



INGEGNERIA DEL SOFTWARE - 2025-26

CONCETTI FONDAMENTALI DELL'INGEGNERIA DEL SOFTWARE

LEZIONE 1
29/09/2025
VINCENZO RICCIO

RIFERIMENTI

- ▶ Sommerville - Capitolo 1.1



Vincenzo Riccio
Ingegneria del Software 2025/2026
Università degli Studi di Udine

OBIETTIVI DELLA LEZIONE

Vincenzo Riccio
Ingegneria del Software 2025/2026
Università degli Studi di Udine



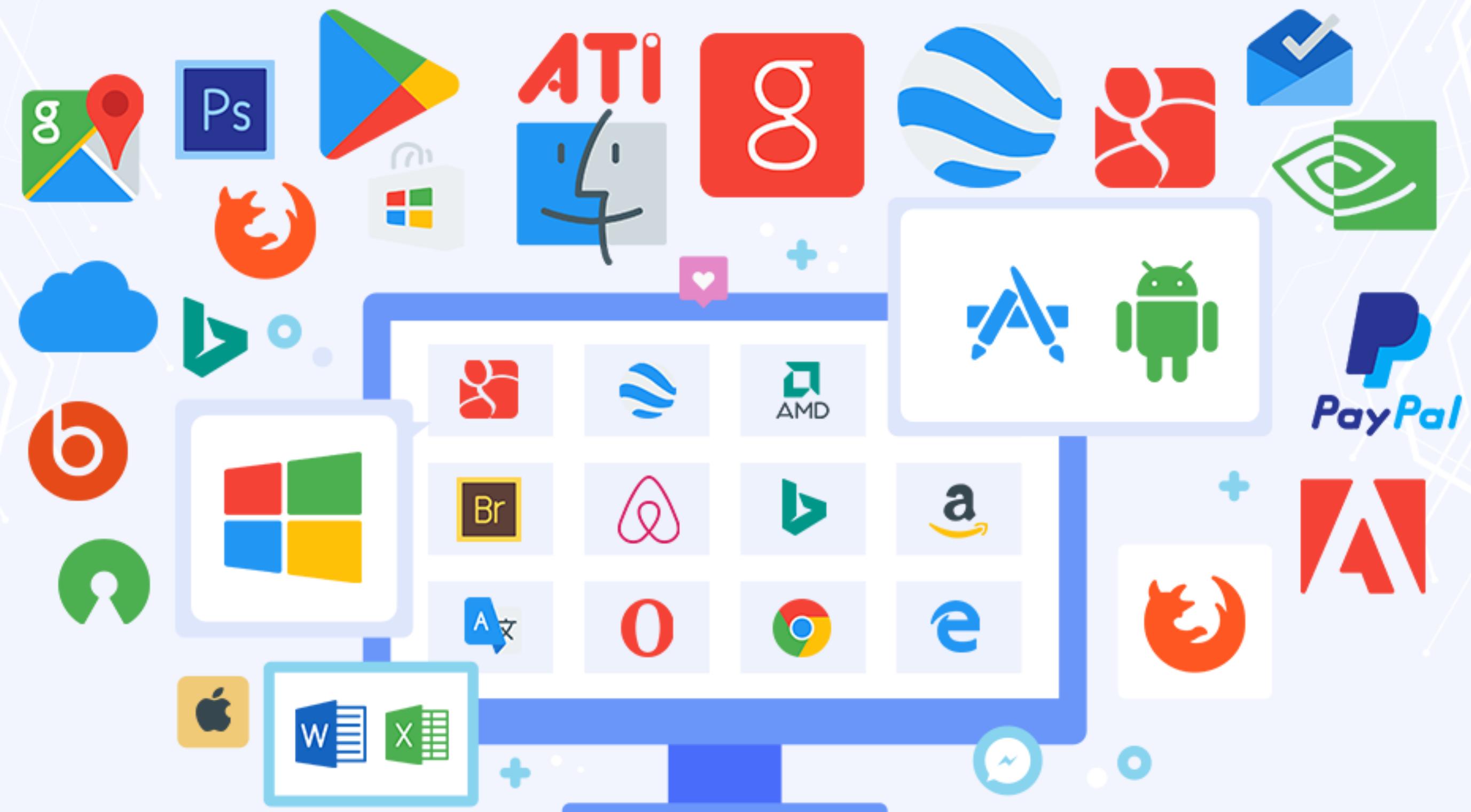
- ▶ Introdurre l'Ingegneria del Software e spiegarne l'importanza
- ▶ Rispondere ad alcune domande chiave relative all'Ingegneria del Software
- ▶ Comprendere le sfide principali allo sviluppo di software di qualità



COS'È IL SOFTWARE

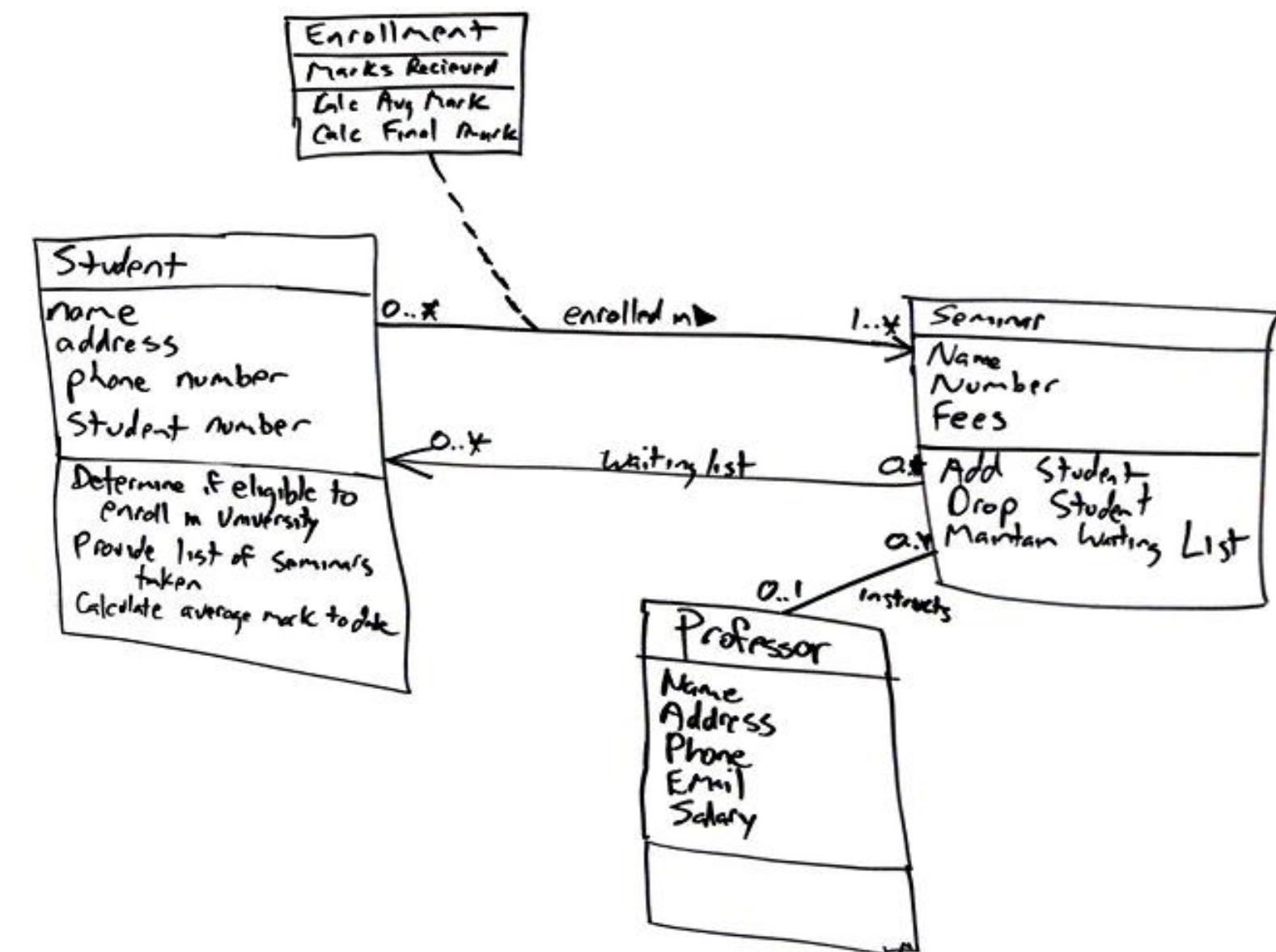


- Software è un termine che indica l'insieme dei **programmi** per computer e la relativa **documentazione**





- ▶ Software è un termine che indica l'insieme dei **programmi** per computer e la relativa **documentazione**
- ▶ Non solo programmi
ma anche la documentazione prodotta durante il loro sviluppo, come **modelli di progetto, manuali utente, siti web di supporto**



SOFTWARE: DEFINIZIONE IEEE

Vincenzo Riccio
Ingegneria del Software 2025/2026
Università degli Studi di Udine



