

1. Introduction

- Getting started with software engineering

Set 1 - Cosa ricordare: concetti, motivazioni, conseguenze, relazioni fra concetti, ecc.

- Importanza e motivazioni dell'Ingegneria del SW (IS), Mercato Italiano del SW
- L'aspetto economico: i costi del SW: manutenzione vs. sviluppo; costruzione vs. testing; distribuzione dei costi dei vari processi sw
- SW include documentazione
- SW generico vs. SW customizzato; approcci allo sviluppo, configuration, re-use
- Definizioni e limiti di IS; system engineering; prog. in the small/in the large
- Prime definizioni di alcuni processi fondamentali
- Ragionamento ANALITICO e R. SINTETICO
- Processo SW; MODELLO dei Processi SW; modelli generici; dipendenza della distribuzione dei costi dal modello usato
- Parametri di Qualità del SW: manut, 'potersi fidare', sicurezza&protezione, efficienza, accettabilità
- Le grandi sfide attuali dell'IS: sistemi Legacy, eterogeneità hd e SW, richiesta di velocità nello sviluppo e nel delivery

Software engineering

Motivazioni dell'importanza del Software:

- The economies of ALL developed nations are dependent on software
- More and more systems are software controlled
- Software engineering is concerned with theories, methods and tools for professional software development
- Software engineering expenditure represents a significant fraction of GNP in all developed countries

Mercato del SW in Italia

- Anitec-Assinform – l'associazione di Confindustria che raggruppa le principali aziende del settore riporta per il 2020:
- In IT. Mercato **totale digitale** (B2B e retail): 70 miliardi di € (più del **3% del PIL**)
- In **.IT** il mercato totale relativamente al **SW** è di circa **7.6** Miliardi di €, con la parte applicative in crescita (+10%).

Da: https://www.anitec-assinform.it/kdocs/1981725/il_digitale_in_italia_2020_vol_i.pdf

SW e IT rispetto a tutto il settore industriale italiano

« ...complessivamente le imprese del **settore IT** sono **più produttive** rispetto al resto delle imprese italiane: il settore IT produce il 3,7% del valore aggiunto del totale economia e registra una produttività mediana (**per addetto**) pari a circa **26mila** euro contro i circa **18mila** delle altre unità economiche.

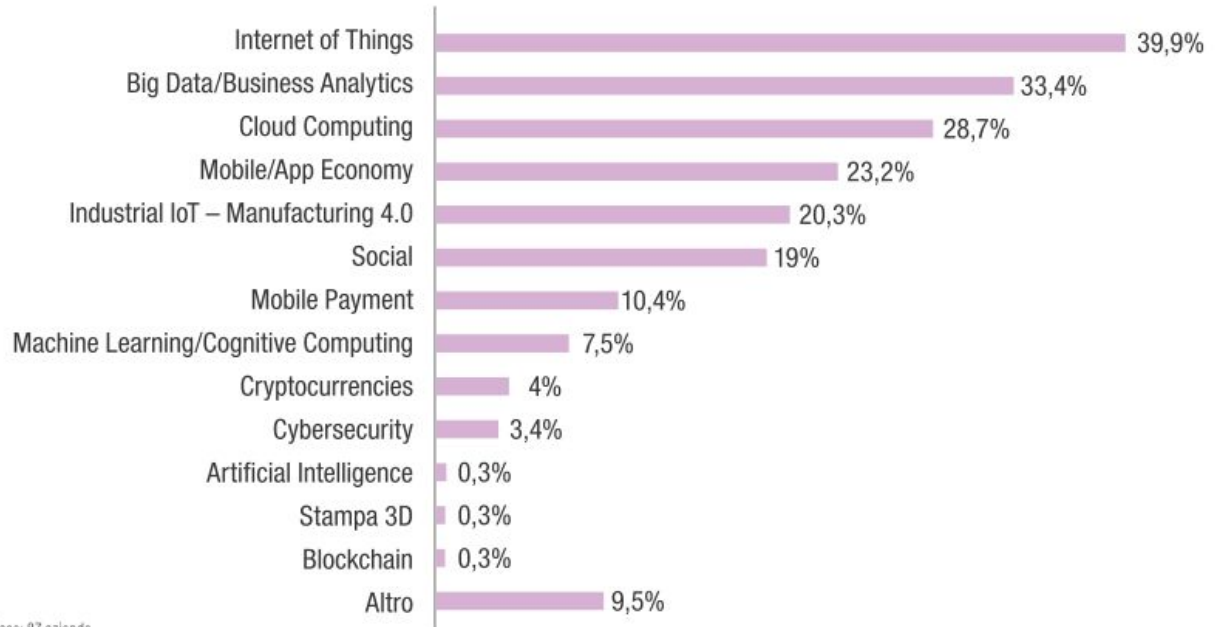
Le imprese dei comparti **software** e degli altri servizi IT sono caratterizzate da una produttività maggiore rispettivamente di circa il **40%** (**con 24.650 euro**) e l'88% (**con 32.637 euro**) rispetto a quella prodotta dalle imprese italiane del terziario (**17.299**). ...» [Rapporto Assinform 2017]

