

Università di Tor Vergata - Roma
Facoltà di Scienze - Informatica
Corso di Metodologia di Specifica del Software (MSS)

Problemi e pressioni del software

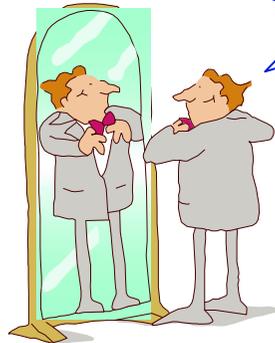
Software Engineering

Docente: Dr. Anna Rita Laurenzi, PhD
Docente aggiunto: Dr. Ercole Colonese

Anno accademico 2005-2006

Un approccio consolidato allo sviluppo del software

Ragazzi, iniziate pure a programmare che io vado dal cliente a sentire cosa vuole!



**Ovvero, la rivincita di
eXtreme Programming
(XP)**

Agenda

- *Introduzione all'Ingegneria del software*
- *Valore, problemi e sfide del software*
- *Un approccio ingegneristico*



Introduzione all'Ingegneria del software



Ingegneria del software (Software Engineering)

- Termine coniato nel corso della conferenza NATO organizzata a Garmisch (Germania) nell'Ottobre del 1968 per discutere del problema del software (in seguito riferito sempre come “*crisi del software*”)
- **Necessità di definire l'evoluzione del software per superare “la crisi”**
- Esigenza di far diventare lo sviluppo del software una “disciplina ingegneristica” con basi teoriche e metodologiche
- **Obiettivo di progettare, costruire e mantenere sistemi software di grandi dimensioni**
- Settore dell'ingegneria in costante evoluzione



Definizioni

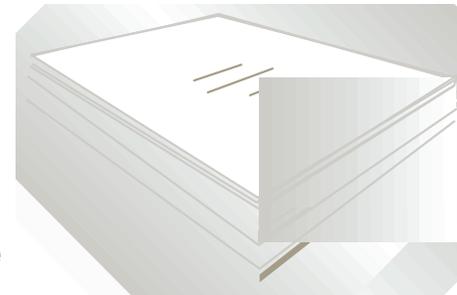
L'ingegneria del software è:

- 1. L'applicazione di un approccio sistematico, disciplinato e quantificabile allo sviluppo, all'operatività e alla manutenzione del software.***
- 2. Lo studio degli approcci di cui al punto 1.***

Il software è definito come:

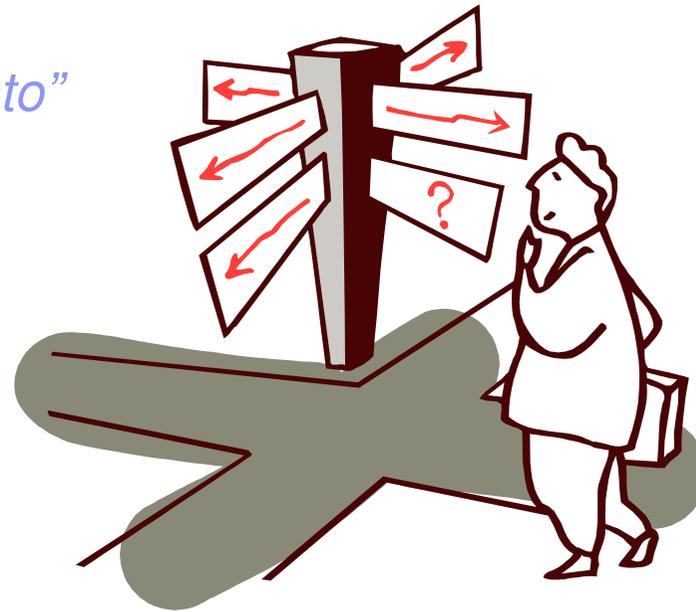
i programmi, le procedure, e l'eventuale documentazione associata, e i dati relativi all'operatività di un sistema di elaborazione.