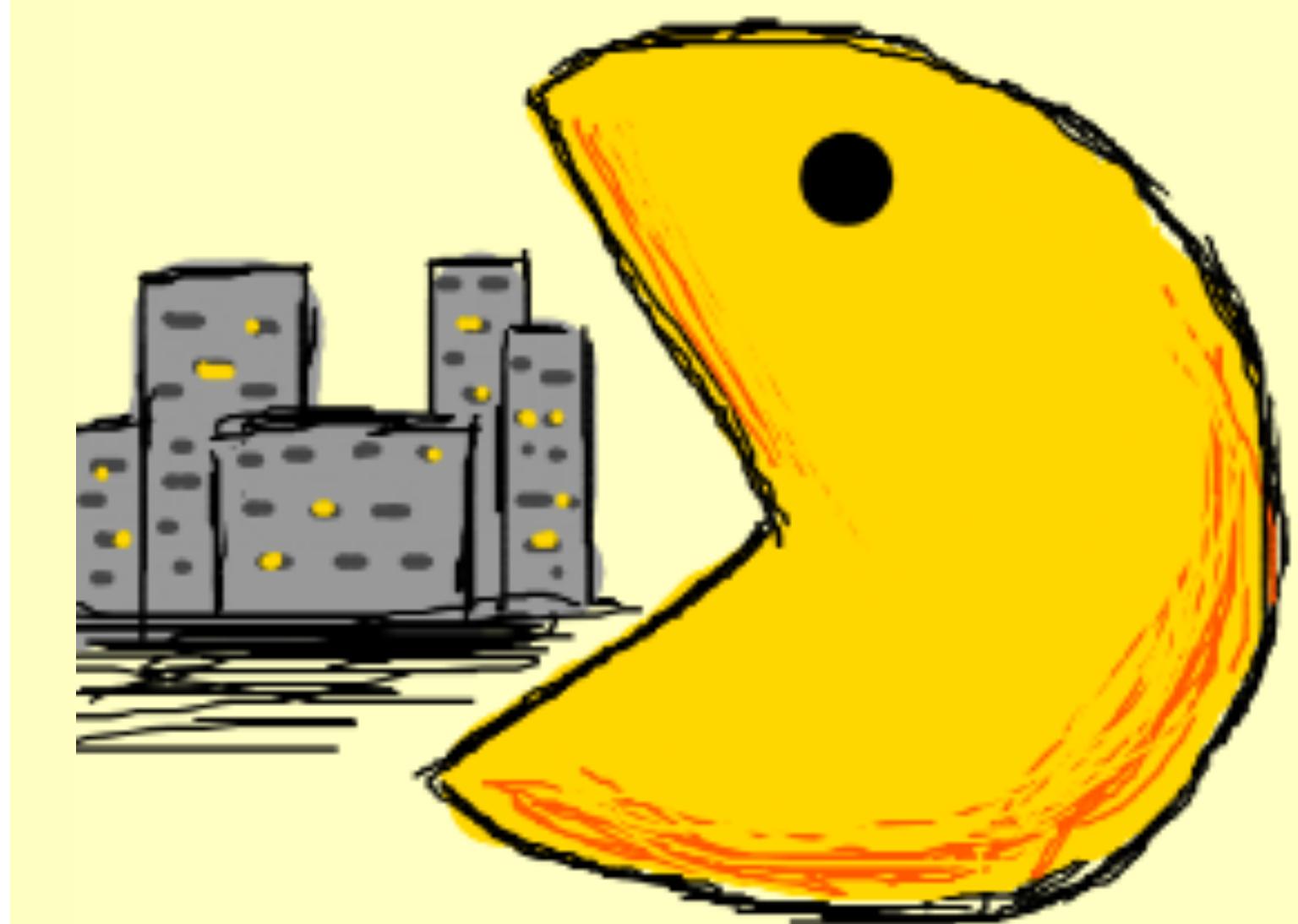
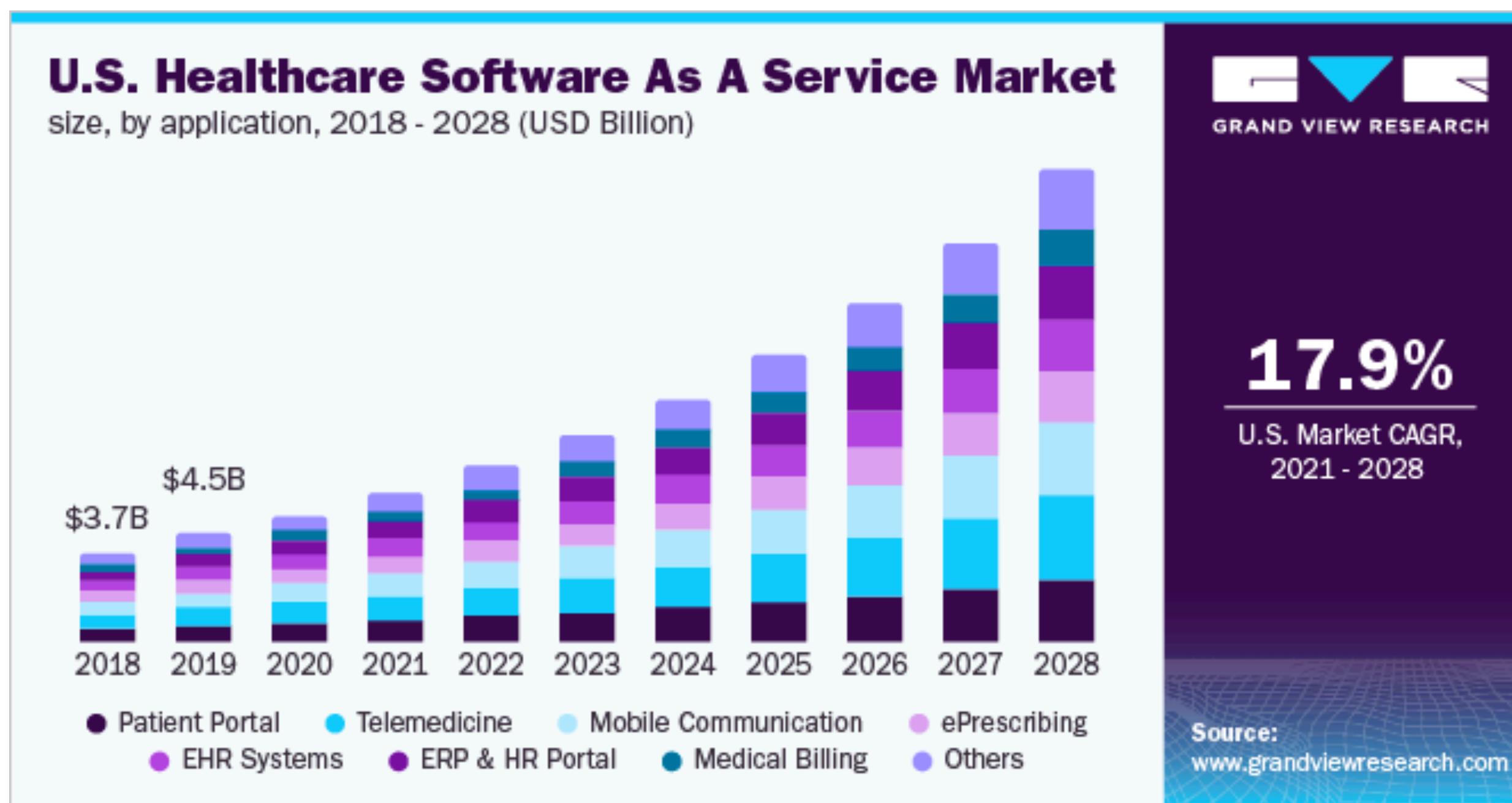
- IEEE (Institute of Electrical and Electronic Engineers) definisce il software come:

*“insieme di programmi, procedure,
regole, e ogni altra documentazione
relativa al funzionamento di un sistema
di elaborazione dati”*



SOFTWARE IS EATING THE WORLD [ANDREESSEN, 2011]

- ▶ Le economie di tutte le nazioni industrializzate dipendono dal software
- ▶ Sempre più sistemi sono controllati dal software (ad es. assistenza sanitaria)



Sanshee @DrawCeption

SOFTWARE IS EATING THE WORLD [ANDREESSEN, 2011]

Vincenzo Riccio
Ingegneria del Software 2025/2026
Università degli Studi di Udine

- ▶ I ricavi generati dal software sono in continua ascesa nel mercato globale (previsti 8.8 miliardi di USD nel 2028) ed una crescita del 17% nel mercato italiano nel 2028 [fonte Statista]





**QUAL È LA SFIDA
PRINCIPALE ALLO
SVILUPPO DI
SOFTWARE DI
ALTA QUALITÀ?**



Rispondete su [menti.com](https://www.menti.com), codice 1372 3211



TIPOLOGIE DI SOFTWARE PROFESSIONALE

SOFTWARE GENERICO VS SU RICHIESTA

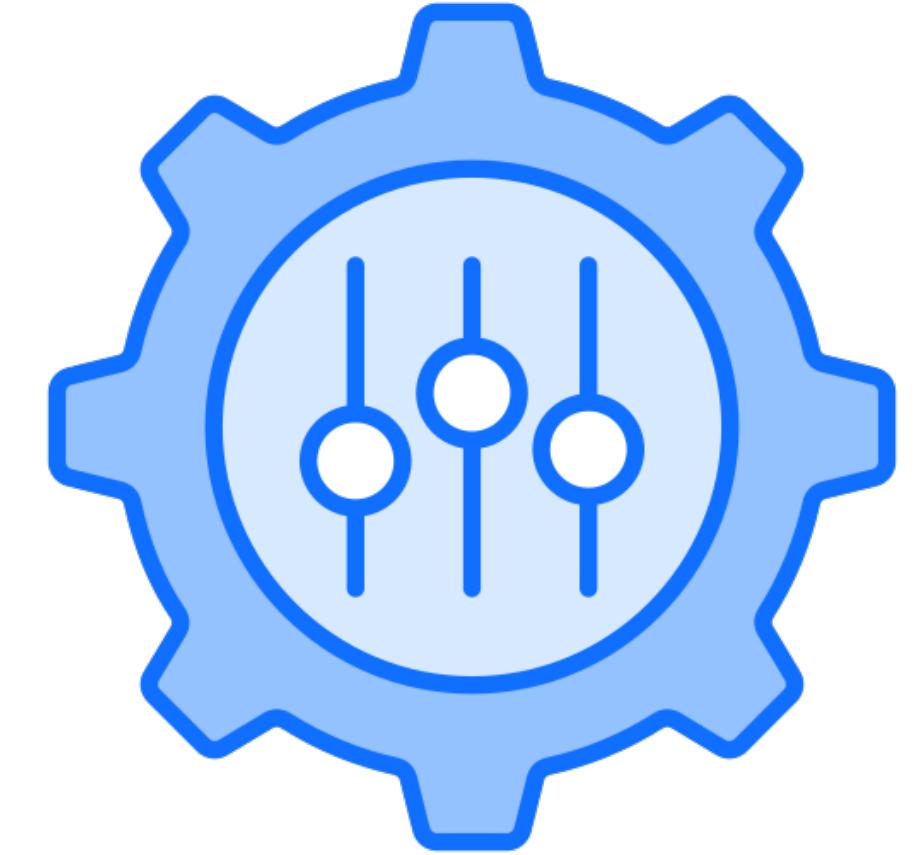


- ▶ **Software generico:**
 - Software prodotto autonomamente da una organizzazione, per incontrare le necessità svariati clienti (ad es. app per smartphone presenti sugli Store ufficiali)
 - Il produttore ha controllo sulle specifiche del software
- ▶ **Software su richiesta:**
 - Software sviluppato da un'organizzazione su commissione di uno specifico cliente (ad es. sistemi di controllo di dispositivi elettronici)
 - Il produttore deve attenersi alle specifiche indicate dal cliente

SOFTWARE PERSONALIZZABILE E/O CONFIGURABILE



- ▶ Sistema generico che può essere adattato alle richieste di un particolare cliente (via di mezzo tra generico e su richiesta)
- ▶ Esempio: SAP è un ERP (Enterprise Resource Planning, pianificazione delle risorse aziendali), ossia un sistema generico, utile a più aziende. Esso viene adattato a una specifica azienda inserendo le informazioni relative alle regole e ai processi aziendali, ai report richiesti ecc...





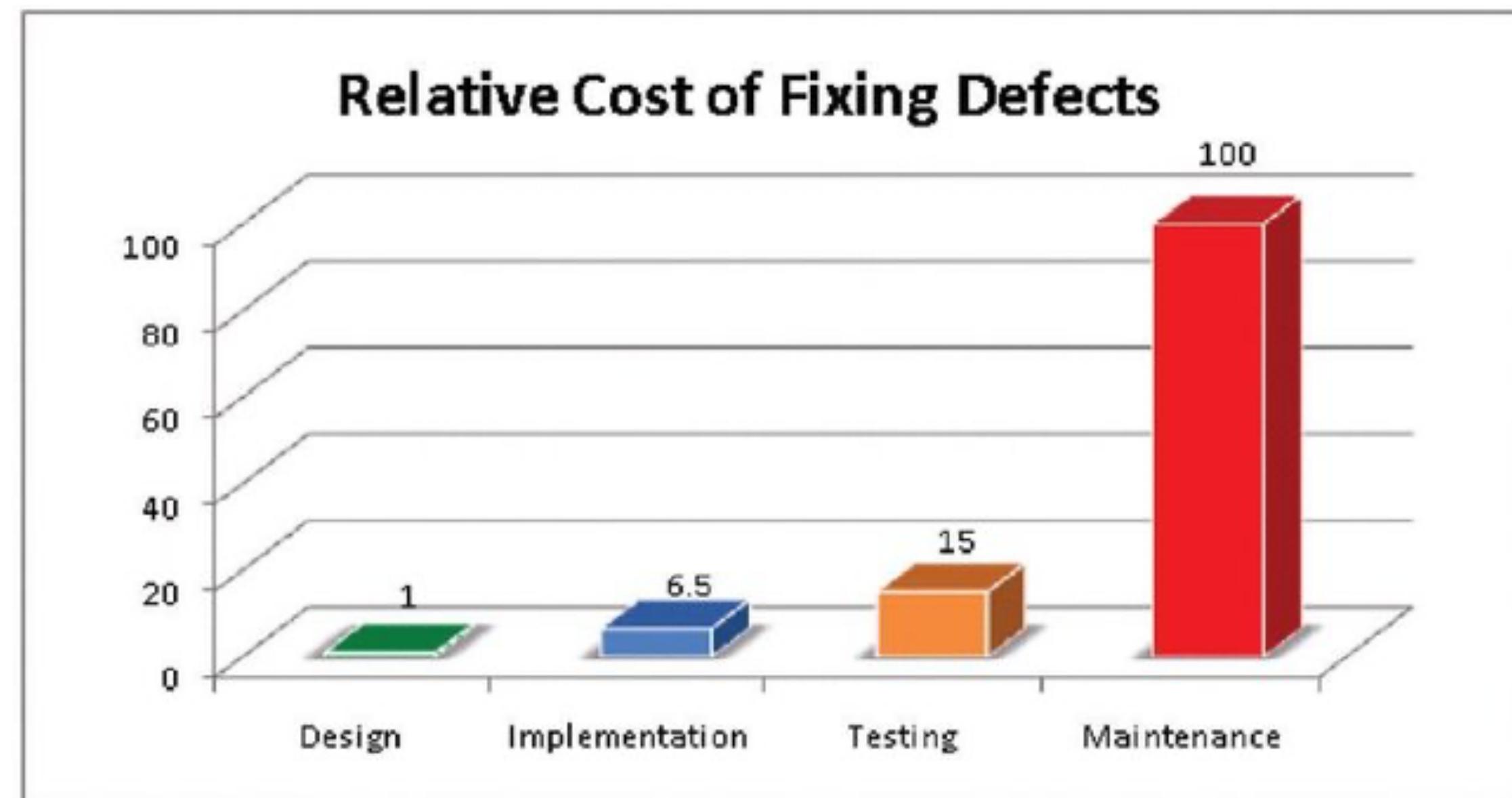
COMPLESSITÀ E COSTI DEL SOFTWARE

COMPLESSITÀ E COSTI DEL SOFTWARE

Vincenzo Riccio
Ingegneria del Software 2025/2026
Università degli Studi di Udine



