

# Ingegneria del Software

## Introduzione

---

Software: Prodotto e Processo

Ingegneria del Software:  
Terminologia

1

1

## Produzione Software: Evoluzione

---

- ◆ **Arte:** applicazioni sviluppate da singole persone e utilizzate dagli stessi sviluppatori
- ◆ **Artigianato:** applicazioni sviluppate da piccoli gruppi specializzati per un cliente
- ◆ **Industria:** diffusione del software in diversi settori; crescita di dimensioni, complessità e criticità delle applicazioni; mercato e concorrenza; necessità di migliorare la produttività e la qualità; gestione dei progetti; evoluzione del software

2

2

## Programmi vs Prodotti (Sommerville)

---

- ◆ **Programma: l'autore è anche l'utente** (e.g., non è documentato, quasi mai è testato, non c'è progetto)
  - non ci serve un approccio formale
- ◆ **Prodotto software: usato da persone diverse da chi lo ha sviluppato**
  - è *software industriale* il cui costo è circa 10 volte il costo del corrispondente programma
  - dobbiamo avere un approccio formale allo sviluppo

3

3

## Prodotti Software

---

- ◆ **Prodotti generici**
  - sistemi stand-alone prodotti da una organizzazione e venduti a un mercato di massa
- ◆ **Prodotti specifici**
  - sistemi commissionati da uno specifico utente e sviluppati specificatamente per questo da un qualche contraente

*La fetta maggiore della spesa è nei prodotti generici ma il maggior sforzo di sviluppo è nei prodotti specifici*

4

4

## Software: prodotto (1)

---

- ◆ Devoto, Oli - Dizionario della lingua italiana 1971
  - *software*: Il corredo dei linguaggi e dei programmi di cui è munito un sistema elettronico per l'elaborazione dei dati
- ◆ Oxford Advanced Learner's Dictionary 1992
  - *software*: data, programmes, etc. not forming part of a computer but used when operating it

5

5

## Software: prodotto (2)

---

- ◆ Standard IEEE (Institute of Electrical and Electronics Engineers) 610.12-1990
    - *software*: the set of computer programs, procedures and possibly associate documentation and data pertaining to the operation of a computer system;
    - *software product*:
      1. The complete set of computer programs, procedures and possibly associate documentation and data designed for delivery to a user.
      2. Any of the individual items in 1.
- ... non solo il codice !
- ... ma tutti gli 'artefatti' che lo accompagnano e che sono prodotti durante l'intero sviluppo: codice, documentazione, casi di test, specifiche di progetto, procedure di gestione, manuali utente ...

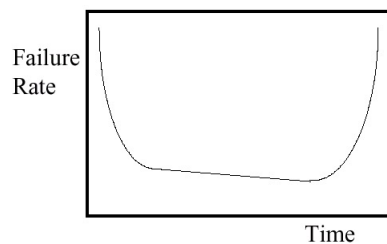
6

6

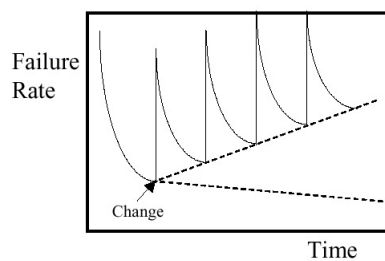
## Natura del Software

### ◆ Differenze con prodotti industriali classici

- Intangibile
- Malleabile
- Ad alta intensità di lavoro umano
- Spesso costruito ad hoc invece che assemblato
- Manutenzione = cambiamento



Failure Curve for Hardware



Failure Curve for Software

7

## Alcuni miti del Software

### ◆ Miti del management

- Abbiamo i più moderni computer ...
- Se siamo in ritardo possiamo recuperare aumentando il numero di programmatori

### ◆ Miti del cliente

- Un'affermazione generica degli scopi è sufficiente per cominciare a scrivere programmi
- I mutamenti nei requisiti di un progetto si gestiscono facilmente grazie alla flessibilità del software

### ◆ Miti del programmatore

- Una volta messo in opera il programma, il lavoro è finito
- Il solo prodotto di un progetto concluso è il programma

8

8

## Problemi della produzione software: Costi

---

- ◆ Il software ha costi elevati
  - Sono i costi delle risorse usate: ore lavoro (manpower), hardware, software e risorse di supporto.
  - Il manpower è dominante !
    - » Il costo è espresso in mesi/uomo
- ◆ Il testing impiega fino al 50% dei costi di sviluppo
- ◆ La manutenzione costa più dello sviluppo
  - Per sistemi che rimangono a lungo in esercizio i costi di manutenzione possono essere svariate volte il costo di produzione

