



Il primo passo  
indispensabile per ottenere  
dalla vita ciò che si vuole:

Il primo passo  
indispensabile per ottenere  
dalla vita ciò che si vuole:

decidere cosa si vuole  
*(cit. Ben Stein)*



# Concetti di base\*

Ingegneria del software

- \* parte del testo contenuto in queste slide è stato preso dai seguenti libri di testo:
  - I. Sommerville. Ingegneria del Software. Pearson
  - C. Ghezzi, M. Jazaheri, D. Mandrioli. Ingegneria del software fondamenti e principi. Pearson

Domanda	Risposta
Cos'è il software?	Programmi per computer e relativa documentazione. I prodotti software possono essere sviluppati per un particolare cliente o per il mercato in generale.
Cos'è l'ingegneria del software?	L'ingegneria del software è una disciplina ingegneristica che si occupa di tutti gli aspetti della produzione del software.
Qual è la differenza tra l'ingegneria del software e l'informatica?	L'informatica si occupa della teoria e dei fondamenti; l'ingegneria del software degli aspetti pratici che riguardano lo sviluppo di software di qualità.
Qual è la differenza tra l'ingegneria del software e l'ingegneria dei sistemi?	L'ingegneria dei sistemi ha come oggetto tutti gli aspetti dello sviluppo di sistemi informatici, inclusi quelli hardware, software e di processo. L'ingegneria del software è una parte di questo processo.
Cos'è il processo software?	È un insieme di attività il cui scopo è lo sviluppo o la modifica del software.
Cos'è un modello di processo software?	È una rappresentazione semplificata di un processo software visto da uno specifico punto di vista.
Quali sono i costi dell'ingegneria del software?	All'incirca il 60% dei costi è legato allo sviluppo, il 40% alle prove. Per il software personalizzato, i costi di evoluzione spesso superano quelli di sviluppo.
In cosa consistono i metodi di ingegneria del software?	Approcci strutturati di sviluppo software, che comprendono modelli di sistema, notazioni, regole, consigli di progettazione e guide al processo.
Cos'è CASE ( <i>computer-aided software engineering</i> , ingegneria del software assistita dal computer)?	Sono sistemi software creati per fornire aiuti automatizzati alle attività di lavorazione del software. I sistemi CASE sono spesso utilizzati come ausilio ai metodi.
Quali sono le caratteristiche di un buon software?	Il software deve fornire le funzionalità e le prestazioni richieste dall'utente, e deve essere mantenibile, fidato e usabile.
Quali sono le sfide chiave che l'ingegneria del software si pone?	Affrontare la crescente diversità e sviluppare software affidabile in tempi sempre più ridotti.

# Ingegneria del software: visione di insieme

- L'ingegneria del software è il settore dell'informatica che si occupa della creazione di sistemi software talmente grandi o complessi da dover essere realizzati da una o più squadre di ingegneri\*

\*C. Ghezzi, M. Jazaheri, D. Mandrilli - *Ingegneria del software fondamenti e principi*

# Testi di riferimento

- B. Bruegge, A. H. Dutoit. Object-Oriented Software Engineering: Using UML, Patterns and Java. Third edition. International edition. ISBN: 0-13-606125-7. Prentice Hall
  
- Ian Sommerville Ingegneria del software 8-9-10/  
Ed. 2007 pp. 848 ISBN 9788871923543

# Ingegneria del software: visione di insieme

- Campi di applicazione dell'ingegneria al software  
(vedere Sommerville)
  - generic
  - customized
- IEEE standard 610.12-1990 “l'ingegneria del software è l'applicazione di un approccio sistematico, disciplinato e quantificabile nello sviluppo, funzionamento e manutenzione del software”

# Ingegneria del software: visione di insieme

- Ingegneria del software è una disciplina il cui obiettivo è lo sviluppo di sistemi software di alta qualità senza sprechi
- Il termine ingegneria del software fu proposto nel 1968 durante una conferenza per discutere quella che venne chiamata la crisi del software
- La crisi del software era dovuta all'introduzione di hardware basato su circuiti integrati, la cui potenza rese disponibili applicazioni informatiche fino ad allora irrealizzabili.
- Il software risultante fu più grande e più complesso, per diversi ordini di grandezza rispetto sistemi precedenti.