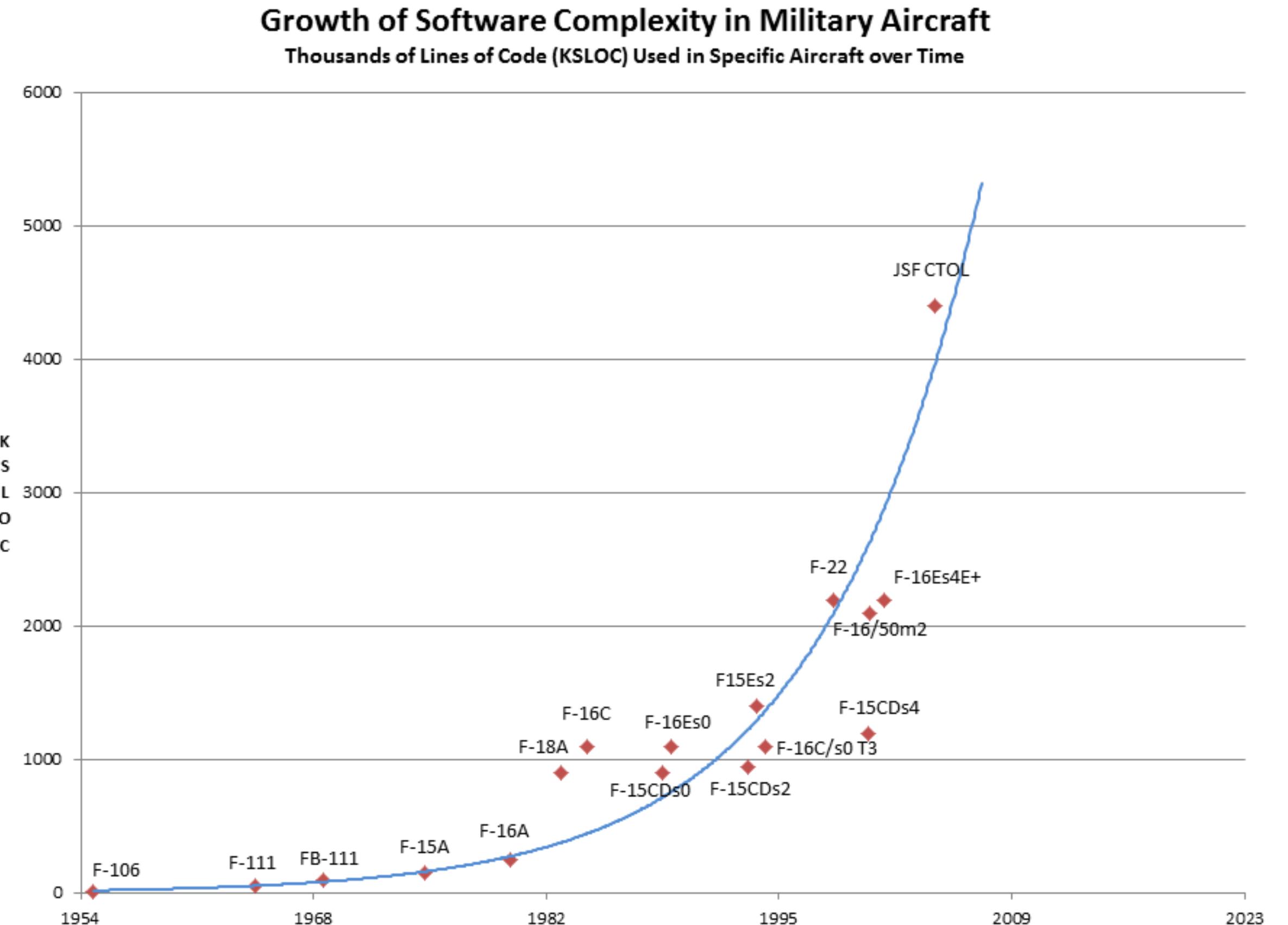
- Il software può diventare estremamente complesso, difficile da capire e costoso da modificare (ad es. in presenza di difetti)



COMPLESSITÀ E COSTI DEL SOFTWARE



- ▶ Nuove tecniche e tecnologie permettono di sviluppare sistemi software sempre più grandi e più complessi, che integrano nuove funzionalità che prima si ritenevano impossibili da realizzare

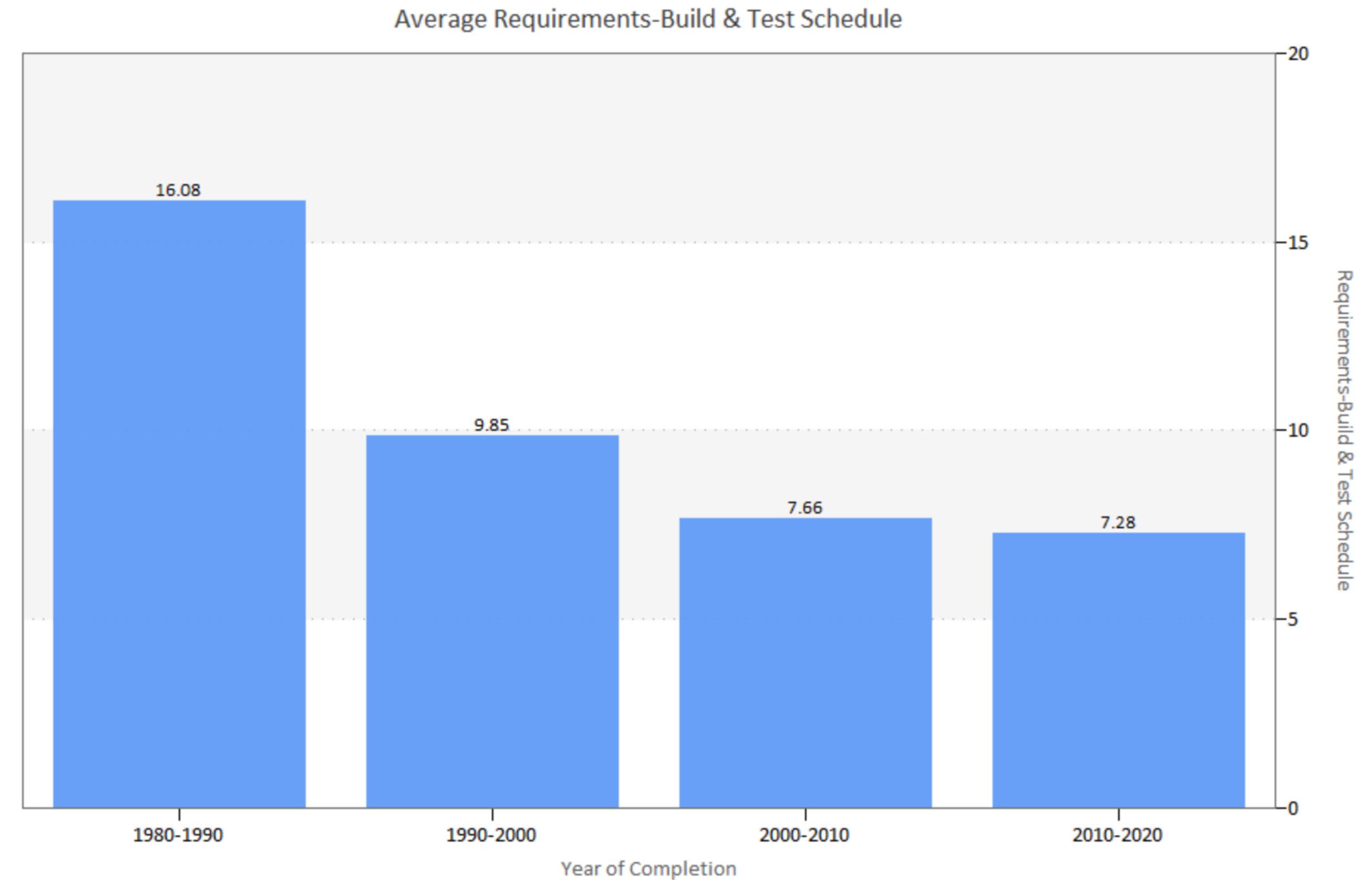


Fonte: SAVI-AVSI (2023)

COMPLESSITÀ E COSTI DEL SOFTWARE



- ▶ Nonostante la maggiore complessità, lo sviluppo software richiede sempre meno tempo
- ▶ Il mercato in continua evoluzione richiede un rilascio rapido dei sistemi
- ▶ D'altra parte, questo miglioramento è legato all'applicazione di approcci dell'Ingegneria del Software

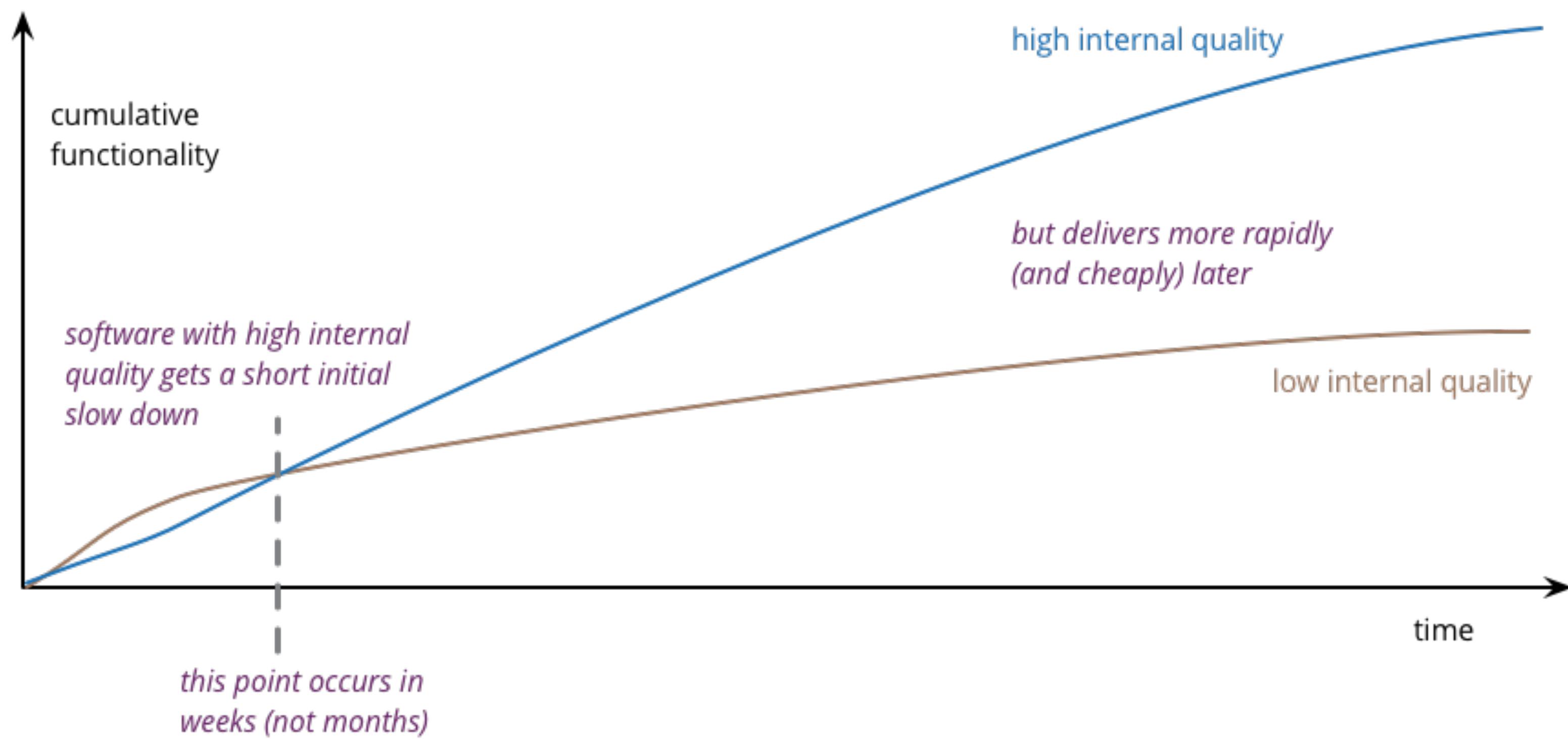


Fonte: QSM (2023)