Ambiti dell'Innovazione nelle aziende IT

Figura 3.6
Principali ambiti
tecnologici
d'innovazione delle
aziende IT, 2015-2018

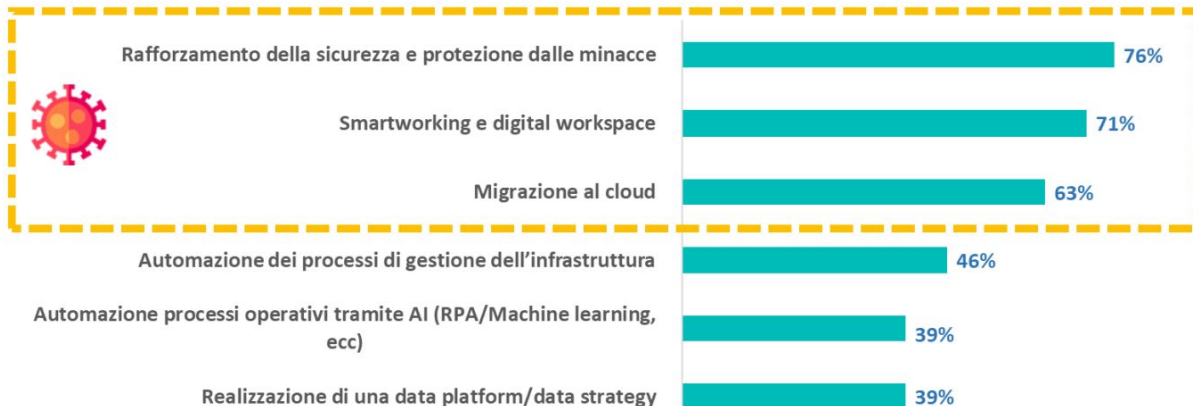
Valori %

Fonte: elaborazioni
NetConsulting cube per
Assinform, 2017



Base: 87 aziende

Il COVID ha cambiato le priorità IT delle aziende



Software costs

Gli aspetti economici legati al Software:

1. Software costs often **dominate** ^{HW} system costs. The costs of software on a PC are often greater than the hardware cost
2. Software costs more to **maintain** than it does to **develop**. For systems with a long life, maintenance costs may be **several times** development costs
↗ allungare il tempo di vita di un SW
3. Come Baseline: considerare un costo TOTALE di sviluppo e utilizzo, suddiviso in due quote del 50%! Ossia la manutenzione costa in totale almeno quanto lo sviluppo!!
4. Software engineering is concerned with **cost-effective** software development

What is software?

- Computer programs and associated documentation
- Software products may be developed for a particular customer or may be developed for a general market
- Software products may be
 - **Generic** –developed to be sold to a range of different customers
 - **Bespoke** (customized / custom) - developed for a **single** customer according to their needs/requirements/specifications → *SW generico: es. Word, Powerpoint... li abbiamo tutti uguali*
- New software can be created by **1. developing from scratch** new programs, **2. configuring/(personalizzare, adattare,)** generic software systems or **3. riusare/ristrutturare** existing software.

Software products

◇ Generic products

- Stand-alone systems that are marketed and sold to any customer who wishes to buy them. Anche denominati **COTS - Commercial off-the-shelf**
- Examples – PC software such as graphics programs, Office, project management tools; CAD software; software for specific markets such as appointments systems for dentists; generic Information Systems (ad es. per contabilità IVA, stipendi, ecc.), APP, ...

◇ Customized products

- Software that is commissioned by a specific customer to meet their own needs.
- Examples – embedded control systems, air traffic control software, traffic monitoring systems, sistemi per gestire processi specifici di un'azienda, quali ad es. Processi produttivi particolari, APP aziendali specifiche, ...