# Ingegneria del software: visione di insieme

- L'esperienza fatta nella costruzione di questi sistemi ha mostrato che lo sviluppo informale del software non era una soluzione **adatta**
- I progetti più grandi erano spesso in ritardo di anni, con costi molto superiori al previsto
- Non erano affidabili
- Erano difficili da mantenere
- Avevano pessime prestazioni
- Sviluppo del software in crisi
- Il costo dell'hardware scendeva vertiginosamente
- Il costo del software saliva a dismisura

# Ingegneria del software: visione di insieme

- Ingegneria del software non è programmazione
- programmatore - scrive un programma completo
- Ing. del sw - **scrive un componente** sw che viene combinato con altri componenti sw scritti da altri ing. dello stesso team (o di team diversi) per creare un sistema completo

# Ingegneria del software: visione di insieme

- Programmare è un' attività individuale
- Progettare è (spesso) un'attività di gruppo

# Ingegneria del software: visione di insieme

# Ingegneria del software: visione di insieme

Da Bruegge, Dutoit. Object Oriented Software Engineering  
using UML, patterns and Java

# Ingegneria del software: visione di insieme

Da Bruegge, Dutoit. Object Oriented Software Engineering  
using UML, patterns and Java

The term software engineering was coined in 1968 as a response to the desolate state of the art of developing **quality** software **on time** and within **budget**.

# Ingegneria del software: visione di insieme

Da Bruegge, Dutoit. Object Oriented Software Engineering using UML, patterns and Java

The term software engineering was coined in 1968 as a response to the desolate state of the art of developing **quality** software **on time** and within **budget**.

Software developers were not able to set concrete objectives, predict the resources necessary to attain those objectives, and manage the customers' expectations.

# Ingegneria del software: visione di insieme

Da Bruegge, Dutoit. Object Oriented Software Engineering using UML, patterns and Java

The term software engineering was coined in 1968 as a response to the desolate state of the art of developing **quality** software **on time** and within **budget**.

Software developers were not able to set concrete objectives, predict the resources necessary to attain those objectives, and manage the customers' expectations.

More often than not, the moon was promised, a lunar rover built, and a pair of square wheels delivered.

# Ingegneria del software: visione di insieme

Da Bruegge, Dutoit. Object Oriented Software Engineering using UML, patterns and Java

The term software engineering was coined in 1968 as a response to the desolate state of the art of developing **quality** software **on time** and within **budget**.

Software developers were not able to set concrete objectives, predict the resources necessary to attain those objectives, and manage the customers' expectations.

More often than not, the moon was promised, a lunar rover built, and a pair of square wheels delivered.

The emphasis in software engineering is on both words, *software* and *engineering*.

# Ingegneria del software: visione di insieme

Da Bruegge, Dutoit. Object Oriented Software Engineering using UML, patterns and Java

The term software engineering was coined in 1968 as a response to the desolate state of the art of developing **quality** software **on time** and within **budget**.

Software developers were not able to set concrete objectives, predict the resources necessary to attain those objectives, and manage the customers' expectations.

More often than not, the moon was promised, a lunar rover built, and a pair of square wheels delivered.

The emphasis in software engineering is on both words, *software* and *engineering*.

An engineer is able to build a high-quality product using off-the-shelf components and integrating them under time and budget constraints. The engineer is often faced with ill-defined problems and partial solutions, and has to rely on empirical methods to evaluate solutions

# Informatica e ingegneria del software

- La principale differenza tra informatica e ingegneria del software è:
- Informatica - si occupa delle teorie e dei metodi che stanno alla base dei sistemi software e di quelli informatici
- Ingegneria del software - si occupa dei problemi pratici relativi alla produzione del software