COMPLESSITÀ E COSTI DEL SOFTWARE



- Il successo o l'insuccesso di un progetto software è spesso legato all'applicazione dell'ingegneria del software per garantire la qualità del software



Fonte: <https://martinfowler.com/articles/is-quality-worth-cost.html>



SVILUPPO PROFESSIONALE DI SOFTWARE DI QUALITÀ

SVILUPPO PROFESSIONALE DEL SOFTWARE



- ▶ **Sviluppo personale** (ad es. software sviluppato durante l'apprendimento delle basi dell'Informatica)
 - Difficilmente il software sarà riusato in futuro da altri utenti
 - Non necessario scrivere una guida, né un documento di progetto
 - Non necessario raggiungere alti livelli di qualità
- ▶ **Sviluppo professionale**
 - Il software sarà usato da altre persone diverse dai propri sviluppatori
 - Sviluppo in team e, quindi, probabilmente il software sarà modificato e manutenuto da altri sviluppatori
 - Il software deve possedere caratteristiche di qualità

QUALITÀ DEL SOFTWARE



- ▶ Un software di qualità deve fornire non solo le funzionalità ma anche le prestazioni richieste
- ▶ Le caratteristiche qualitative di un software professionale sono:
 - **Accettabilità:** il software deve essere accettato dai suoi utenti. Deve essere comprensibile, usabile e compatibile con gli altri sistemi utilizzati dagli utenti
 - **Fidatezza e Protezione:** il software non deve causare danni fisici o economici in caso di malfunzionamento. Utenti malintenzionati non devono poter accedere o danneggiare il sistema.
 - **Efficienza:** il software non deve sprecare le risorse del sistema
 - **Mantenibilità:** il software deve poter evolvere per soddisfare le nuove richieste dei clienti

QUALITÀ DEL SOFTWARE



- ▶ L'insieme specifico dei caratteri qualitativi di un sistema software dipende, ovviamente, dalla sua applicazione
 - Un sistema di controllo aereo deve essere sicuro
 - Un gioco interattivo deve rispondere velocemente
 - Un software che risiede su piccoli processori embedded deve essere molto efficiente
 - Un software utilizzato in una grande azienda con sedi dislocate deve essere mantenibile per poter soddisfare le nuove esigenze del mercato



INGEGNERIA DEL SOFTWARE

FALLIMENTI “CELEBRI” DEL SOFTWARE

Vincenzo Riccio
Ingegneria del Software 2025/2026
Università degli Studi di Udine



- ▶ ARIANE 5, Flight 501 crash (1996)
- ▶ Razzo disintegrato dopo 39 secondi dal decollo
- ▶ Meccanismo di auto-distruzione erroneamente attivato a causa di dati scorretti provenienti dall'Inertial Reference System
- ▶ Costo: 500 milioni di \$

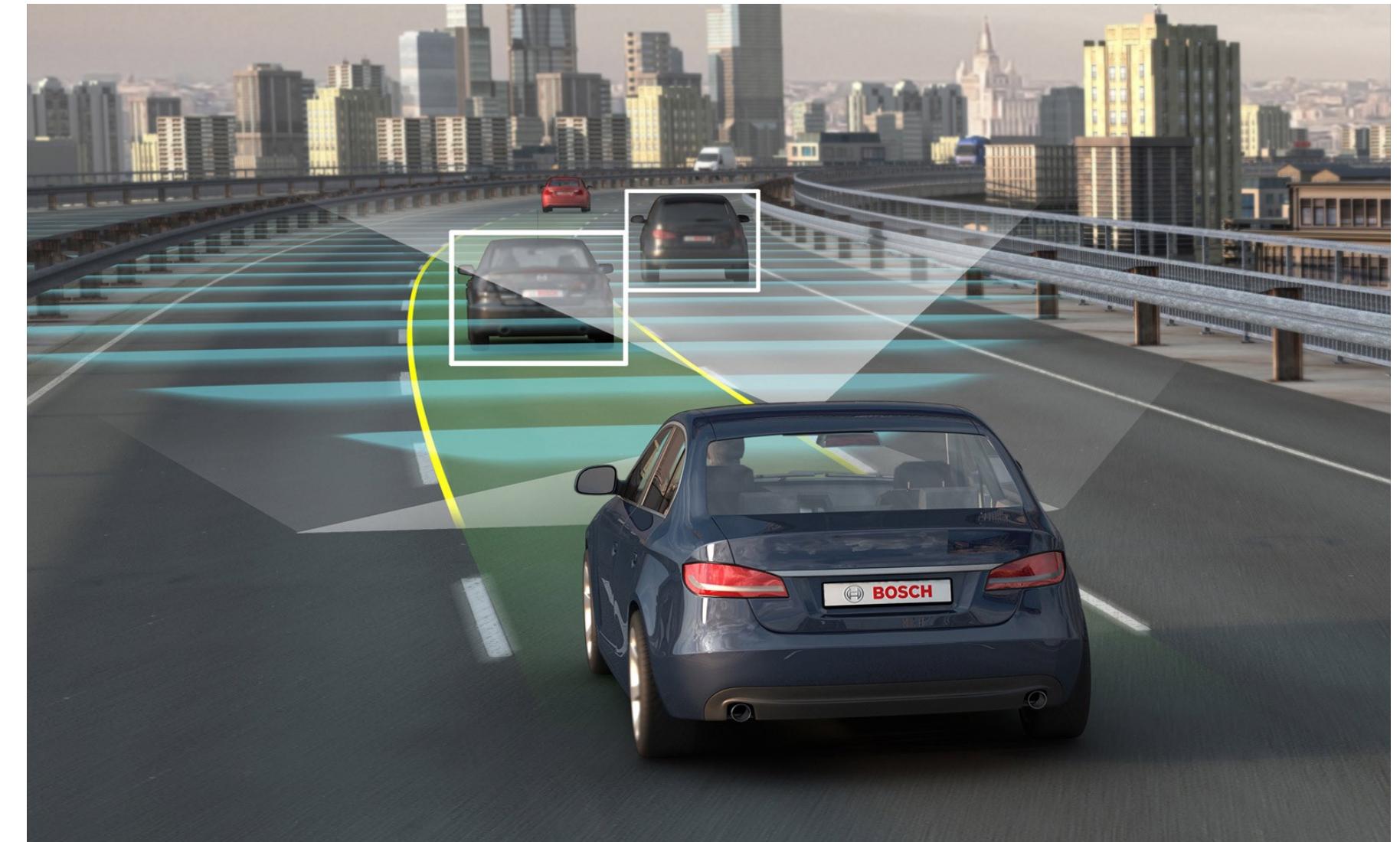
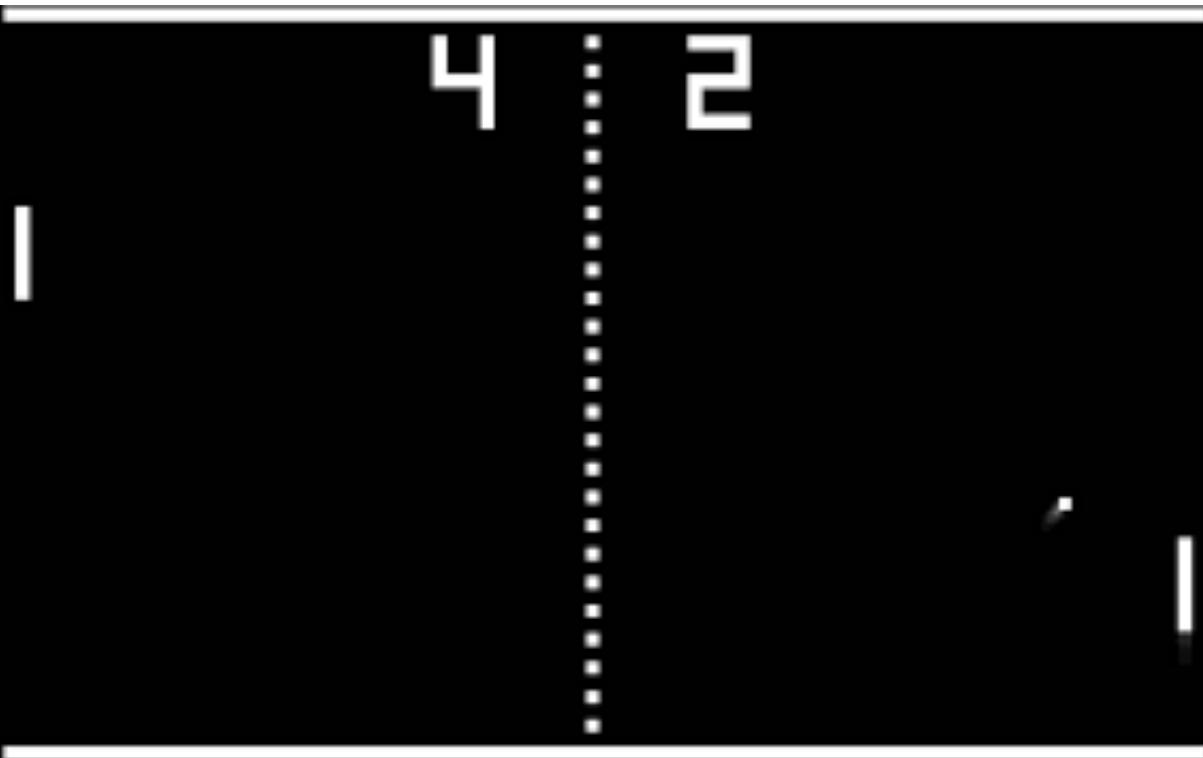


CAUSE DEI FALLIMENTI DEL SOFTWARE

Vincenzo Riccio
Ingegneria del Software 2025/2026
Università degli Studi di Udine