IEEE Standard Glossary Engineering Terminology, 1990



Valore, problemi e sfide del software

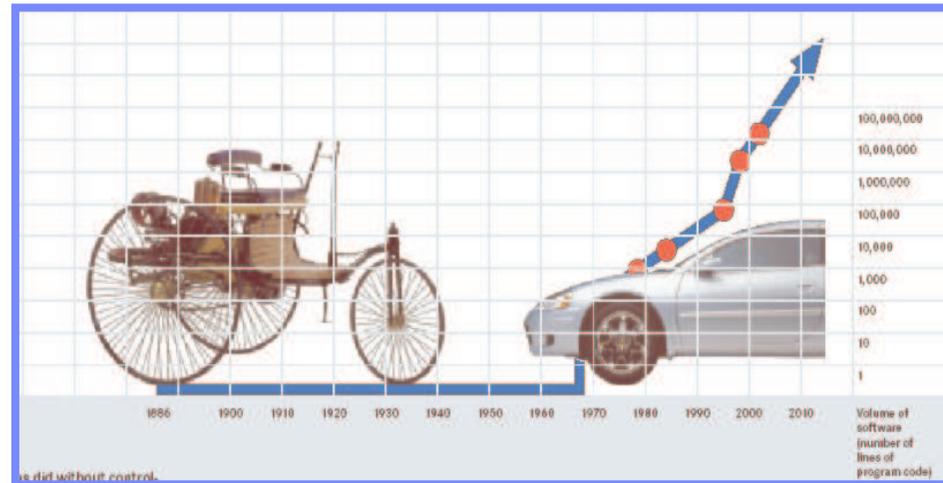
*“Fare di più e meglio a minor costo”
(faster, cheaper, and better)*



Il software come supporto all'innovazione

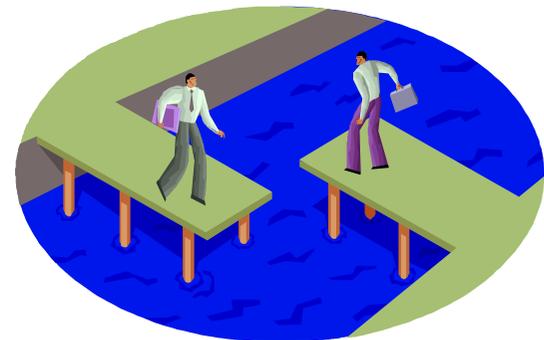
- Il software ha accompagnato lo sviluppo tecnologico degli ultimi 50 anni, spesso determinandolo in maniera significativa
- E' sempre più presente nella vita quotidiana e lo sarà ancora di più
- Supporta e controlla la maggior parte delle attività svolte nel mondo (diventando critico per molte funzioni)
- La competitività delle aziende sul mercato (oggi globale) richiede la disponibilità di software sempre più evoluto (e anche più complesso), in tempi più brevi ed a costi più bassi
- La sfida è raccolta dalle organizzazioni software che si vedono costrette ad operare in un mercato più competitivo e con margini di guadagno inferiori al passato
- L'ingegneria del software si evolve costantemente proponendo nuove tecnologie e consolidando best practice
- Ma la crisi del software rimane inalterata e ancora troppi progetti software soffrono degli stessi problemi:

difficoltà a rispettare Tempi, Costi e Qualità!



Sintomi e cause profonde dei problemi dello sviluppo del software

- Interpretazione errata delle esigenze dell'utente finale
- Difficoltà nel tenere conto dei cambiamenti nei requisiti
- Difficoltà nel riunire moduli diversi (integrazione)
- Software difficile da mantenere e da far evolvere
- Scoperta tardiva di gravi errori di progettazione
- Scarsa qualità del software
- Prestazione inaccettabile del software
- Cattiva organizzazione del lavoro che rende impossibile capire a posteriore chi ha modificato che cosa, quando, dove e perchè
- Processo di *build* e *release* inaffidabile.



