



UNIVERSITÀ DEGLI STUDI DI NAPOLI
FEDERICO II

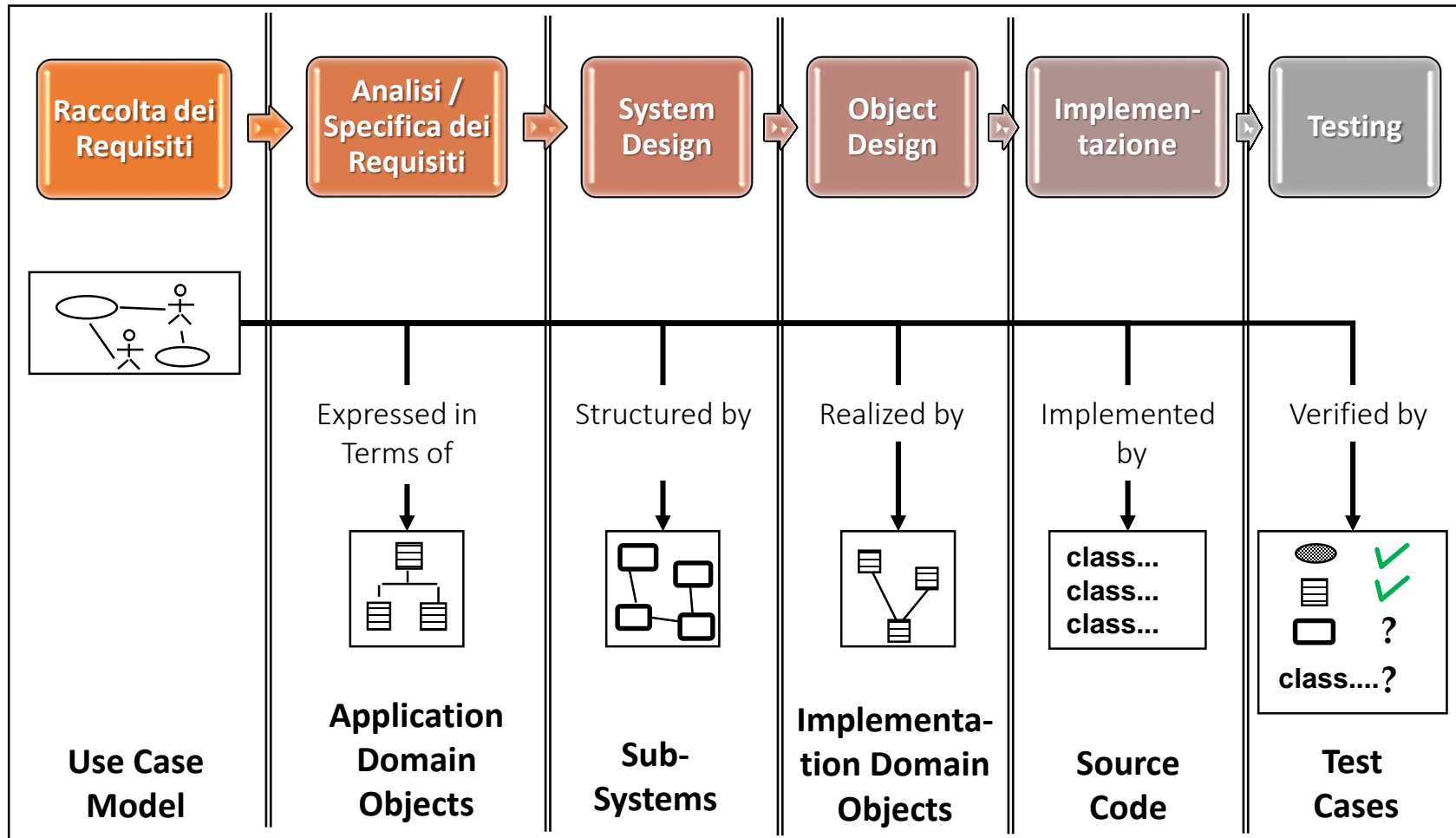
Ingegneria del Software – Introduzione al Corso

