessanta a oggi

- L'ingegneria del software ha prodotto:
 - Linguaggi e strumenti
 - Approcci rigorosi (formali) per
 - specifica
 - verifica
 - Standard
 - Metodologie pratiche, organizzative e gestionali

L'ingegneria del software è una disciplina relativamente giovane

- I problemi
 - non esistono parametri universalmente accettati per definire i requisiti
 - le tecniche formali sono poco sviluppate e di applicabilità incerta
 - (quasi) ogni nuovo prodotto pone problemi nuovi
- Intuito ed esperienza giocano ancora un ruolo fondamentale

“No silver bullet”

- In un articolo del 1987 Brooks afferma che non esiste un “proiettile d’argento”, ovvero una soluzione magica ai problemi del software (*dal fatto che i proiettili d’argento uccidono i lupi mannari*)
- La progettazione e lo sviluppo del software richiedono **sforzo intellettuale, creatività e tempo**

Newsroom

Press Release

Share: [Tweet](#) [in Share](#) 339 [G+](#)

STAMFORD, Conn., July 13, 2017

[View All Press Releases](#)

Gartner Says Worldwide IT Spending Forecast to Grow 2.4 Percent in 2017

Analysts Discuss Latest IT Spending Outlook During Gartner Webinar

Worldwide IT spending is projected to total \$3.5 trillion in 2017, a 2.4 percent increase from 2016, according to Gartner, Inc. This growth rate is up from the previous quarter's forecast of 1.4 percent, due to the U.S. dollar decline against many foreign currencies (see Table 1.)

"Digital business is having a profound effect on the way business is done and how it is supported," said [John-David Lovelock](#), vice president and distinguished analyst at Gartner. "The impact of digital business is giving rise to new categories; for example, the convergence of "software plus services plus intellectual property." These next-generation offerings are fueled by business and technology platforms that will be the driver for new categories of spending. Industry-specific disruptive technologies include the [Internet of Things \(IoT\)](#) in manufacturing, [blockchain](#) in financial services (and other industries), and smart machines in retail. The focus is on how technology is disrupting and enabling business."

The [Gartner Worldwide IT Spending Forecast](#) is the leading indicator of major technology trends across the hardware, software, IT services and telecom markets. For more than a decade, global IT and business executives have been using these highly anticipated quarterly reports to recognize market opportunities and challenges, and base their critical business decisions on proven methodologies rather than guesswork.

Dimensione economica

<http://www.gartner.com/newsroom/id/3759763>

Table 1. Worldwide IT Spending Forecast (Billions of U.S. Dollars)

	2016 Spending	2016 Growth (%)	2017 Spending	2017 Growth (%)	2018 Spending	2018 Growth (%)
Data Center Systems	170	-0.3	171	0.3	173	1.2
Enterprise Software	326	5.3	351	7.6	381	8.6
Devices	630	-2.4	654	3.8	677	3.6
IT Services	894	3.2	922	3.1	966	4.7
Communications Services	1,374	-1.3	1,378	0.3	1,400	1.6
Overall IT	3,396	0.3	3,477	2.4	3,598	3.5

Source: Gartner (July 2017)

The worldwide [enterprise software](#) market is forecast to grow 7.6 percent in 2017, up from 5.3 percent growth in 2016. As software applications allow more organizations to derive revenue from digital business channels, there will be a stronger need to automate and release new applications and functionality.