Product specification

- ◇ **Specifiche:** cosa il SW deve 'fare' → *che funzionalità deve avere?
input? output?*
- ◇ **Generic products** → *le specifiche sono decise dal venditore in quanto è un software di massa*
 - The **specification** of what the software should do is owned by the software **developer** and decisions on software change are made by the developer.
- ◇ **Customized products** → *le specifiche sono richieste dal cliente, dipendono da lui (cliente)*
 - The specification of what the software should do is owned by the **customer** for the software and they make decisions on software changes that are required.

What is software engineering?

- Software engineering is an **engineering discipline*** which is concerned with ^{RIGUARDA} **all** aspects of software production and later use
- Software engineers should adopt a **systematic, disciplined, organised** and **quantifiable** approach to their work and use appropriate tools and techniques depending on the problem to be solved, the development ^{VINCOLI} constraints and the resources available

* Using appropriate theories and methods to solve problems **bearing in mind organizational and economic/financial constraints**

What is the difference between software engineering and computer science?

- Computer science is concerned with theory and fundamentals; software engineering is concerned with the practicalities of developing and delivering useful software. In other terms: SE is concerned with solving real problems with software systems
- Computer science theories are currently insufficient to act as a complete underpinning for software engineering

What is the difference between software engineering and **system engineering**?

- **System engineering** is concerned with **all** aspects of computer-based systems development including **hardware**, software, **process engineering**, **people**, ...
- Software Engineering (SwE) **is part** of this process
- Le attività del System engineering sono le stesse che vengono svolte per il SwE, ma riguardano, oltre al SW, anche l'HW, TLC, ...
- System engineers are involved in system **specification**, **architectural design**, **implementation/construction**, **integration** and **deployment** →

✉ le ritroveremo studiando il SwE

Software engineering vs. programming

- ◇ More and more, **individuals and society** rely on advanced **software systems**. We need to be able to produce *reliable* and *trustworthy* systems economically and quickly.
- ◇ It is usually cheaper, in the long run, to use software engineering methods and techniques for **software systems** rather than just write the programs as if it was a **personal programming project**.
- ◇ **Programming in the large** vs. programming **in the small**
- ◇ For most types of system, the majority of costs are the costs of changing the software after it has gone into use.

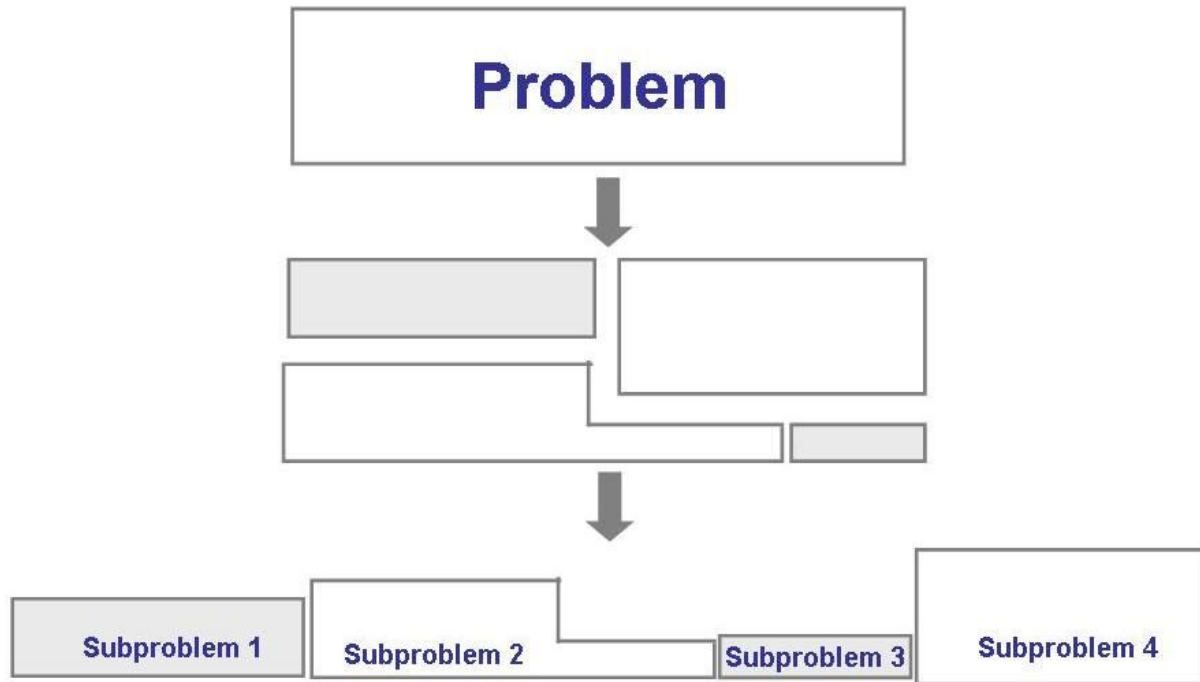
Tecniche generali di ragionamento nel SE