# Ingegneria del software: visione di insieme

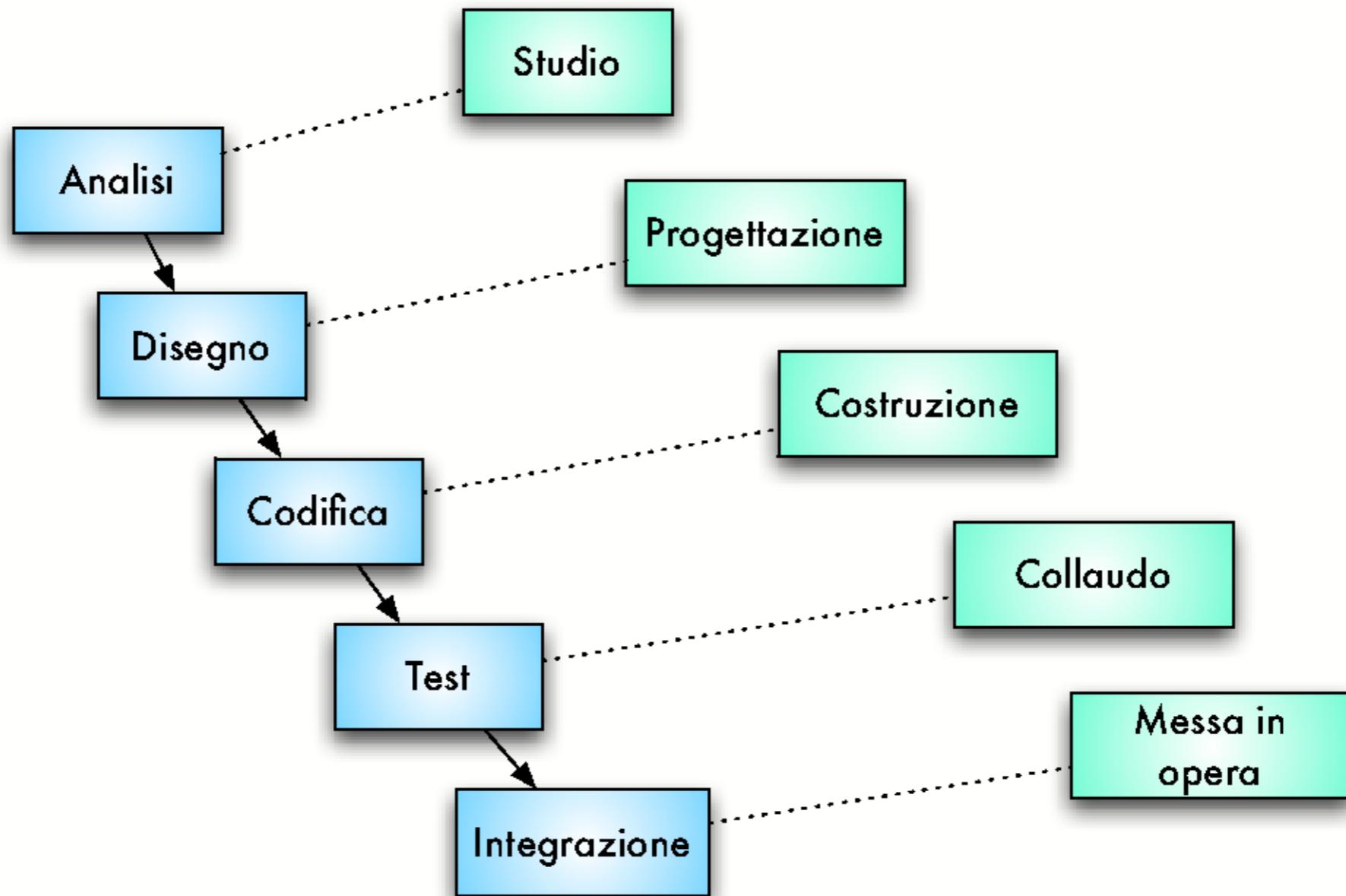
- Il SW subisce uno sviluppo e un'evoluzione dall'idea iniziale di un possibile prodotto o sistema software fino a quando viene implementato e consegnato al cliente.
- Il software ha un ciclo di vita composto da varie fasi
- A ciascuna di queste è associato lo sviluppo di una parte del sistema o di qualche elemento a questo legato come ad esempio un manuale per l'utente o il piano di test