Prof. Sergio Di Martino

Contenuti della lezione

- Cos'è l'Ingegneria del Software
- Informazioni pratiche sul corso
- Informazioni sul docente
- Introduzione all'Ingegneria del Software
 - Discutere i concetti di Prodotto Software
 - Introdurre il concetto di Ciclo di Vita del Software

Ciclo di Vita del Software



Dimensioni di un tipico software

- Dimensioni in termini di Linee di Codice per alcuni sw di uso comune:

Sistema	Linee di codice
App Media per iOS	400.000
Firefox	18.000.000
Windows 7	40.000.000
Vettura alto di gamma	100.000.000

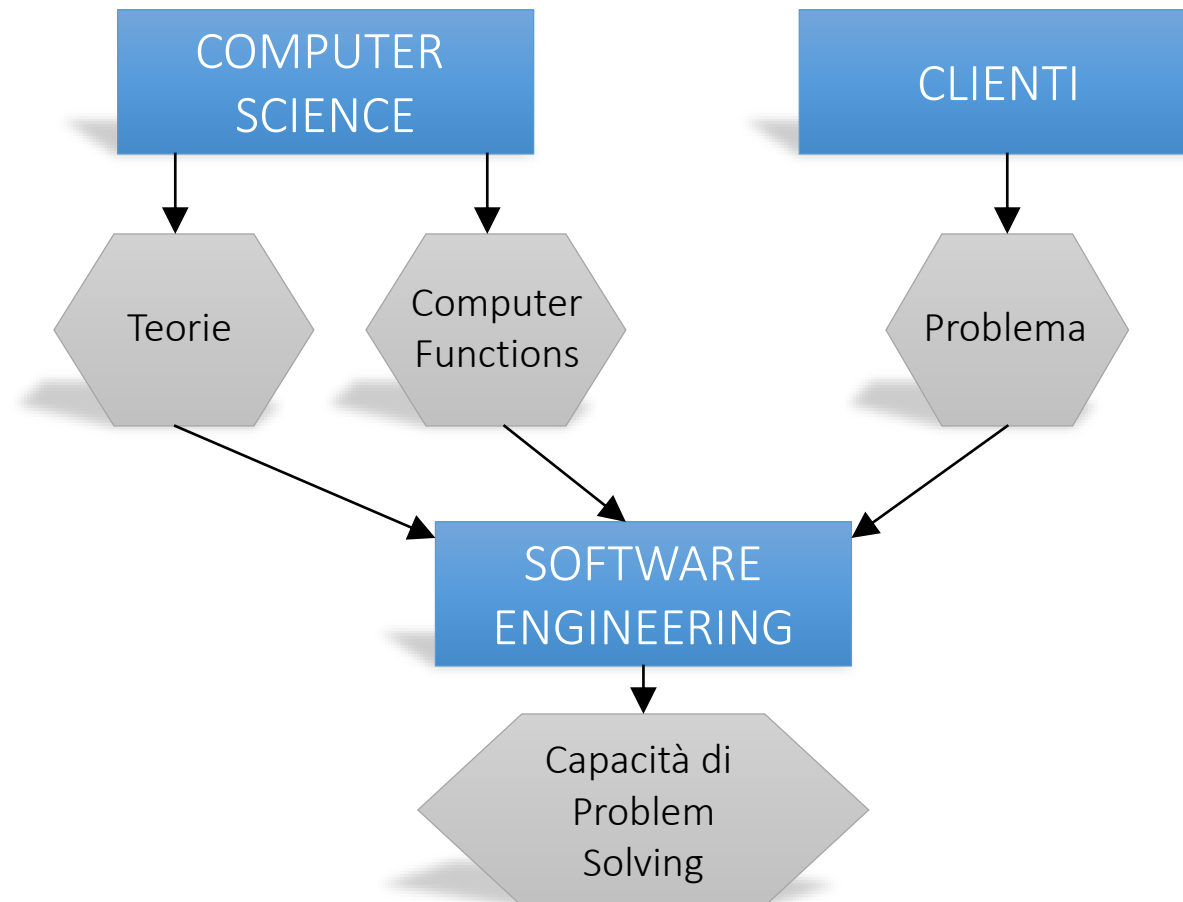
L'Ingegneria

- L'ingegneria propone ***metodologie di sviluppo***, che riassumono e formalizzano esperienze e conoscenze pregresse
 - Edilizia
 - Marco Vitruvio Pollione, *De Architectura* (1° Secolo A.C.)
 - Aeronautica, Automobili, etc...
 - Elettronica
 - Nucleare