9

9

## Problemi della produzione software: Ritardi

---

- ◆ Molti progetti sono “runaway” (in ritardo o fuori dal budget e lo schedule stimato)
- ◆ Esempio: US Air Force sistema di comando e controllo
  - stima iniziale della azienda vincitrice per la fornitura: 400.000\$,
  - costo successivamente rinegoziato a 700.000\$ e poi a 2.500.000\$ ...
  - costo finale 3.200.000\$, maggiore di circa 10 volte la stima iniziale !
  - ... e con notevole ritardo rispetto alla stima iniziale

10

10

## Problemi della produzione software: Abbandoni

---

- ◆ Molti progetti vengono abbandonati ...
- ◆ Una azienda nel settore della grande distribuzione di prodotti aveva richiesto un sistema che, stima iniziale, sarebbe stato sviluppato in 9 mesi al prezzo di 250.000 \$
  - Due anni dopo, e dopo una spesa di 2.500.000 \$, il lavoro non era stato ancora completato e fu stimato che erano necessari altri 3.600.000 \$ (!!)
  - Il progetto fu abbandonato !

11

11

## Problemi della produzione software: Affidabilità

---

- ◆ Il software è (spesso) inaffidabile
  - Molti malfunzionamenti sono rilevati durante l'operatività del sistema
  - Da un' analisi del ministero della difesa USA risulta che più del 70 % di tutti i malfunzionamenti sono dovuti al software (in sistemi con complicati e sofisticati apparati meccanici, elettrici, idraulici ....)
- ◆ .... l' Arienne 5 ...
- ◆ .... la sonda MARINER .....

12

12

## Necessità di un approccio ingegneristico

---

### ◆ Necessità di applicare principi ingegneristici alla produzione software per sviluppare:

- il *giusto* prodotto
- al *giusto* costo
- nel tempo *giusto*
- con la *giusta* qualità

13

13

## Ingegneria del Software: Scopo

---

### ◆ Riguarda la costruzione di software:

- di grandi dimensioni
- di notevole complessità
- sviluppati tramite lavoro di gruppo

### ◆ Progetti software di questo tipo hanno tipicamente:

- versioni multiple
- lunga durata
- frequenti cambiamenti
  - » eliminazione di difetti
  - » adattamento a nuovi ambienti
  - » miglioramenti e nuove funzionalità

*Multi-person construction of multi-version software (Parnas)* <sup>14</sup>

14

## Ingegneria del Software: Contesto

---

- ◆ Software Engineering  $\subset$  System Engineering
  - La maggior parte del SW è collocata all' interno di un "sistema" misto HW/ SW
  - L' obiettivo finale di chi produce è creare tale sistema che soddisfa globalmente i requisiti dell' utente
  - Coinvolgimento nella definizione dei requisiti del sistema
- ◆ Conoscenza del dominio applicativo
  - È essenziale per un efficace sviluppo del SW
  - Il SW è utile quando riesce a condensare nei suoi algoritmi la conoscenza del dominio applicativo
  - Altrimenti inutile o dannoso
    - » Per esempio, sistema di controllo di un aeroplano

15

15

## Ingegneria del Software: Definizioni

---

- ◆ Applicazione di una strategia sistematica, disciplinata e misurabile allo sviluppo, esercizio e manutenzione del software (programmi, procedure, regole e associata documentazione, dati)
  - IEEE (Institute of Electrical and Electronics Engineers) Standard Glossary of Software Engineering Terminology, 1993
- ◆ La disciplina tecnologica e manageriale che riguarda la produzione sistematica e la manutenzione dei prodotti software che vengono sviluppati e modificati entro i tempi e i costi preventivati
  - D. Farley

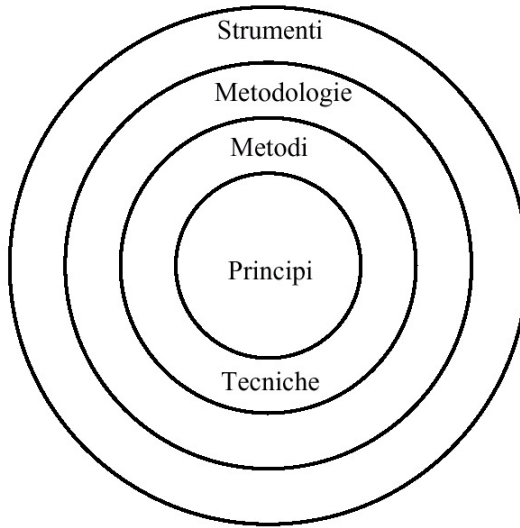
16

16



## Ingegneria del Software: Fondamenti

---



**L'ingegneria del software si occupa dei *metodi*, delle *metodologie*, dei *processi* e degli *strumenti* per la gestione professionale (sviluppo, manutenzione, ritiro) del software**