# Processo Software

- È l' insieme delle attività e dei risultati che creano un prodotto software
- Quattro attività fondamentali comuni a tutti processi
  - Specifica: il cliente e gli ingegneri definiscono cosa deve fare il software e quali sono i vincoli per la sua progettazione
  - Sviluppo: progettazione programmazione
  - Convalida: si verifica che il software sia esattamente quello che il cliente richiede
  - Evoluzione: si modifica il prodotto per adeguarlo ai requisiti dell' utente e del mercato che cambiano

# Waterfall



# Metodo di ingegneria del software

- Un metodo di ingegneria del software è un approccio strutturato allo sviluppo del software, che facilita una produzione di alta qualità e a costi contenuti
- Non c'è un metodo ideale e diversi metodi possono essere usati in aree diverse
- Per esempio i metodi orientati agli oggetti sono solitamente adatti a sistemi interattivi ma non a sistemi che devono essere strettamente real time

# CASE tool

- L'acronimo CASE significa Computer aided software engineering
- Comprende una vasta gamma di programmi diversi usati per aiutare le attività di processo del software
- dall' analisi dei requisiti al debugging e testing

# Cosa è un modello

- Un artefatto che descrive il sistema/software
  - per esempio un diagramma o un insieme di diagrammi
- Un insieme di affermazioni su un sistema di interesse o sotto studio

# Cosa è un modello

DA WIKIPEDIA, L'ENCICLOPEDIA LIBERA.

NELL'USO SCIENTIFICO E TECNICO-PROGETTUALE, UN MODELLO È UNA RAPPRESENTAZIONE DI UN OGGETTO O DI UN FENOMENO, CHE CORRISPONDE ALLA COSA MODELLATA PER IL FATTO DI RIPRODURNE (EVIDENTEMENTE ALLA LUCE DI PRECISI RISCONTRI FATTUALI OTTENUTI A PARTIRE DA UN METODO CHE SIA GARANZIA DI CONTROLLABILITÀ) ALCUNE CARATTERISTICHE O COMPORTAMENTI FONDAMENTALI; IN MODO TALE CHE QUESTI ASPETTI POSSANO ESSERE MOSTRATI, STUDIATI, CONOSCIUTI LADDOVE L'OGGETTO MODELLATO NON SIA DIRETTAMENTE ACCESSIBILE.