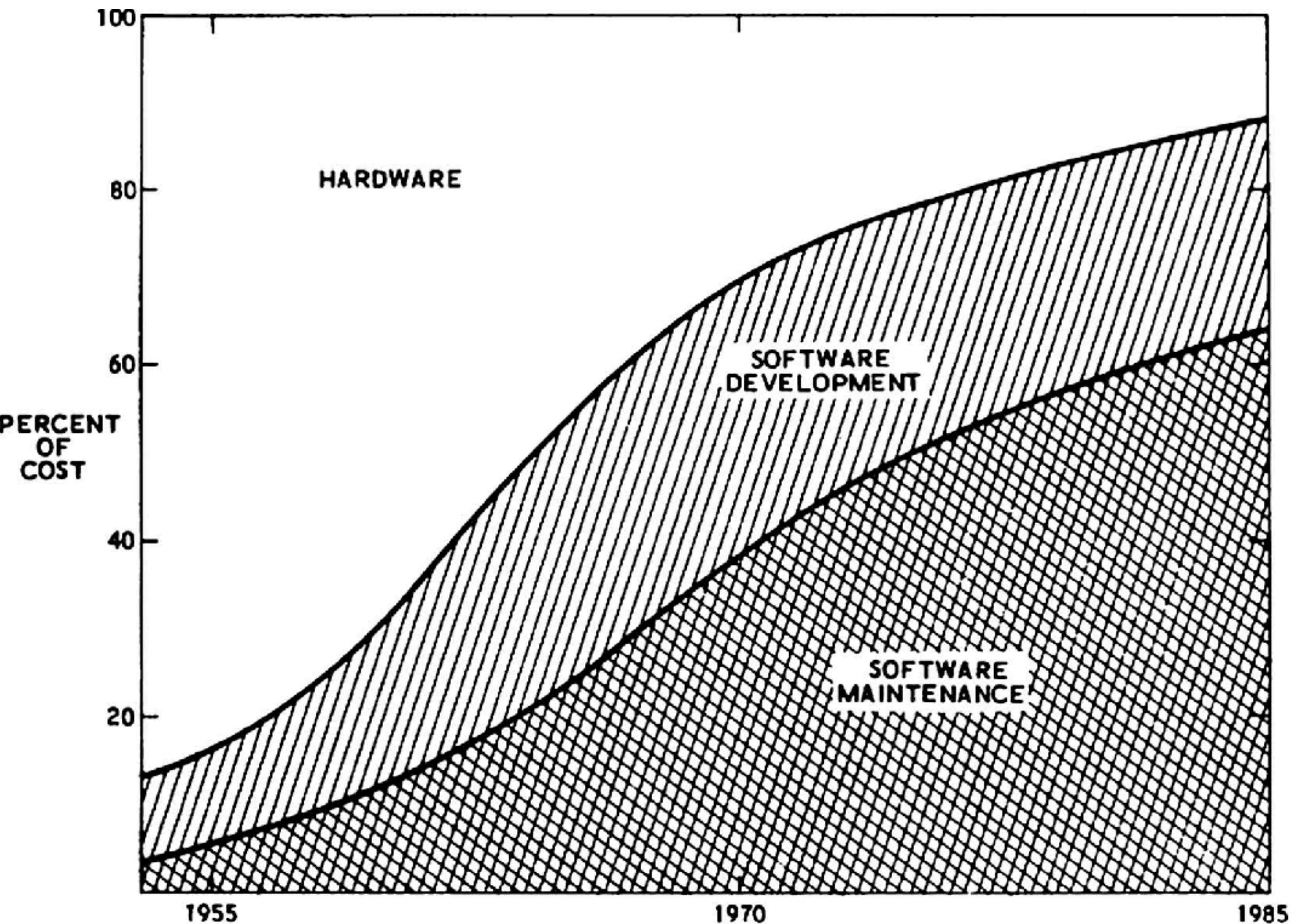


- ▶ Complessità dei sistemi software crescente
- ▶ Sviluppo senza adottare un approccio ingegneristico (ad es. Testing)
- ▶ Richiesta elevata con tempi di rilascio estremamente brevi



NASCITA DELL'INGEGNERIA DEL SOFTWARE

- ▶ Negli anni 70 del secolo scorso, i costi del software in un PC iniziavano a dominare sui costi dell'hardware
- ▶ I costi di manutenzione iniziavano ad essere maggiori di quelli di sviluppo
- ▶ **Problema:** sviluppare software facilmente mantenibile in maniera economicamente vantaggiosa (cost-effective)



NASCITA DELL'INGEGNERIA DEL SOFTWARE

Vincenzo Riccio
Ingegneria del Software 2025/2026
Università degli Studi di Udine



- ▶ Nel Gennaio del 1969 a Garmisch (Germania) si discusse per la prima volta della Crisi del Software
- ▶ Sistemi software erano **sempre più complessi, difficili da manutenere, inaffidabili, più costosi del previsto, rilasciati in ritardo**





- ▶ **Soluzione alla crisi:** adottare approcci ingegneristici alla produzione del software
- ▶ Definizione IEEE:

"Software Engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software"



DEFINIZIONE DI INGEGNERIA DEL SOFTWARE

*“L’Ingegneria del Software
è una **disciplina**
ingegneristica che
riguarda tutti gli aspetti
della produzione del
software”*

Vincenzo Riccio
Ingegneria del Software 2025/2026
Università degli Studi di Udine

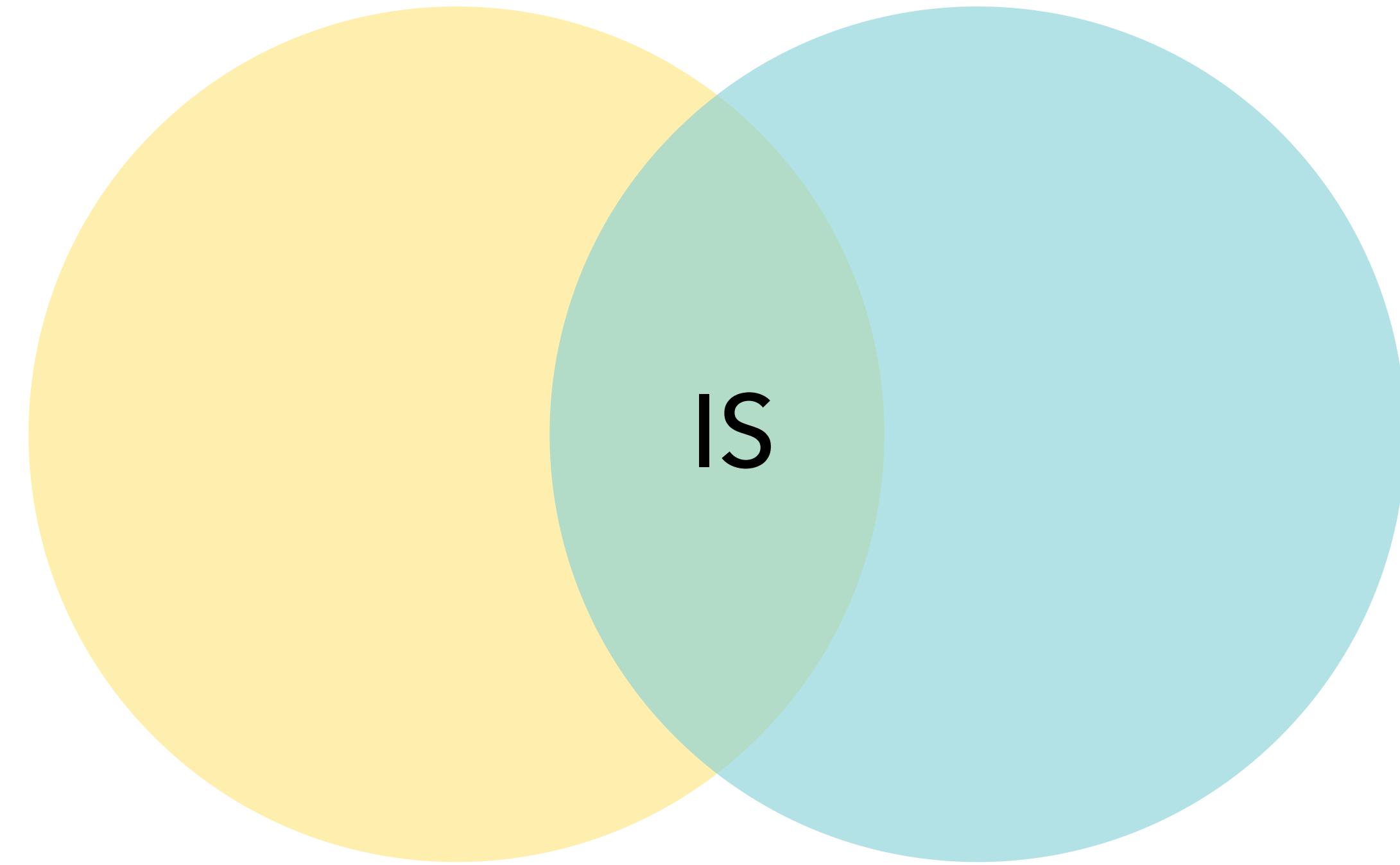


INGEGNERIA DEL SOFTWARE: TRA INGEGNERIA E INFORMATICA

Vincenzo Riccio
Ingegneria del Software 2025/2026
Università degli Studi di Udine



- ▶ L'Ingegneria del Software (IS) è una disciplina ingegneristica
- ▶ IS complementa le competenze informatiche occupandosi dei problemi pratici relativi alla produzione di software
- ▶ IS supporta lo sviluppo di software professionale

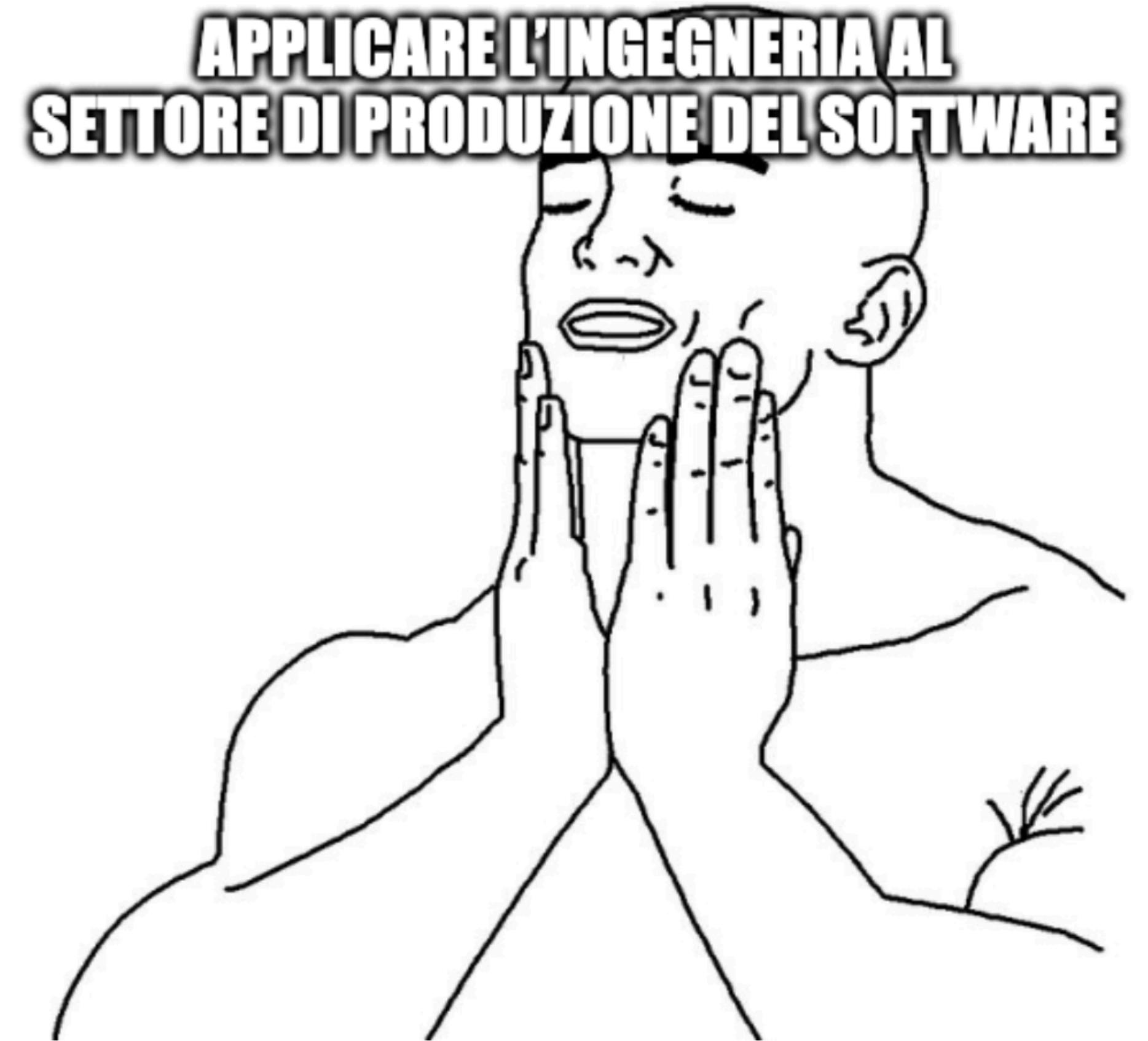


INGEGNERIA DEL SOFTWARE: TRA INGEGNERIA E INFORMATICA

Vincenzo Riccio
Ingegneria del Software 2025/2026
Università degli Studi di Udine