17

17

## Ingegneria del Software: Principi

---

- ◆ **Rigore:** concetto primitivo (precisione, accuratezza)
- ◆ **Formalità:** oltre il rigore (fondamento matematico)
- ◆ **Separazione di aspetti diversi:** affrontare separatamente i vari aspetti di un problema complesso
- ◆ **Modularità:** suddividere un sistema complesso in parti più semplici
- ◆ **Astrazione:** si identificano gli aspetti cruciali in un certo istante ignorando gli altri
- ◆ **Anticipazione del cambiamento:** la progettazione deve favorire l'evoluzione del SW
- ◆ **Generalità:** tentare di risolvere il problema nella sua accezione più generale
- ◆ **Incrementalità:** lavorare per passi successivi

18

18

## Metodi e Metodologie

---

- ◆ **Metodo (o tecnica):** procedimento generale per risolvere classi di problemi specificati di volta in volta
  - linee guida o regole che governano le attività
  - il metodo dei minimi quadrati, il metodo di Montecarlo, il metodo di Newton, come fare il brodo di carne, come fare il lessso, ...
- ◆ **Metodologia:** insieme di principi, di metodi, degli elementi di cui una o più discipline si servono per garantire la correttezza e l'efficacia del proprio procedere
  - e.g. la metodologia della macerazione carbonica permette di ottenere vini novelli, freschi, profumati, ...

19

19

## Strumenti, Procedure, Paradigmi

---

- ◆ **Strumento (tool):** un artefatto, un sistema per fare qualcosa in modo **migliore**
  - e.g., il frullatore per fare la maionese, un cavatappi per aprire una bottiglia
  - supporti SW pratici all'applicazione
- ◆ **Procedura:** una combinazione di **strumenti e metodi** che assieme permettono di produrre un certo prodotto
  - e.g., la ricetta della Saker Torte: montare a neve le chiare di 4 uova ...
- ◆ **Paradigma:** un particolare approccio o filosofia per fare qualcosa
  - e.g., è lo stile della cucina, noi riconosciamo la cucina Francese, quella Italiana, quella Cinese ...

20

20

## Processo

---

Un processo è un particolare *metodo* per *fare qualcosa* costituito da una **sequenza di passi** che coinvolgono **attività, vincoli e risorse** (Pfleeger)

Processo: una particolare metodologia operativa che nella tecnica definisce le singole operazioni fondamentali per ottenere un prodotto industriale (Zingarelli)

***Processo software: un metodo per sviluppare del software***  
(Sommerville)

21

21

## Processo software

---

- ◆ Insieme organizzato di *attività* che sovrintendono alla costruzione del prodotto da parte del team di sviluppo utilizzando metodi, tecniche, metodologie e strumenti.
- ◆ È suddiviso in varie *fasi* secondo uno schema di riferimento (*il ciclo di vita del software*)
- ◆ Descritto da un *modello*: informale, semi-formale o formale (*maturità del processo*)

22

22

## Processo software: Standard IEEE 610.12-1990

---

- ◆ *Software development process*: The process by which user needs are translated into a software product. The process involves translating user needs into software requirements, transforming the software requirements into design, implementing the design in code, testing the code, and sometimes, installing and checking out the software for operational use.
- ◆ Note: These activities may overlap or be performed iteratively.
- ◆ See also: incremental development; rapid prototyping, spiral model, waterfall model.

23

23

## Classi di Progetti (B. Boehm)

---

Classe	linee di codice	Persone (media)	Mesi
Small	2000	1-2	4-5
Intermediate	8000	2-6	8-9
Medium	32000	6-16	14
Large	128000	16-51	24
Very Large	512000	60-157	41-42
...	...	...	...

- ◆ Necessità del project management
  - project planning (def. attività, risorse da allocare, tempi)
  - project monitoring e controllo
  - metriche di prodotto e di processo per misurare e controllare...

24

24

## CASE (Computer-Aided Software Engineering)

---

- ◆ Sistemi Software che intendono fornire un supporto automatico per le attività di un processo software
  - **Upper-CASE.** Strumenti che supportano le attività delle fasi di analisi e specifica dei requisiti e progettazione di un processo software. Includono editor grafici per sviluppare modelli di sistema, dizionari dei dati per gestire entità del progetto
  - **Lower-CASE.** Strumenti che supportano le attività delle fasi finali del processo, come programming, testing e debugging. Includono generatori di graphical UI per la costruzione di interfacce utente, debuggers per supportare la ricerca di program fault, traduttori automatici per generare nuove versioni di un programma

25