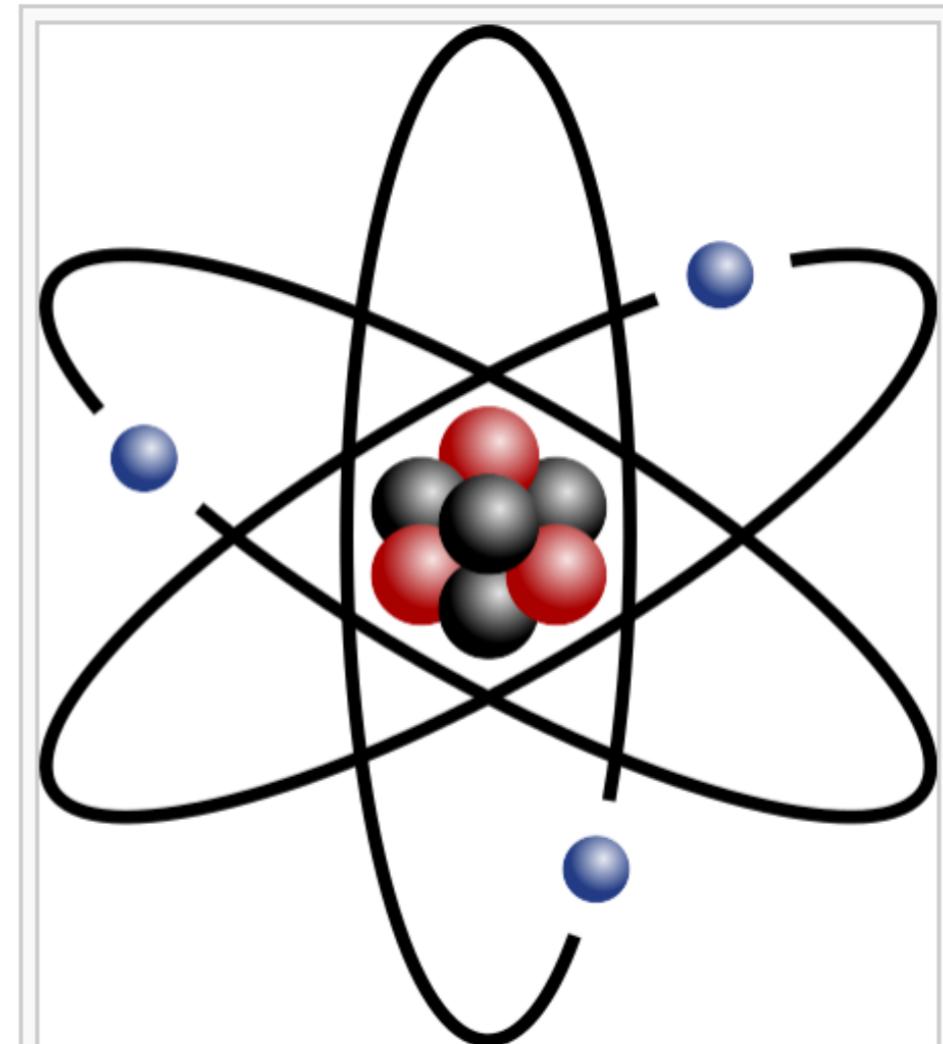


Immagine stilizzata del **Modello atomico di Bohr**, che si propone di spiegare struttura e funzionamento degli atomi

# Cosa è un modello



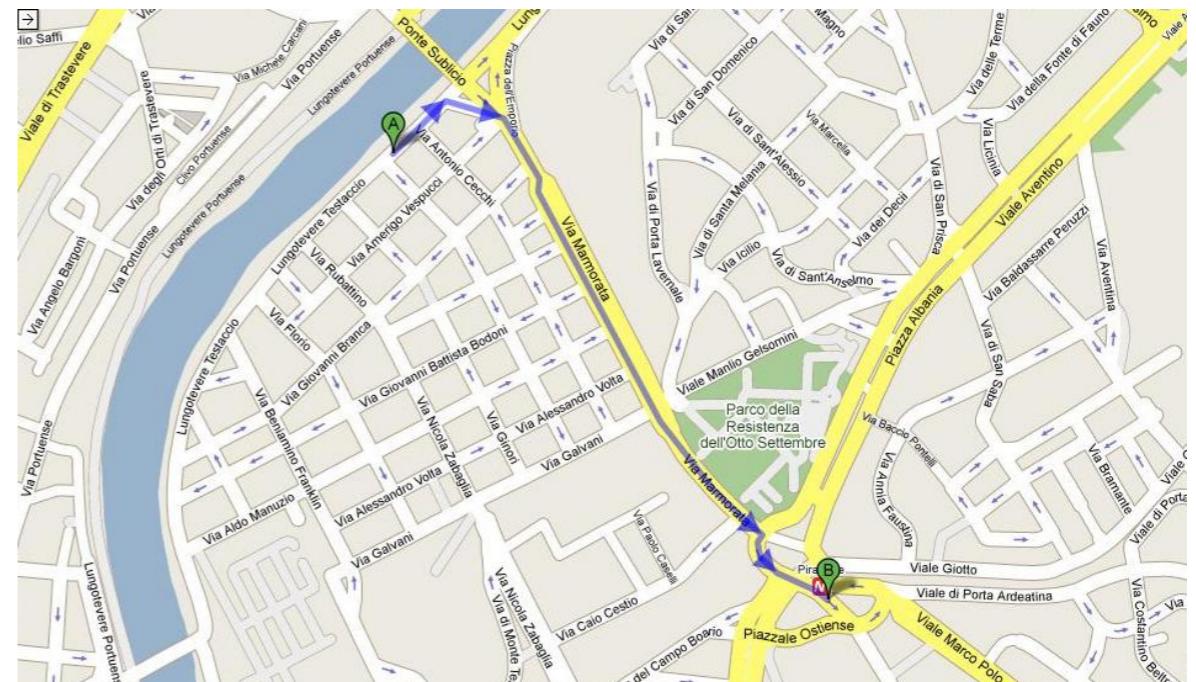
**chair** (tchère), n., chaise, f.; siège, m.; (of a professor) chaire, f.; (of the chairman or president of an assembly) fauteuil, m.; (rail) coussinet, m. Arm—; fauteuil. Bath —; viari-grette, f. Sedan—; chaise à porteurs, f. Easy —; bergère, f. Rocking—; chaise berceuse, f. Deck —; chaise longue, f.; pliant, m. To be in the —; occuper le fauteuil. —! —! à l'ordre / à l'ordre! To fill the —; présider. To leave the —; lever la séance. With . . . in the —; sous la présidence de . . .