SE as problem solving

- Software products are large and complex
- Development requires two types of reasoning/problem solving approaches:
 - **Analysis**: decompose a large problem into smaller, understandable pieces → *dividi et impera*
 - abstraction is the key
 - **Synthesis**: build (compose) software from smaller building blocks → *ricomporre i sottoproblemi*
 - composition is challenging (✉ concetto di INTEGRAZIONE)

(from Pfleeger and Atlee, © 2006 Pearson/Prentice Hall)

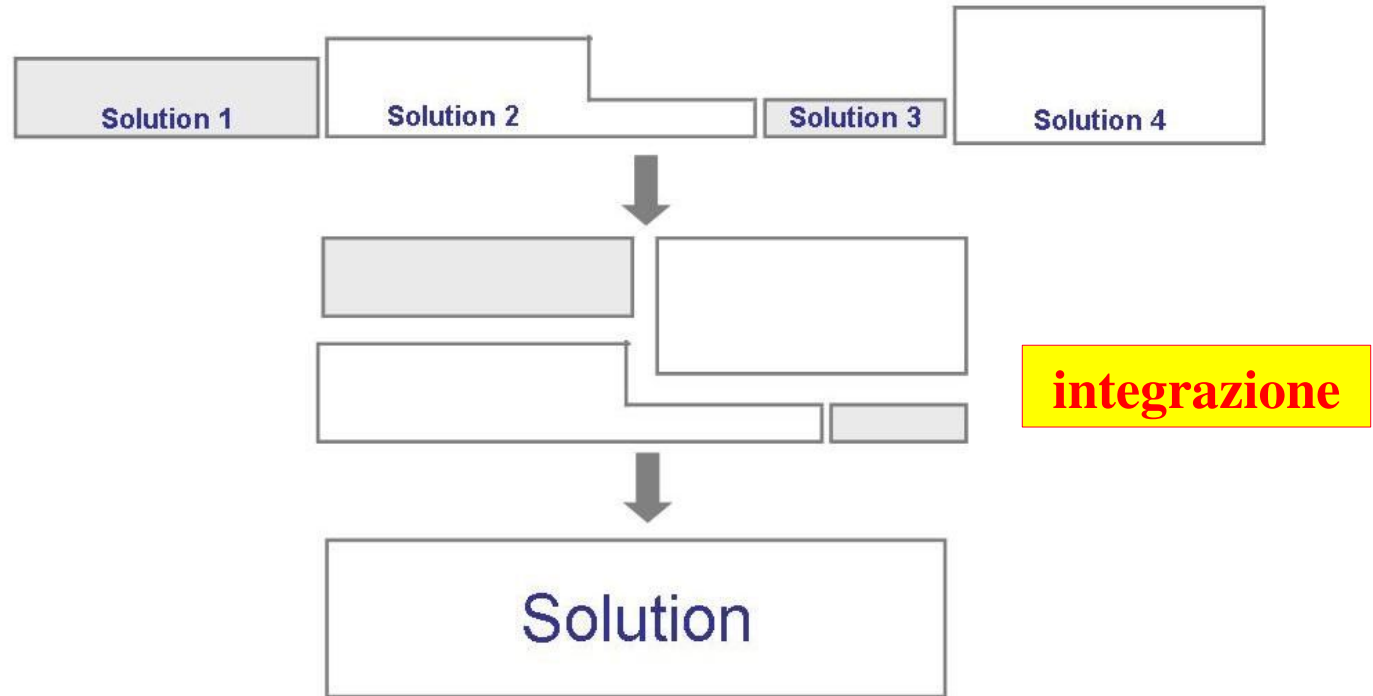
SE as problem solving: **analysis**



(from Pfleeger and Atlee, © 2006 Pearson/Prentice Hall)

Mettere ASSIEME i pezzi (sottoproblemi risolti) si dice INTEGRAZIONE

SE as problem solving: **synthesis**



(from Pfleeger and Atlee, © 2006 Pearson/Prentice Hall)

What is a **software process**?