Lo sviluppo software

- Molti fattori hanno storicamente limitato l'utilizzo di approcci ingegneristici nella produzione di software
 - L'intangibilità del software
 - Disciplina (relativamente) nuova
 - Approccio "artistico", non strutturato allo sviluppo
 - Mancanza di formalismi e strumenti di supporto

Ruolo dell'Ingegneria del Software



L'Ingegneria del Software

- Definizione IEEE (2007)
 - *“L'ingegneria del software è la disciplina tecnologica e manageriale che riguarda la produzione sistematica e la manutenzione dei prodotti software entro tempi e costi preventivati”*
- Informalmente: sviluppo cost-effective di software di **qualità**



Informazioni sul corso

Ingegneria del Software

- Conoscenze che si intendono trasmettere (sapere):
 - Concetti di base dell'ingegneria del software, dei processi di ingegneria del software e delle relative fasi, attività e deliverable;
 - Metodi di analisi e progettazione, e importanza dei linguaggi di modellazione del software per la comunicazione tra diversi attori coinvolti in un processo di ingegneria del software;
 - Concetti e tecniche di testing e validazione del software;
 - Concetti di manutenzione del software, e principali problematiche della gestione dei progetti software

Ingegneria del Software

- Capacità che si intendono sviluppare (saper fare):
 - Saper usare un approccio ingegneristico/strutturato all'analisi, design e testing del software.
 - Saper comprendere e produrre gli appropriati documenti durante tutte le varie fasi del processo di sviluppo software
 - Sapere usare la notazione UML per modellare il software
 - Saper sviluppare software di qualità

Informazioni pratiche

- Corso obbligatorio per la laurea triennale
 - Nuovo ordinamento (Mat. N86): 9 CFU
- Svolgimento
 - I semestre: 6 ore settimanali
 - Presenza non obbligatoria ma fortemente consigliata

Comunicazioni

- Gruppo Google per la comunicazione-discussione, aggiornamenti e materiale didattico:

UNINA ING SW 2016-17

- Indirizzo per chiedere adesione

<https://groups.google.com/d/forum/unina-ing-sw-2016-17>

- Nella richiesta si DEVE specificare

Cognome Nome Matricola

- Ulteriori riferimenti sul Sito del docente:

www.docenti.unina.it/sergio.dimartino

Modalità di esame

- Progetto obbligatorio di gruppo
 - Analisi, Progettazione, Implementazione e Testing di un piccolo sistema software
 - Presentazione di gruppo al docente del progetto.
- Scritto: esercizi e domande sull'intero programma.
 - Obbligatorio.
- Orale Facoltativo per chi abbia avuto allo scritto un voto ≥ 21
- Voto: Media di Progetto e Scritto

Progetto come gioco di ruolo

- Un'unica traccia, declinata diversamente
- Committente: Docente
 - Saranno fornite specifiche incomplete e potenzialmente inconsistenti.
 - Le specifiche vanno raffinare in incontri programmati e contingentati (per numero e durata).
- Azienda produttrice: Gruppo di studenti.
- Si dovranno produrre tre documenti, oltre al codice
 - Analisi dei requisiti
 - Progettazione di sistema
 - Progettazione dei casi di test

Valutazione del Progetto

- Valutazione dell'intera interazione committente-contraente.
 - Interazione e uso degli strumenti di comunicazione.
 - Qualità grafica dei documenti prodotti.
 - Qualità della progettazione.
 - Qualità della presentazione finale.
- E' necessaria una valutazione almeno sufficiente su tutti gli artefatti prodotti.