© Thomas Kühne

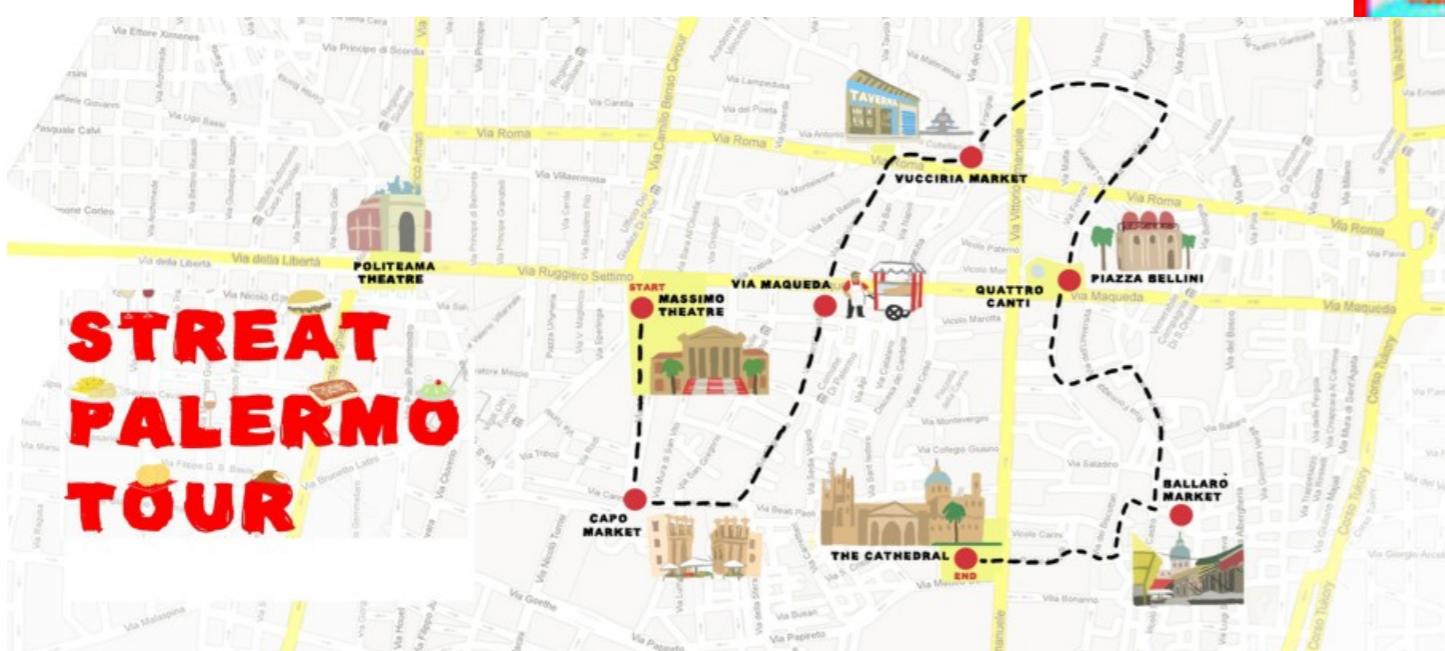