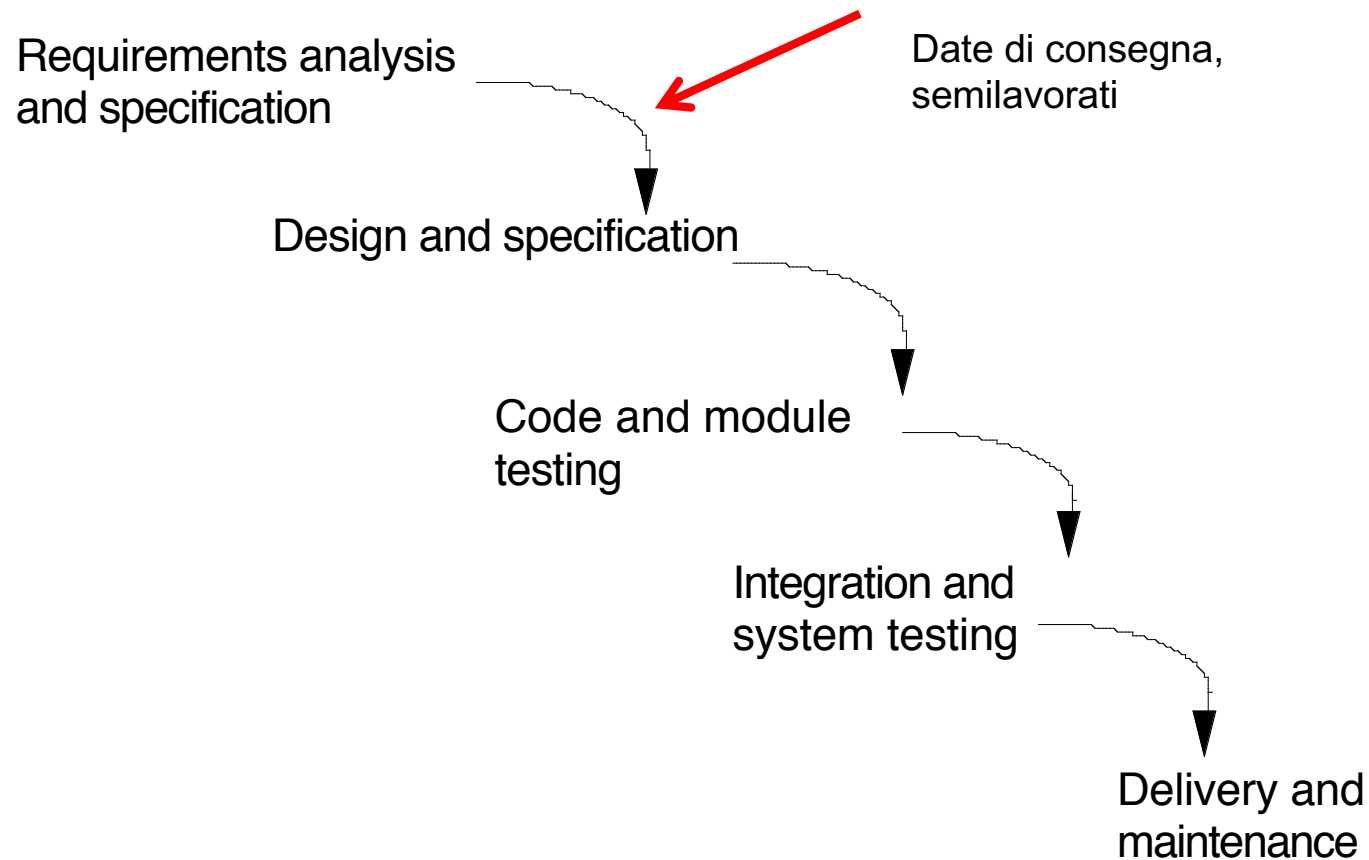
Ruolo dell'ingegnere del software

- Non solo programmare:
 - comprendere i requisiti e tradurli in specifiche precise
 - progettare (architettura dei componenti)
 - operare a diversi livelli di astrazione
 - comunicare con la squadra, gli utenti, i clienti
 - rispettare tempi e costi

Attività nell'ingegneria del software

- Analisi e specifica dei requisiti
- Progettazione e specifica di sistema
- Codifica e verifica di modulo
- Integrazione e verifica di sistema
- Consegna e manutenzione

Modello a cascata del ciclo di vita del software



Analisi e specifica dei requisiti

- Dopo studio di fattibilità (*costi/benefici, diversi scenari e soluzioni*)
- Obiettivo: identificare e documentare i requisiti che il sistema software dovrà rispettare (*che cosa*)
- In caso di sistemi innovativi, richiede ampia interazione fra cliente e ingegnere
- Il documento dei requisiti dovrebbe
 - essere comprensibile per l'utente finale
 - poter portare alla creazione del manuale utente e di casi di test