Formazione dei gruppi

- Consistenza numerica: Min 2 persone Max 4 persone
 - Esperienza di lavoro in team.
 - Saranno ammessi gruppi singoli solo per motivati e documentati impedimenti (ad esempio lavorativi)
- Formazioni dei gruppi
 - Autonome comunicate nel gruppo del corso
 - Operate dal docente in base alle disponibilità per studenti che non riescano a stabilire formazioni autonome.
- Variazione dei gruppi
 - Ogni variazione di un gruppo ufficializzato deve essere concordata con il docente

Cheating Policy

- In caso di due o più progetti ritenuti troppo simili, il progetto sarà annullato ad entrambi i gruppi e sarà data una nuova traccia, più estesa e complessa di quella originaria
- Detection rate: almeno una coppia di progetti l'anno.

Materiali di studio (1)

- Libri di testo generali consigliati
- Parti generali
 - I. Sommerville. Software Engineering, VIII ed., Addison Wesley, 2007.
- Progettazione ad oggetti
 - C. Larman, Applicare UML e i Pattern - Analisi e Progettazione orientata agli Oggetti, III ed. Prentice-Hall, 2005.
 - B. Bruegge, A. Dutoit. Object-Oriented Software Engineering, Pearson, 2008. (Alternativo a C. Larman).

Materiali di studio (2)

- Libri di testo generali consigliati
- UML
 - Stevens Rod Pooley, Usare UML, Addison Wesley, 2008.
 - J. Arlow, Ila Neustadt, UML2 e Unified Process, McGraw-Hill, 2006.
- Altri testi su aspetti specifici
 - P. Amman, J. Offutt. Introduction to software testing, Cambridge University Press, 2008.
 - E. Gamma, R. Helm, R. Johnson, J. Vissides. Design patterns, Addison Wesley

Materiali di studio (3)

- Lucidi delle lezioni
 - I lucidi saranno disponibili on-line.
 - I lucidi delle lezioni **non sono sostitutivi** dei libri di testo.
- Esempio di un progetto svolto.
- Materiali vari indicati di volta in volta al corso.

Informazioni sul docente

Prof. Sergio Di Martino

Contatti

- Ufficio: Stanza T.03, via Claudio 21
- Ricevimento: Venerdì 11:15 – 12:30, auletta P, MSA
- Sito del docente:
www.docenti.unina.it/sergio.dimartino

Comunicazione via mail

- Mail:
 - sergio.dimartino@unina.it
 - Subject: [INGSW1] *e poi l'oggetto*
 - Firmare SEMPRE le mail
 - Non mandare mail per quesiti su aspetti già descritti nel sito web e/o sul gruppo Google.
 - Le mail che non rispettano queste regole non solo non riceveranno risposta, ma saranno fonte di valutazione negativa!
- La comunicazione cliente/committente è uno degli aspetti chiave nell'Ingegneria del Software!