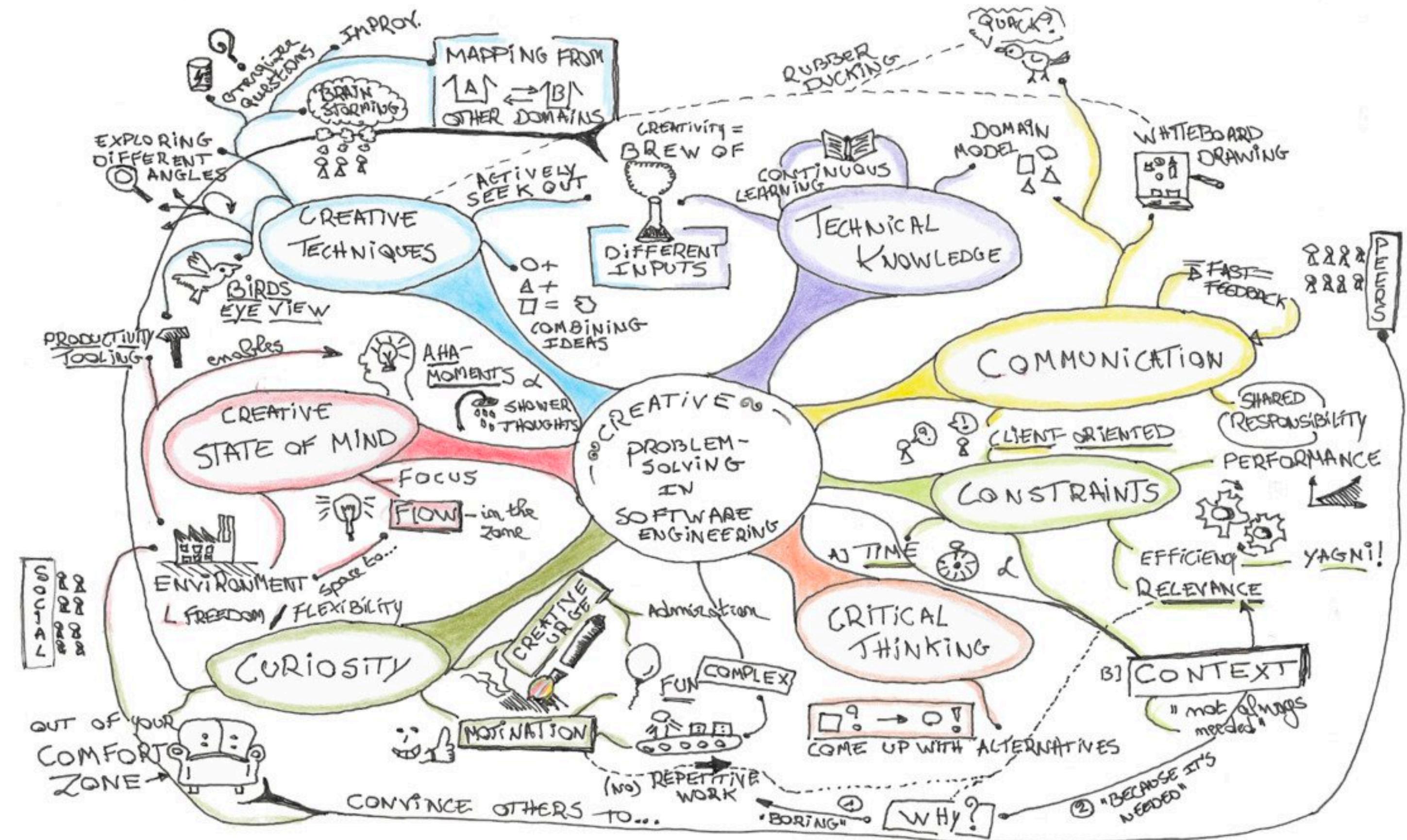


- ▶ Si occupa di tutti gli aspetti della produzione di software, che vanno oltre la programmazione/sviluppo, come specifica, progettazione, evoluzione, manutenzione dei programmi dopo che sono entrati in uso
- ▶ Non basta far funzionare il programma, bisogna anche rispettare vincoli organizzativi e di budget (tempo e denaro)



ORGANIZZAZIONE E CREATIVITÀ

- Spesso sono richiesti compromessi, imposti da vincoli
- Combinazione tra un approccio sistematico e organizzato (processo software) e capacità creative (scelta del metodo più adatto tra le possibili alternative)





PROCESSI SOFTWARE

PROCESSO SOFTWARE

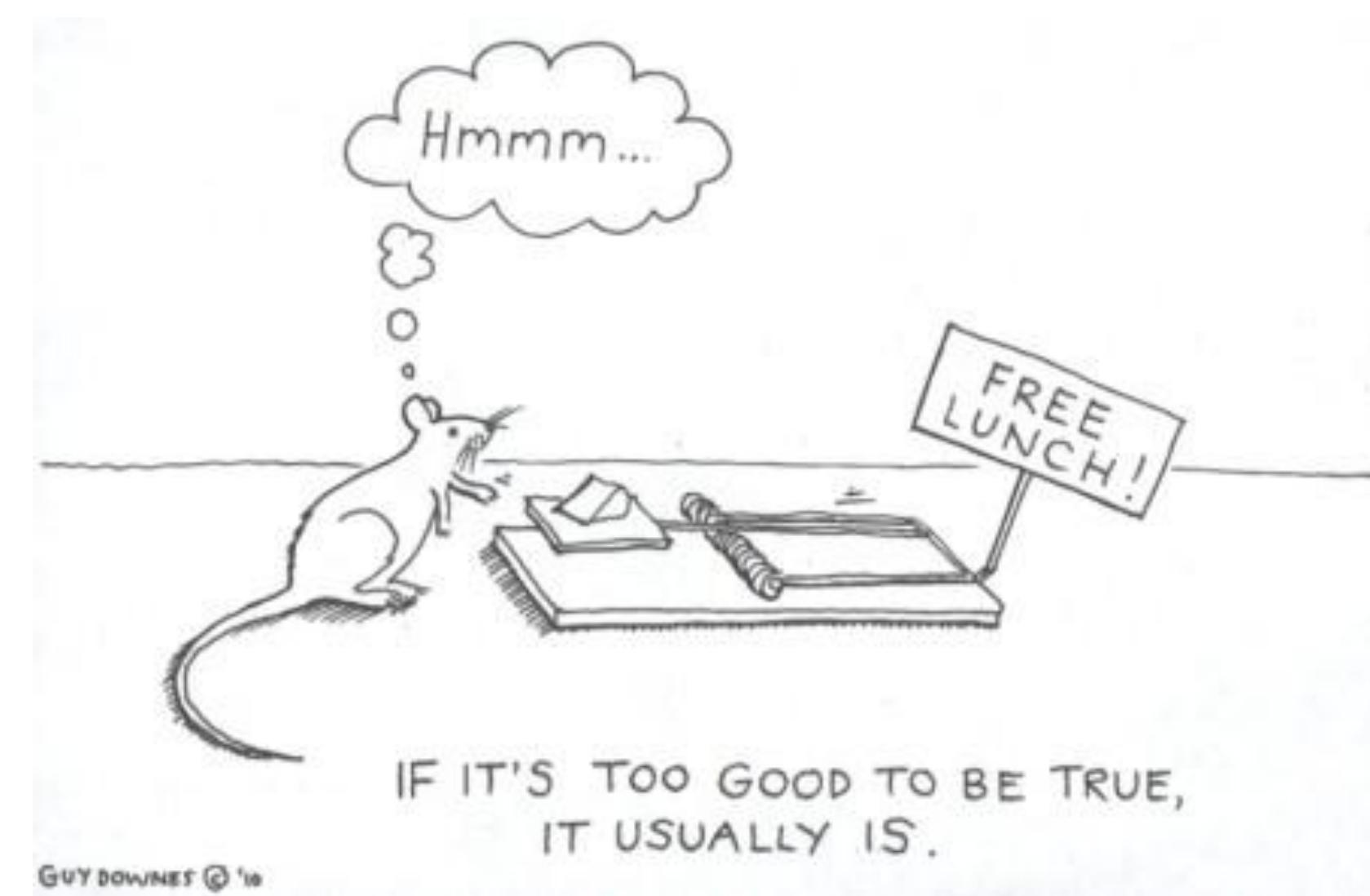


- ▶ **Processo software:** insieme di attività che porta alla creazione o all'evoluzione di un prodotto software
 - *Acquisizione, analisi e specifica dei requisiti:* clienti e ingegneri definiscono le funzionalità e i vincoli operativi del software da produrre
 - *Progettazione e Sviluppo:* progettazione e programmazione
 - *Verifica e Validazione:* si verifica che il software sia esattamente ciò che il cliente richiede
 - *Evoluzione:* il software viene modificato per soddisfare eventuali cambiamenti dei requisiti del cliente e del mercato

NO FREE LUNCH



- ▶ Non esiste una soluzione che risolve tutti i problemi!
- ▶ Differenti tipi di software richiedono differenti processi di sviluppo, che però condividono le quattro attività fondamentali
- ▶ Non esistono tecniche e metodi di ingegneria del software che possono essere universalmente utilizzati per tutti i tipi di software





- ▶ **Metodi:**
 - Approcci strutturati per sviluppare software di qualità, a costi contenuti ed entro i tempi di consegna
 - Tali metodi forniscono una guida alle attività dei processi e alla relativa organizzazione
- ▶ **Strumenti:**
 - Sistemi software usati per aiutare le attività dei processi software (ad es., analisi, modellazione, debugging, testing)

ALCUNE SFIDE PER I PROCESSI SOFTWARE



- **Diversità:** Definire metodi per produrre software eseguito su dispositivi eterogenei (computer o cellulari con diversi sistemi operativi)
- **Consegna:** Consentire la consegna del software in tempi rapidi, rispondendo ai cambiamenti veloci del mondo circostante (ad es. nuove tecnologie)
- **Fiducia:** Sviluppare tecniche che dimostrino all'utente che può fidarsi del software, garantendo che le informazioni siano sempre protette (ad es. da utenti malintenzionati)
- **Scala:** Il software deve essere distribuito su molti sistemi diversi

PRINCIPI FONDAMENTALI DELL'INGEGNERIA DEL SOFTWARE

Vincenzo Riccio
Ingegneria del Software 2025/2026
Università degli Studi di Udine