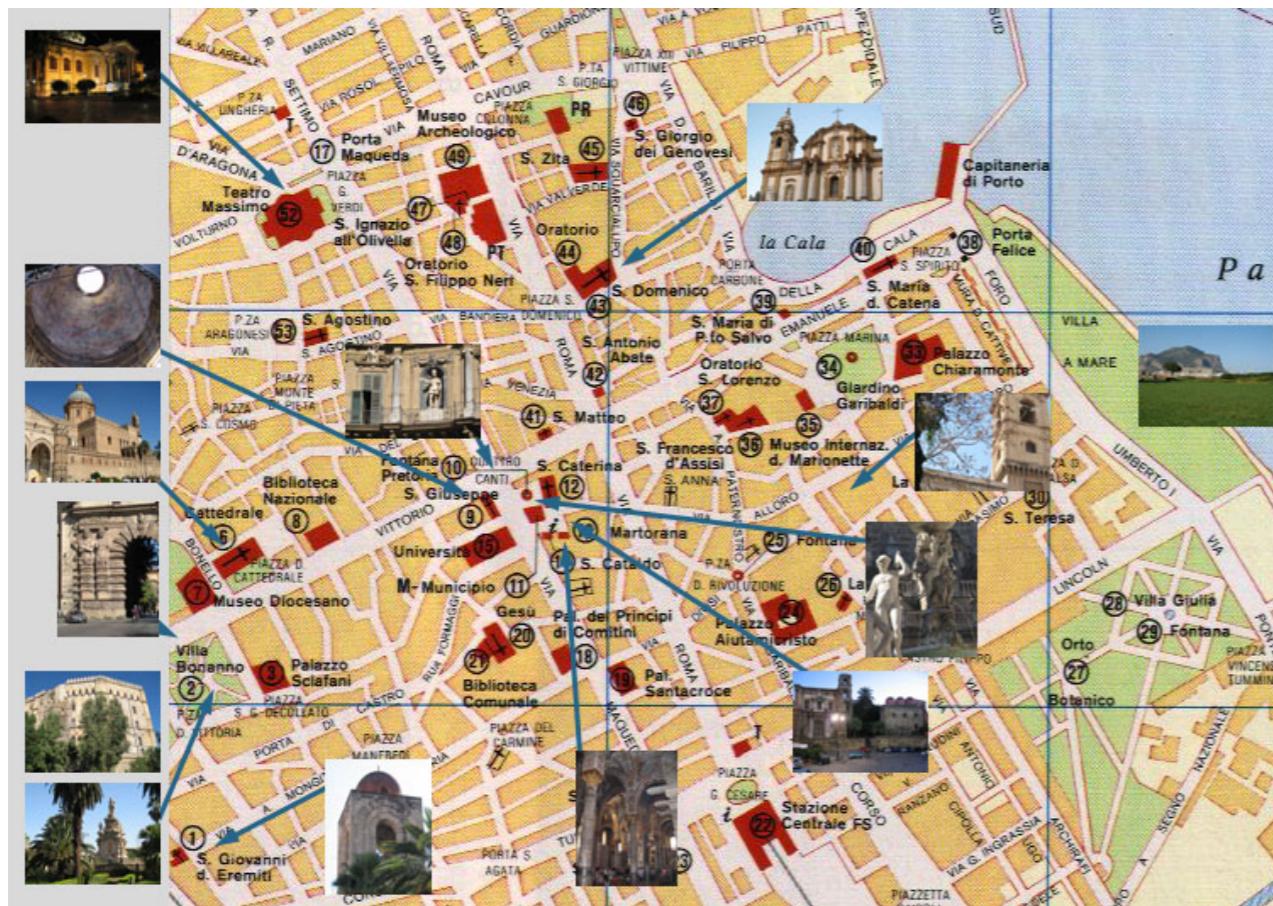
# Cosa è un modello