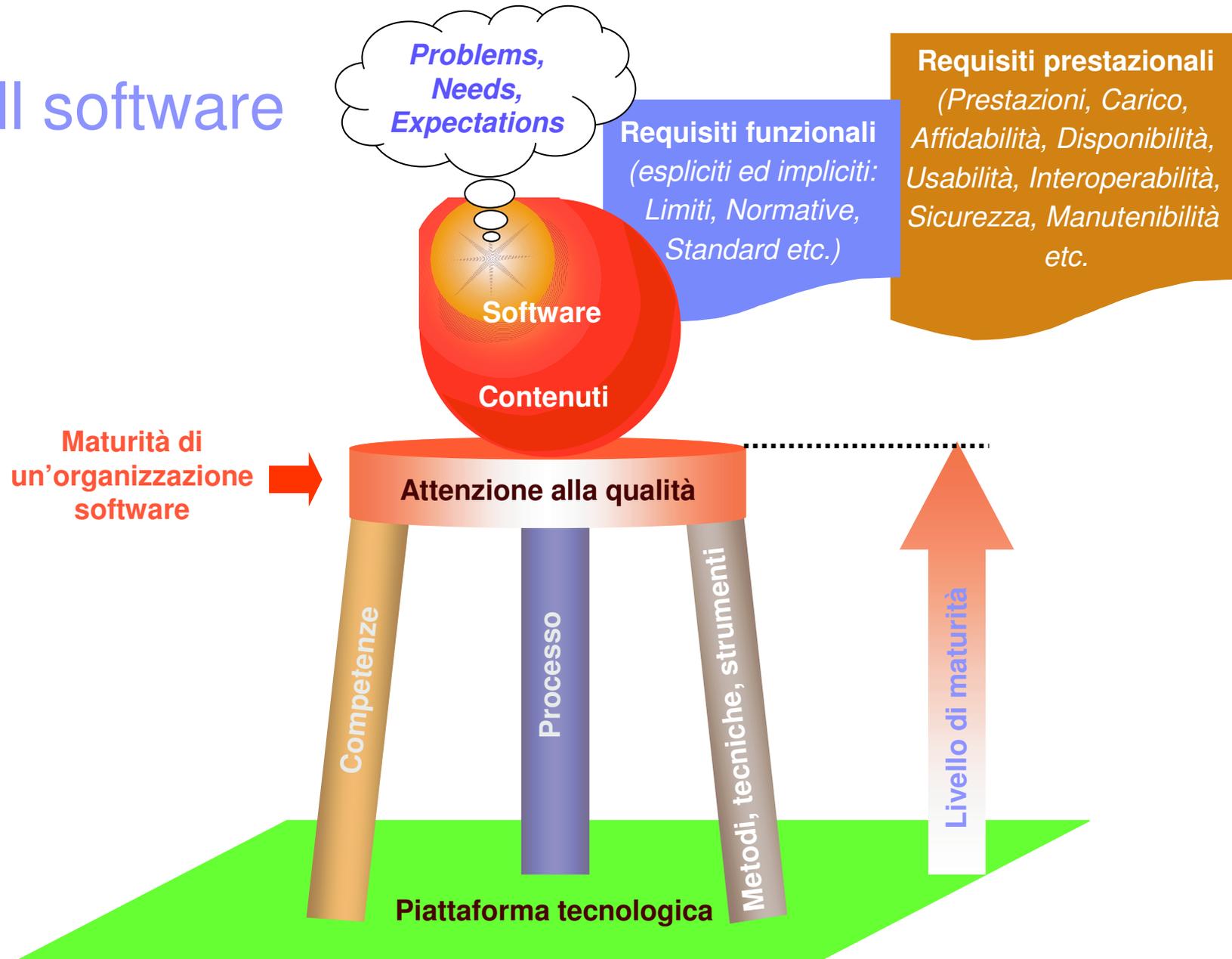
Origine: Caper Jones, "Patterns of Software Systems Failure and Success", 1996
Edward Yourdon, Death March: Managing "Mission Impossible" Projects, 1997