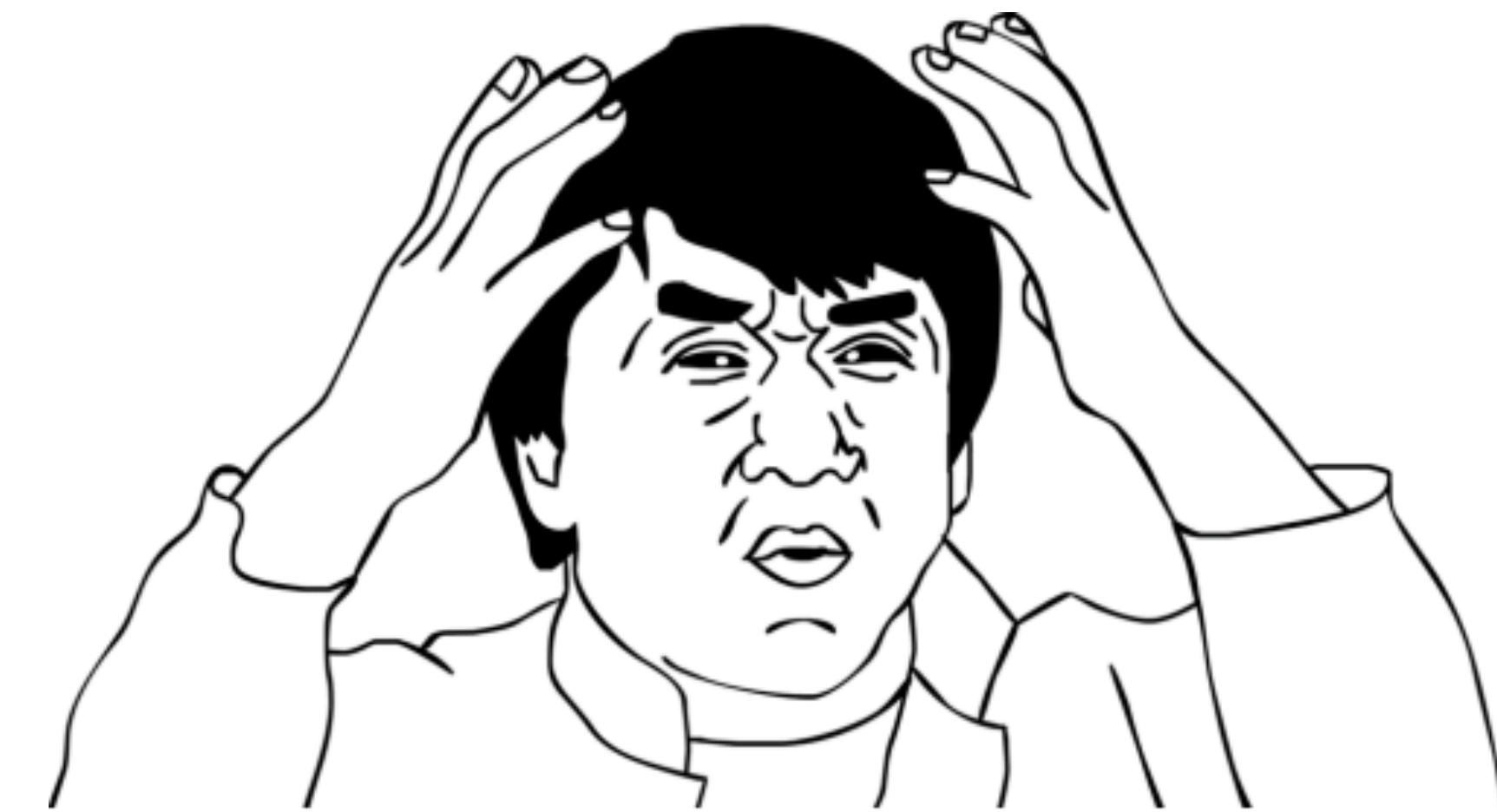


- I metodi specifici, le tecniche e gli strumenti utilizzati dipendono dall'organizzazione che sviluppa il software, dal tipo di software e dalle persone coinvolte nel processo di sviluppo
- Non ci sono metodi di ingegneria del software universali che sono applicabili a tutti i sistemi e a tutte le aziende
- I concetti fondamentali dell'ingegneria del software sono **indipendenti dal linguaggio di programmazione** utilizzato per sviluppare il software

PROCESSO CHIARO



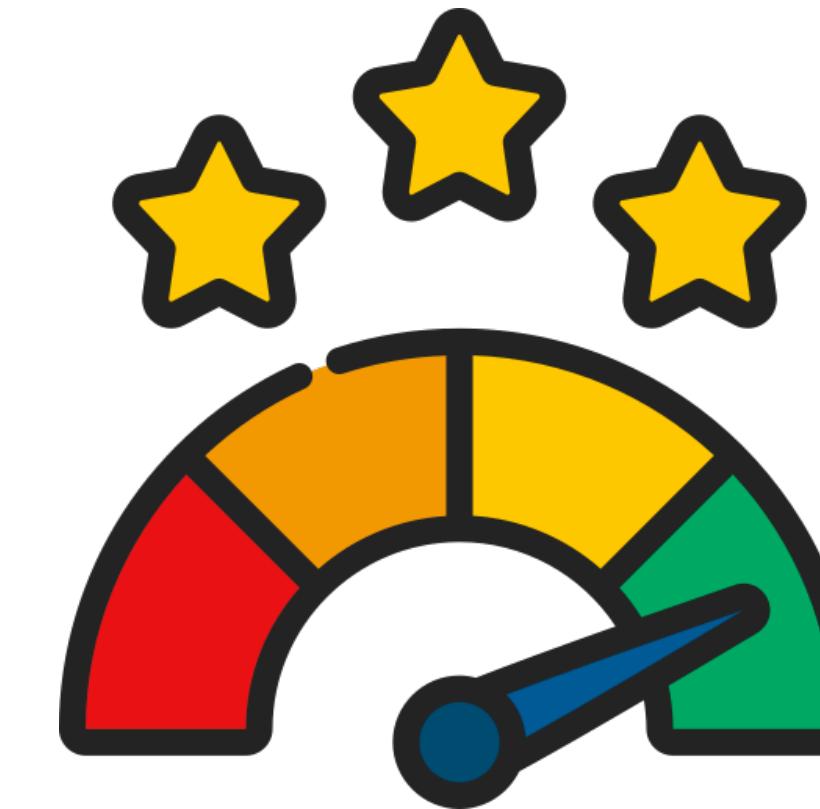
- ▶ Il processo specifico da utilizzare dipende dal tipo di software che sarà sviluppato
- ▶ Indipendentemente dal processo specifico, esso deve essere chiaramente definito e condiviso
- ▶ Le persone coinvolte nel processo devono avere idee chiare sulle proprie responsabilità, cosa sarà prodotto e le fasi del processo che le coinvolgono



FIDATEZZA E PRESTAZIONI



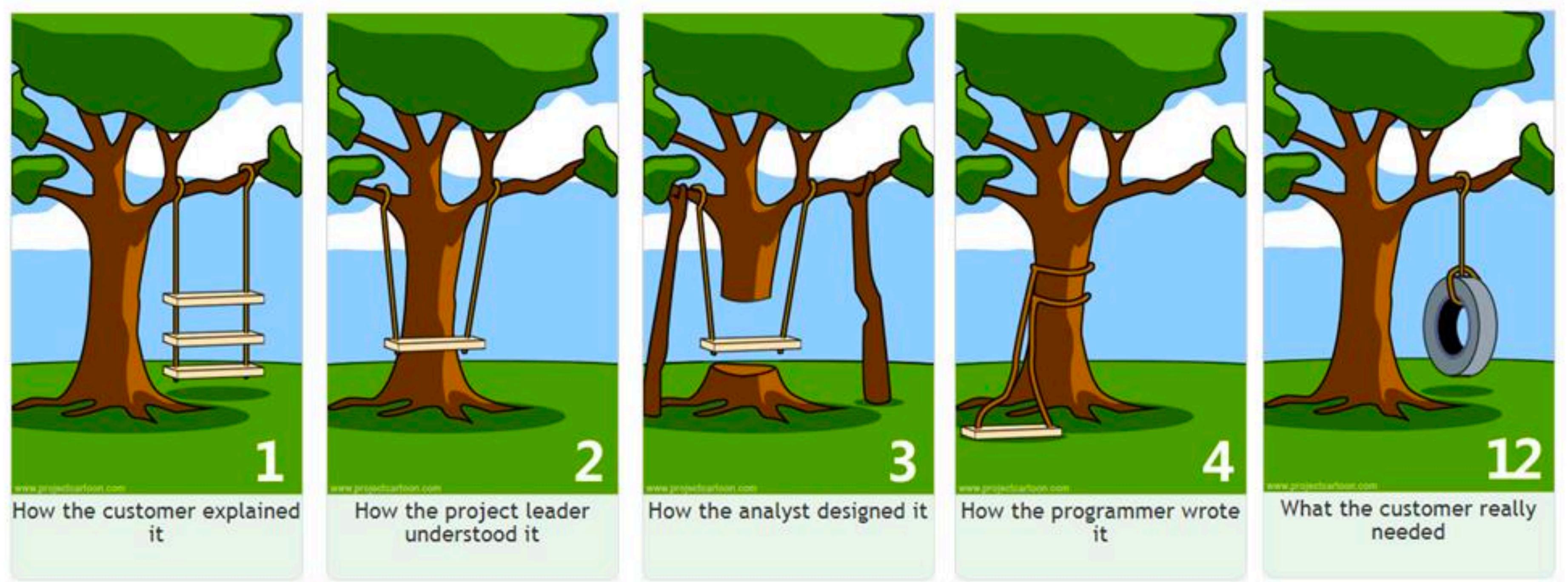
- ▶ La fidatezza e le prestazioni sono importanti per tutti i tipi di sistemi
- ▶ Il software dovrà comportarsi come previsto, senza fallimenti, e dovrà essere disponibile all'uso quando richiesto
- ▶ Il software dovrà essere protetto il più possibile contro attacchi esterni
- ▶ Il sistema dovrà essere eseguito senza spreco di risorse