Dirige cosa fare prima e cosa fare dopo.

- A **set of inter-related activities whose goal is the development or evolution of software**
- Generic activities in all software processes are:
 - **Requirement Analysis** – where customers' needs are acquired, discovered, and analysed by software engineers
 - **Specification** – where customers and software engineers define what software is to be produced and the constraints on its operation
 - **Development** – production: first **design** and then **programming** (progettazione e realizzazione) of the software system
 - **Validation** - checking that the software is what the customer requires
 - **Evolution** – modifying the software to reflect **changing** customer and market requirements

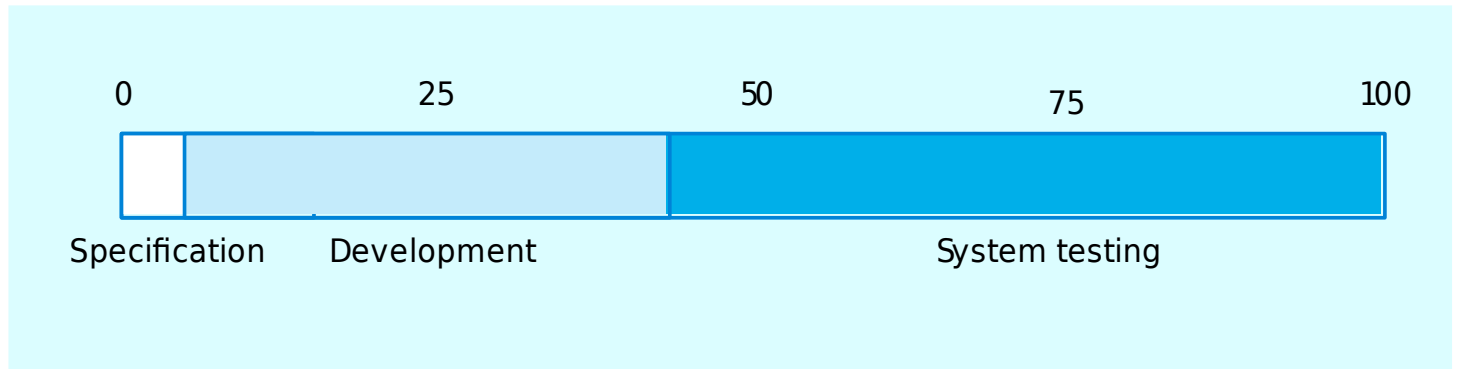
What is a software **process model**?

- A simplified representation of a software process, presented from a specific perspective
- Examples of process **perspectives** are
 - Workflow perspective - sequence of activities
 - Data-flow perspective – data/information flow
 - Role/action perspective - who does what
- **Generic** process models
 - Waterfall
 - Evolutionary development, Prototyping
 - Formal transformation
 - Integration from reusable components
 - Agile
 -

What are the **costs** of software engineering?

- Roughly **60%** of the system costs are concerned with **building** the software (costruzione sw vera e propria, i.e. analisi, progettazione e programmazione), the remaining **40%** are **testing** costs. For custom software, evolution (maintenance) costs often **exceed** development costs
- Costs vary depending on the type of system being developed and the requirements of system attributes such as **performance** and system **reliability**
- Distribution of costs depends on the development model that is used

Distribuzione dei costi di sviluppo



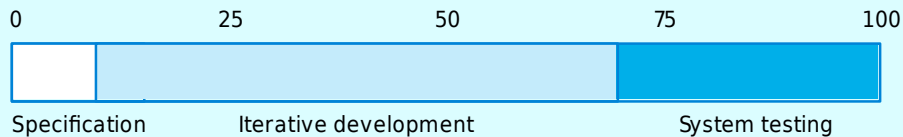
Activity cost distribution

In base al modello
che scelgo cambia il
tempo di ciascuna attività

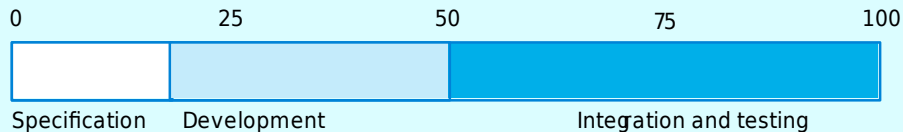
Waterfall model



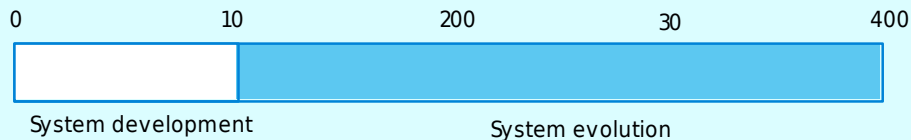
Iterative development



Component-based software engineering



Development and evolution costs for long-lifetime systems



What are the **attributes** of good software? (parametri di **qualità** del SW)