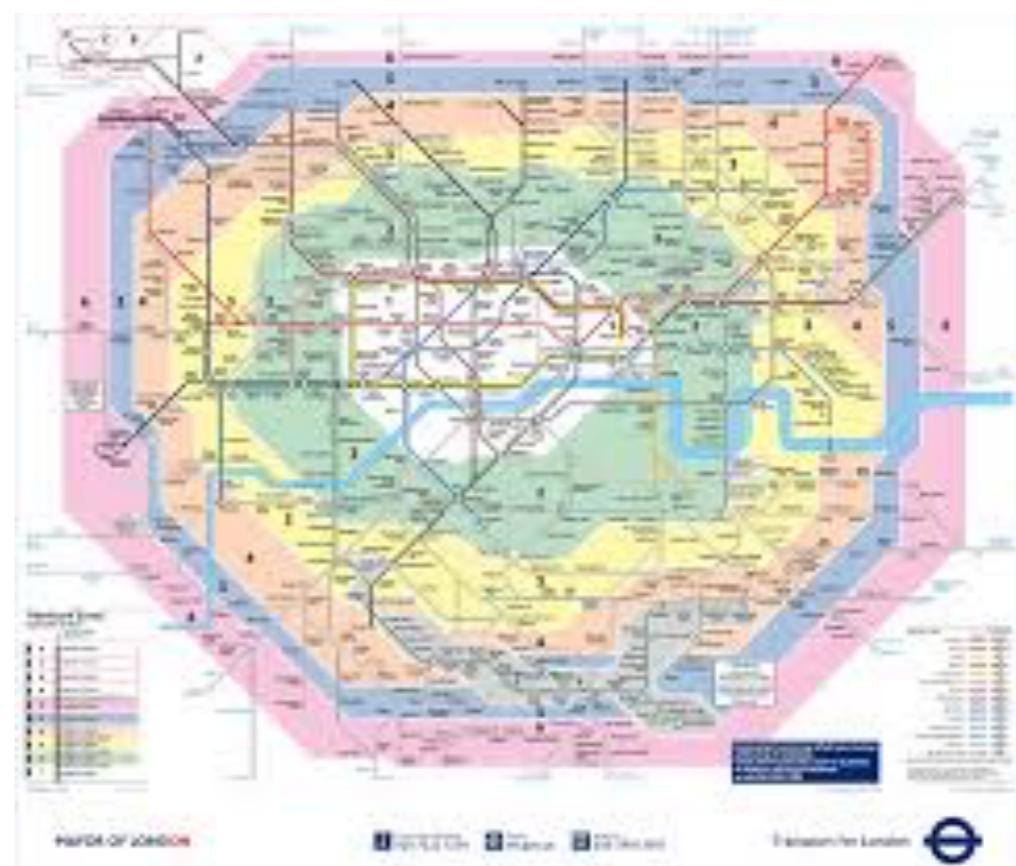
© Thomas Kühne



Un modello cattura un aspetto  
di interesse, fornisce una vista



Viste differenti sullo stesso oggetto



# Viste e notazione



Ancora sui modelli: La notazione

# Linguaggio di modellazione

- Un modello è sviluppato (disegnato) sulla base di una notazione
- La notazione dipende dal linguaggio di modellazione
- Linguaggio: sintassi e semantica

# UML cenni

- UML è un linguaggio di modellazione
- UML-unified modeling language
- Linguaggio di modellazione unificato
- Famiglia di notazioni grafiche che si basano su un singolo meta modello
- Servono a supportare la descrizione e il progetto del software, in particolare quelli costruiti seguendo il paradigma orientato agli oggetti

# UML come

- UML non fornisce alcuna metodologia di modellazione
- Alcuni aspetti metodologici sono impliciti negli elementi che costituiscono un modello UML
- UML fornisce esclusivamente la sintassi visuale da utilizzare per la costruzione dei modelli

# Unified Process-UP

- Una metodologia: indica le risorse, le attività e i manufatti che dobbiamo utilizzare, effettuare e creare per modellare un sistema software

# Progettazione e modellazione