REQUISITI



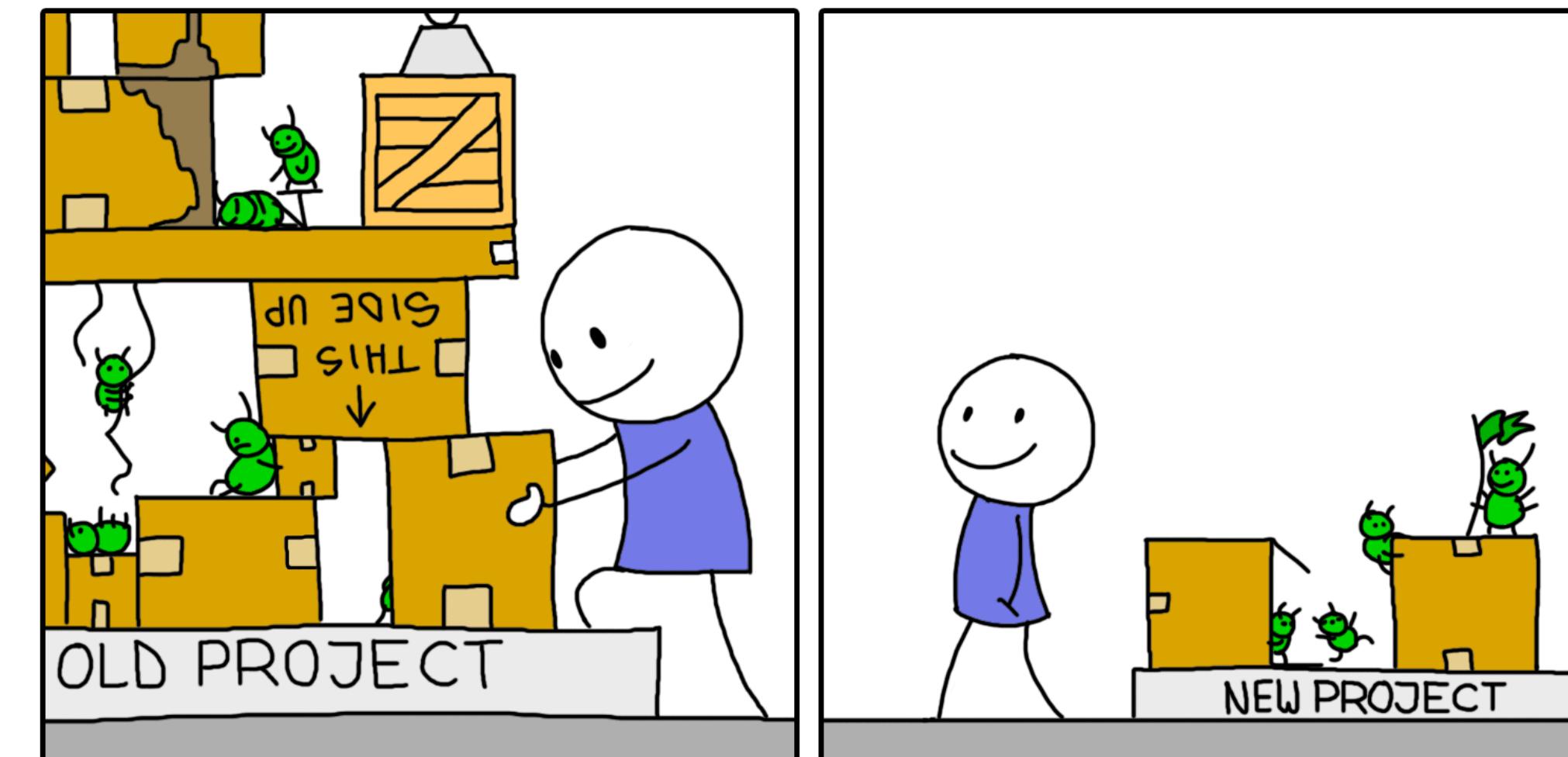
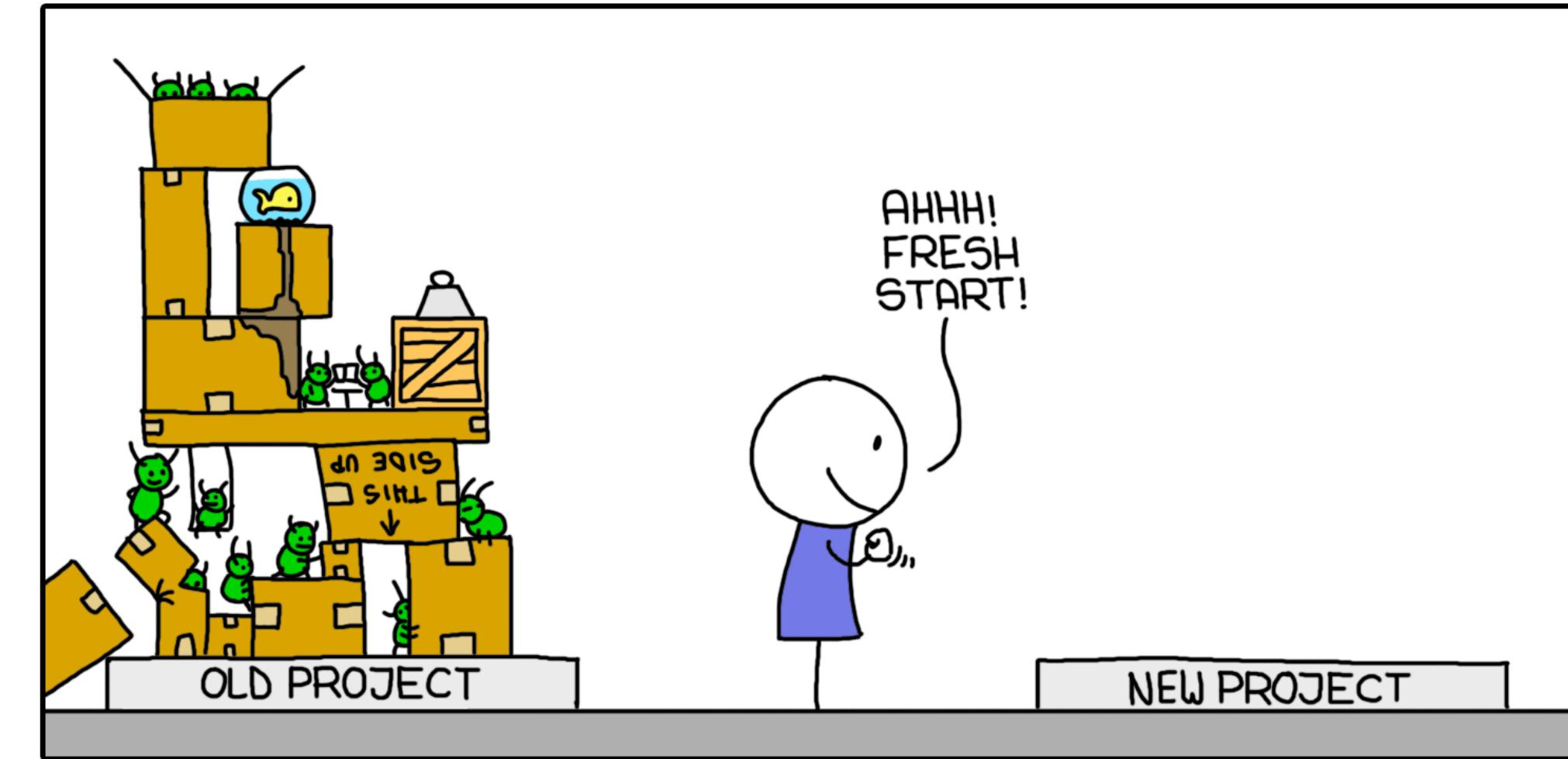
- ▶ È importante capire e gestire la specifica e i requisiti del software (cosa deve fare il software)
- ▶ Occorre sapere che cosa si aspettano clienti e utenti del sistema, e occorre gestire le loro aspettative in modo che possa essere rilasciato un sistema utile entro i costi e i tempi previsti





- ▶ Occorre utilizzare con efficienza le risorse esistenti
- ▶ Quando appropriato, occorre riusare il software che è già stato sviluppato, anziché scrivere un nuovo software

CODE REUSE

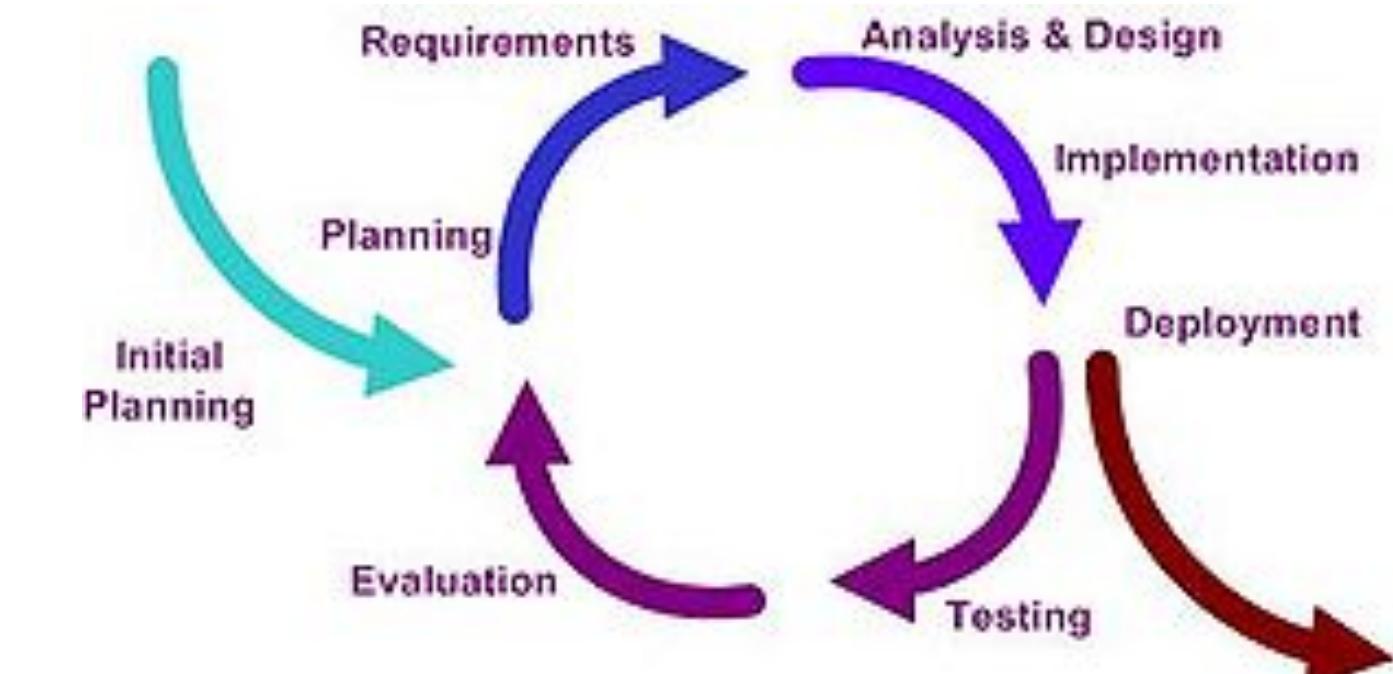
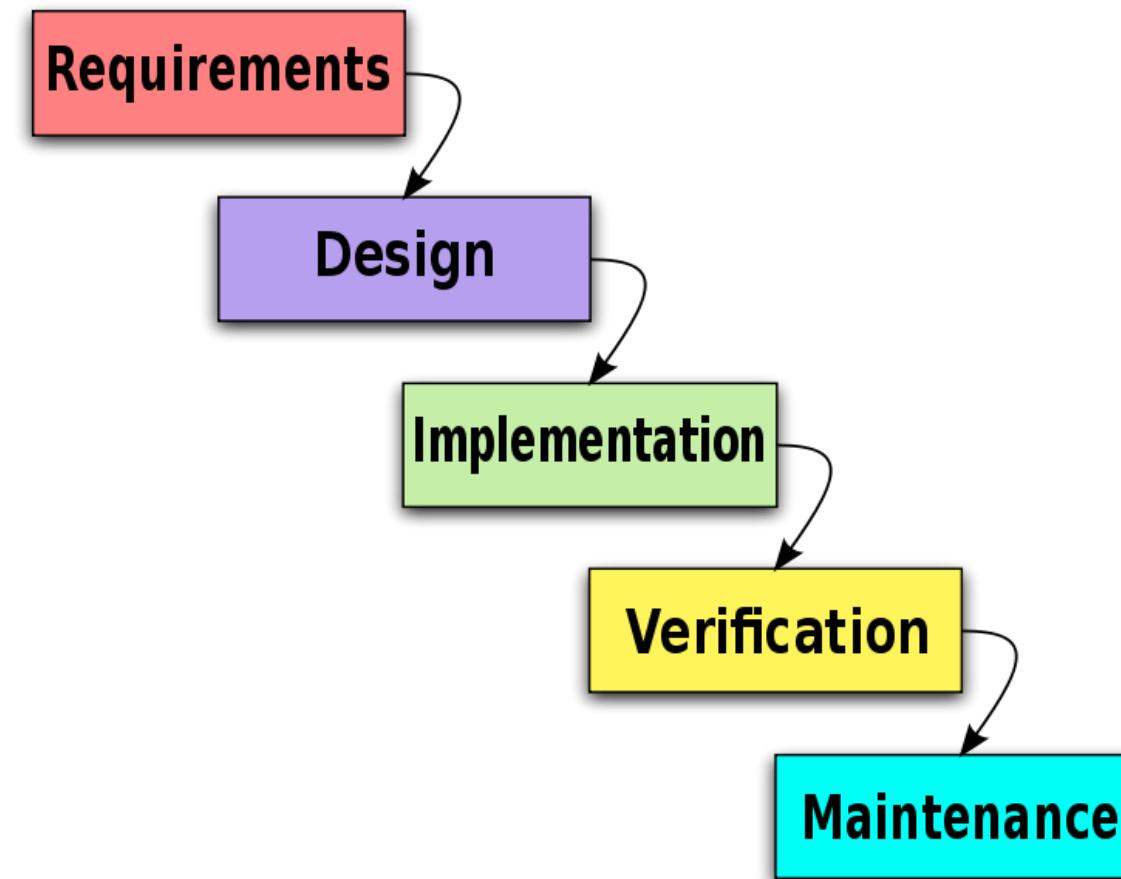


MONKEYUSER.COM

MODELLO DI PROCESSO SOFTWARE



- ▶ Ciascuna organizzazione utilizza il proprio processo
- ▶ Il **modello di processo software** è una rappresentazione semplificata e astratta che descrive l'intero ciclo di vita del software



- ▶ Questo corso sarà *lifecycle-based...*