Combinazione delle cause più profonde

- Gestione non disciplinata dei requisiti
- Comunicazione ambigua e imprecisa
- Architettura instabile
- Complessità ingestibile
- Incoerenza latente tra requisiti, progettazione e implementazione
- Test insufficienti
- Impostazione soggettiva nella valutazione dello stato del progetto
- Incapacità di affrontare i rischi
- Assenza di controllo sulla propagazione delle modifiche
- Scarsa automazione delle attività

Origine: Grady Booch, “The Rational Unified Process - An Introduction”, 1999

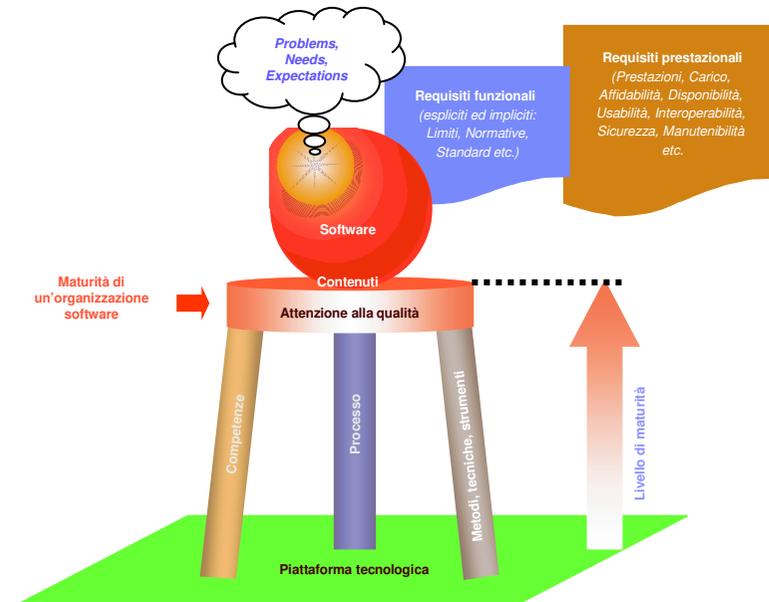
Il software



Il software

- Sviluppare software è un'attività creativa ad alto contenuto umano e quindi soggetta ad errori
- L'ingegneria del software è una disciplina giovane, “poco matura” e quindi soggetta periodicamente a “crisi”
- Il software risulta essere un oggetto con stabilità “precaria”
- I requisiti che implementa risultano spesso poco chiari, mutevoli e difficili da gestire
- La complessità del software è aumentata con l'evoluzione delle tecnologie
- Il personale impiegato non sempre ha la preparazione per affrontare le sfide crescenti
- E' ancora alto il numero di progetti software che non raggiungono gli obiettivi (rispetto dei tempi di consegna, del budget e della qualità attesa)
- Le tecnologie si evolvono di continuo ma i problemi del software rimangono

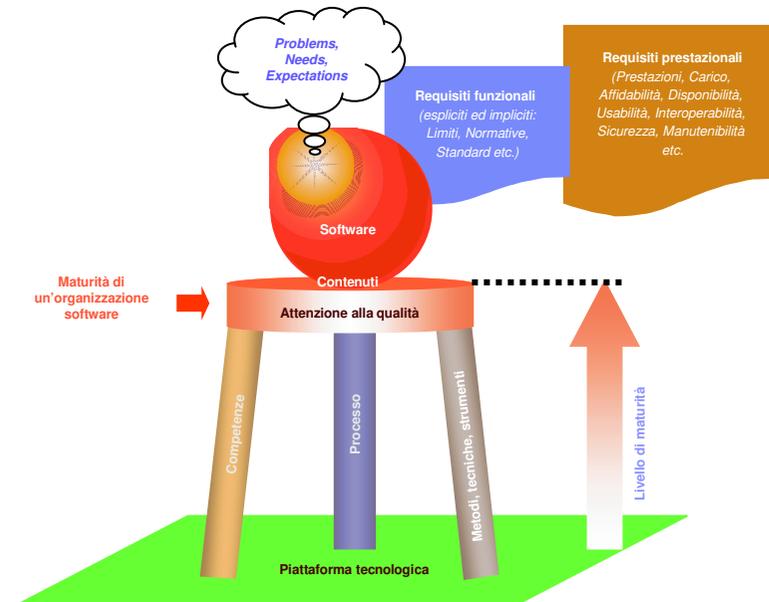
Il problema non è quindi “tecnologico” ma di tipo “metodologico”