- The software should deliver the required functionality and performance to the user and should be maintainable, dependable and usable
- **Maintainability** (MANUTENIBILITÀ)
 - Software must evolve to meet **changing needs**
↳ poter dipendere
- **Dependability and security**
 - Software must be **trustworthy**, secure (**Protezione**: non danneggiare i propri dati) and safe (**Sicurezza**: non provocare danni a cose, persone, ...)
- **Efficiency**
 - Software should not make wasteful use of system resources
- **Acceptability**
 - Software must be acceptable to the type of users for which it is designed. This means that it must be **understandable**, **usable** and compatible with other systems that they use

What are the **key challenges** facing TODAY software engineering?

- Coping with **legacy** systems, coping with increasing **diversity** and coping with demands for **reduced delivery times**:

1. **Legacy** systems

sistemi vecchi, con tecnologie superate e obsolete

Old, valuable systems must be maintained and updated → finché funziona si tiene

2. **Heterogeneity** → *ETERogeneità degli strumenti, molti sistemi operativi e linguaggi diversi all'interno di un progetto e bisogna far funzionare tutto contemporaneamente*

Increasingly, systems are required to operate as distributed systems across networks that include different types of computer and mobile devices, problem di integrazione e interoperabilità

3. **Delivery, security and trust**

Business, social and cultural change are causing an increasing pressure for **faster delivery** of software

Web-based software engineering

- ◇ Web-based systems are complex distributed systems but the **fundamental principles of software engineering** discussed previously are as applicable to them as they are to any other types of system.
- ◇ **The fundamental ideas of software engineering apply to web-based software in the same way that they apply to other types of software system.**

Professional and ethical responsibility

- Software engineering involves wider responsibilities than simply the application of technical skills:
INFATTI essi decidono come gestire un processo, che dati utilizzare, come utilizzarli, come proteggerli o meno, ecc.
- Software engineers must behave in an honest and ethically responsible way if they are to be respected as professionals
- Ethical behaviour is more than simply upholding the law.

Key points

- Software engineering is an engineering discipline which is concerned with all aspects of software production.
- Software products consist of developed programs and associated documentation. Essential product attributes are maintainability, dependability, efficiency and usability.
- The software process consists of activities which are involved in developing software products. Basic activities are software specification, development, validation and evolution.
- Methods are organised ways of producing software. They include suggestions for the process to be followed, the notations to be used, rules governing the system descriptions which are produced and design guidelines.



Key points

- CASE tools are software systems which are designed to support routine activities in the software process such as editing design diagrams, checking diagram consistency and keeping track of program tests which have been run.
- Software engineers have responsibilities to the engineering profession and society. They should not simply be concerned with technical issues.
- Professional societies publish codes of conduct which set out the standards of behaviour expected of their members.



Set 1 - Cosa ricordare: concetti, motivazioni, conseguenze, relazioni fra concetti, ecc.

- Importanza e motivazioni dell'Ingegneria del SW (IS), Mercato Italiano del SW
- ~~L'aspetto economico: i costi del SW: manutenzione vs. sviluppo; costruzione vs. testing; distribuzione dei costi dei vari processi sw~~
- SW include documentazione
- SW generico vs. SW customizzato; approcci allo sviluppo, configuration, re-use
- Definizioni e limiti di IS; system engineering; prog. in the small/in the large
- Prime definizioni di alcuni processi fondamentali
- Ragionamento ANALITICO e R. SINTETICO
- Processo SW; MODELLO dei Processi SW; modelli generici; dipendenza della distribuzione dei costi dal modello usato
- Parametri di Qualità del SW: manut, 'potersi fidare', sicurezza&protezione, efficienza, accettabilità
- Le grandi sfide attuali dell'IS: sistemi Legacy, eterogeneità hd e SW, richiesta di velocità nello sviluppo e nel delivery
- Web e IS