Progettazione e specifica di sistema

- Obiettivo: progetto di un particolare sistema che soddisfi i requisiti (*come*)
- Divisa in due fasi:
 - Progetto architettuale: organizzazione globale del sistema in componenti di alto livello e loro interazioni
 - Progetto dettagliato: scomposizione a livelli di dettaglio sempre maggiori, fino a una specifica diretta della codifica

Codifica e verifica di modulo

- Produzione del codice sorgente secondo il progetto dettagliato del sistema
- Verifica che i singoli moduli prodotti rispettino la loro specifica

Integrazione e verifica di sistema

- I singoli moduli prodotti vengono
 - assemblati
 - integrati
 - testati come sistema unico

Consegna e manutenzione

- Quando il sistema ha superato tutti i test viene consegnato all'utente
- Successivamente, si rende necessaria la manutenzione del sistema:
 - per correggere i difetti che si manifestano
 - per adattare il sistema a cambiamenti dei requisiti (evoluzione)

Modello a cascata del ciclo di vita del software

Requirements analysis and specification

Design and specification

Code and module testing

Integration and system testing

Delivery and maintenance

- A volte è necessario ritornare ai passi precedenti
- A volte le fasi sono parzialmente sovrapposte

Relazione con altri campi dell'informatica

- Linguaggi di programmazione
- Sistemi operativi
- Basi di dati
- Intelligenza artificiale
- Metodi formali

Relazione con linguaggi di programmazione

- Forte influenza dell'ingegneria del software sui linguaggi di programmazione: ad esempio,
 - Costrutti per programmazione modulare
 - Costrutti per la gestione delle eccezioni
 - Separazione tra interfaccia e implementazione (Java, etc)

Relazione con linguaggi di programmazione

- Influenza inversa:
 - Specifiche di progetto rigorose, adatte all'implementazione in un linguaggio di programmazione
 - Formalizzazione del linguaggio con cui si scrive l'input di un software (da Job Control Language di OS 360 a linguaggi a linea di comando alla Unix)
 - Formalizzazione per favorire l'automazione (analizzatori lessicali, parser)

Sistemi operativi

- Hanno avuto influenza notevole sull'ingegneria del software:
 - primi sistemi di grandi dimensioni
 - macchine virtuali, livelli di astrazione, separazione fra politiche e meccanismi
- Influenza inversa:
 - modularità e portabilità dei sistemi operativi moderni
 - ad es., separazione dell'interprete dei comandi
 - architetture a micro-kernel

Basi di dati

- Hanno influenzato l'ingegneria del software:
 - indipendenza dalla rappresentazione dei dati (astrazione, separazione degli interessi)
 - uso di DBMS come componente risolve “gratis” i problemi legati all'accesso concorrente a grosse quantità di dati

Basi di dati

- **Influenza inversa:**
 - necessità di nuovi modelli per usare DBMS come strumenti per l'ingegneria del software
 - memorizzazione di codice sorgente, documenti, file binari
 - memorizzazione di diverse versioni dello stesso oggetto
 - transazioni lunghe

Intelligenza artificiale

- **Influenza su ingegneria del software**
 - Tecniche per produrre sistemi software grossi e complessi con requisiti incerti (sviluppo esplorativo)
 - Utilizzo della logica
- **Influenza inversa**
 - Sistemi di IA modulari (a regole o shell)
 - Concezione di tecniche di intelligenza artificiale da applicare a ingegneria del software
 - Assistenti di programmazione
 - Elaborazione del linguaggio naturale per le interfacce

Sistemi formali

- Modelli formali utilizzati in ingegneria del software (specifiche e modelli)
 - automi a stati finiti
 - automi a pila
 - reti di Petri
 - logica matematica
- Inversamente: sviluppo di nuovi metodi formali:
 - specifiche algebriche
 - tipi di dato astratto
 - logica temporale

Relazione con altre discipline

- Scienze organizzative
 - Modelli gestionali applicati a produzione del software per effettuare stime, gestire le risorse umane, monitorare il processo
 - Necessità di nuovi modelli
- Ingegneria dei sistemi
 - Software come componente di un sistema più complesso