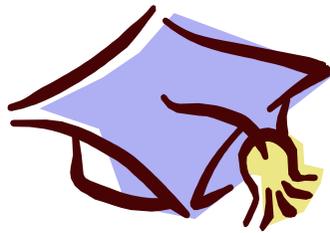
L'approccio al software

- Sviluppare software richiede quindi un approccio metodologico
- Un approccio metodologico richiede a sua volta un livello di maturità organizzativa adeguato al volume di affari gestito ed alla complessità del software sviluppato
- Il livello di maturità di un'organizzazione coincide con la sua “*attenzione alla qualità*”
- Il livello di maturità di un'organizzazione software poggia su tre pilastri fondamentali:
 1. la **Competenza** del personale addetto al software, a tutti i livelli, dirigenziale, manageriale, professionale e tecnico
 2. il **Processo** maturo, cioè definito, utilizzato, valutato e migliorato di conseguenza
 3. **Metodi, tecniche e strumenti** adeguati alle esigenze ed utilizzate in maniera costante e produttiva: “best practice” consolidate, “tecniche” efficaci, “strumenti” produttivi

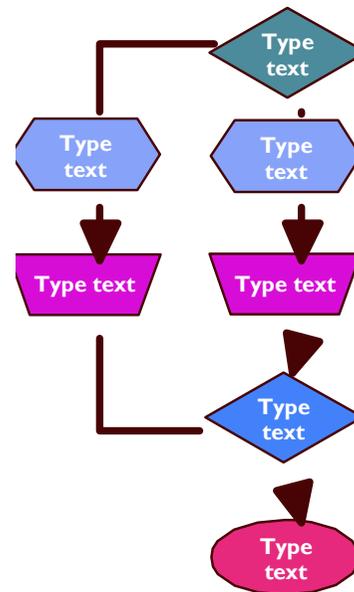
Questi tre elementi devono essere “proporzionati” e “sinergici”



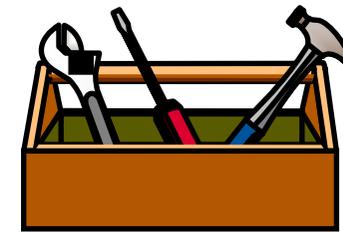
Un approccio ingegneristico dello sviluppo del software



“Competenze”



“Processo”



*“Metodi-tecniche-strumenti”
(Best Practices)*

L'approccio metodologico

- **Competenza delle persone:**

Ruoli e responsabilità ben definite

Competenze necessarie definite ed acquisite (Body of Knowledge)

Motivazione, assunzione di responsabilità, approccio proattivo, partecipazione attiva, lavoro di gruppo, ecc.

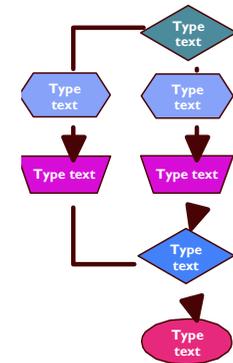


- **Processi maturi:**

Processi ben definiti, documentati, divulgati e ... UTILIZZATI

Misurazione e valutazione dell'efficacia dei processi

Miglioramento dei processi in base ai risultati delle misure



- **Utilizzo di metodi, tecniche e strumenti:**

Selezione di metodi e tecniche più consolidate (best practice)

Selezione degli strumenti più adatti all'organizzazione

Investimento in acquisizione e formazione

Verifica e controllo del livello di utilizzo, utilità ed efficacia

Analisi del mercato, valutazione di nuovi metodi, tecniche e strumenti