L'Ingegneria del Software: Introduzione

Evoluzione della Produzione di Software

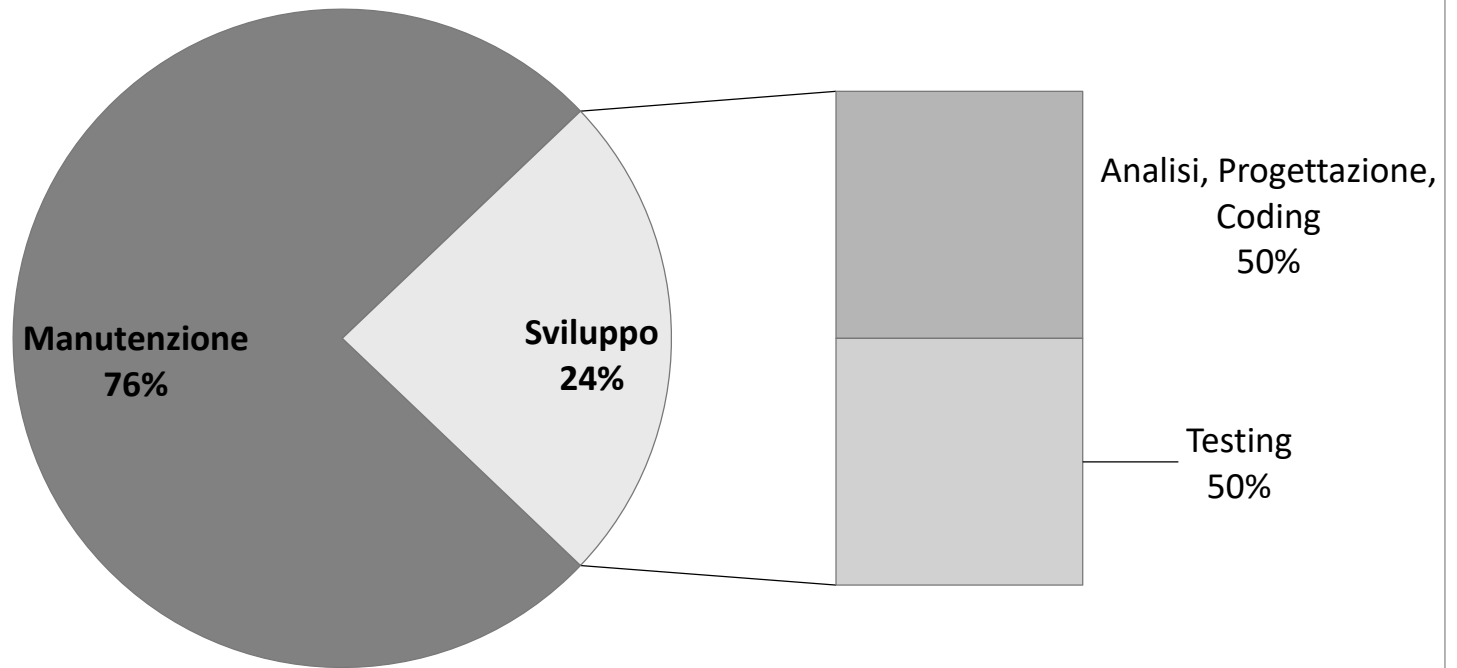
- **Arte:** applicazioni sviluppate da singole persone e utilizzate dagli stessi sviluppatori
- **Artigianato:** applicazioni sviluppate da piccoli gruppi specializzati per un cliente
- **Industria:** diffusione del software in molteplici settori (tutti?); crescita di dimensioni, complessità e criticità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

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
 - Tipico costo di un programmatore per un'azienda: da 5.000 a 10.000€/mese
 - 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

Software e costi

Costi durante la vita di un software



Tendenza sulle professioni più retribuite (2011)

- Registered Nurses \$64.690
- Accountants and Auditors \$61.690
- Management Analysts \$78.160
- **Computer Application Software Engineers \$91.180**
- Physicians and Surgeons \$94.000-153.000
- **Computer Systems Analysts \$77,740**
- Market Research Analysts \$60.570
- Civil Engineers \$77,650
- Dental Hygienist \$68.250
- Personal Financial Advisor \$64.750

Computer Application Software Engineers

- Total new jobs (2008-2018): 175,100
- Pct. increase: 34%
- Median income: 94,180
- States with most jobs per capita: Washington, Colorado, Virginia
- *In the age of the mobile app and cloud computing, few positions are becoming as essential as software application developers. In 2008, 514,000 people worked in this position. By 2018, the BLS National Employment Matrix projects there will be 175,000 additional positions. According to the BLS's description of the job prospects for software developers, "As a result of rapid employment growth over the 2008 to 2018 decade, job prospects for computer software engineers should be excellent. Those with practical experience and at least a bachelor's degree in a computer-related field should have the best opportunities." Those willing to put in the time to get certified will be well compensated. The median annual income is nearly \$100,000.*

Computer System Analysts

- Total new jobs (2008-2018): 108.010
- Pct. increase: 20%
- Median income: 77.740
- States with most jobs per capita: Virginia, Delaware, new Jersey
- *Computer systems analysts build and manage computer networks for companies for use in file sharing and inter-office communication. They also maintain web security within the network. This position, only 30-years-old, is expected to grow by 20%, adding nearly 110,000 jobs before the end of this decade. While an increasing number of businesses are requiring bachelor's degrees, it is still very possible to get a job as a systems analyst with just a high school diploma and some certification. The median annual income, \$77,740, is also one of the best salaries one can get without a college degree.*