



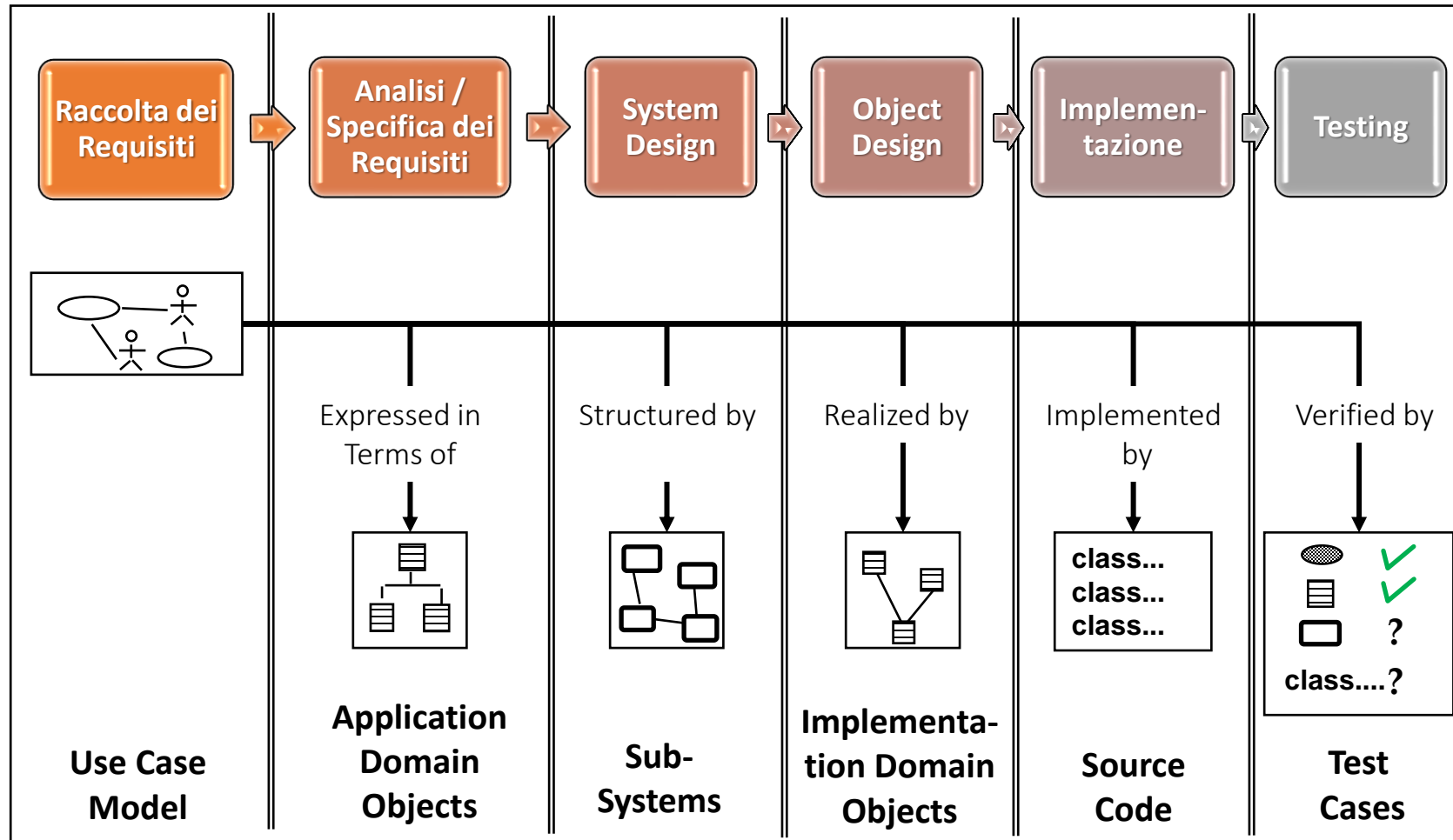
UNIVERSITÀ DEGLI STUDI DI NAPOLI  
**FEDERICO II**

# Ingegneria del Software – Mod.A e B

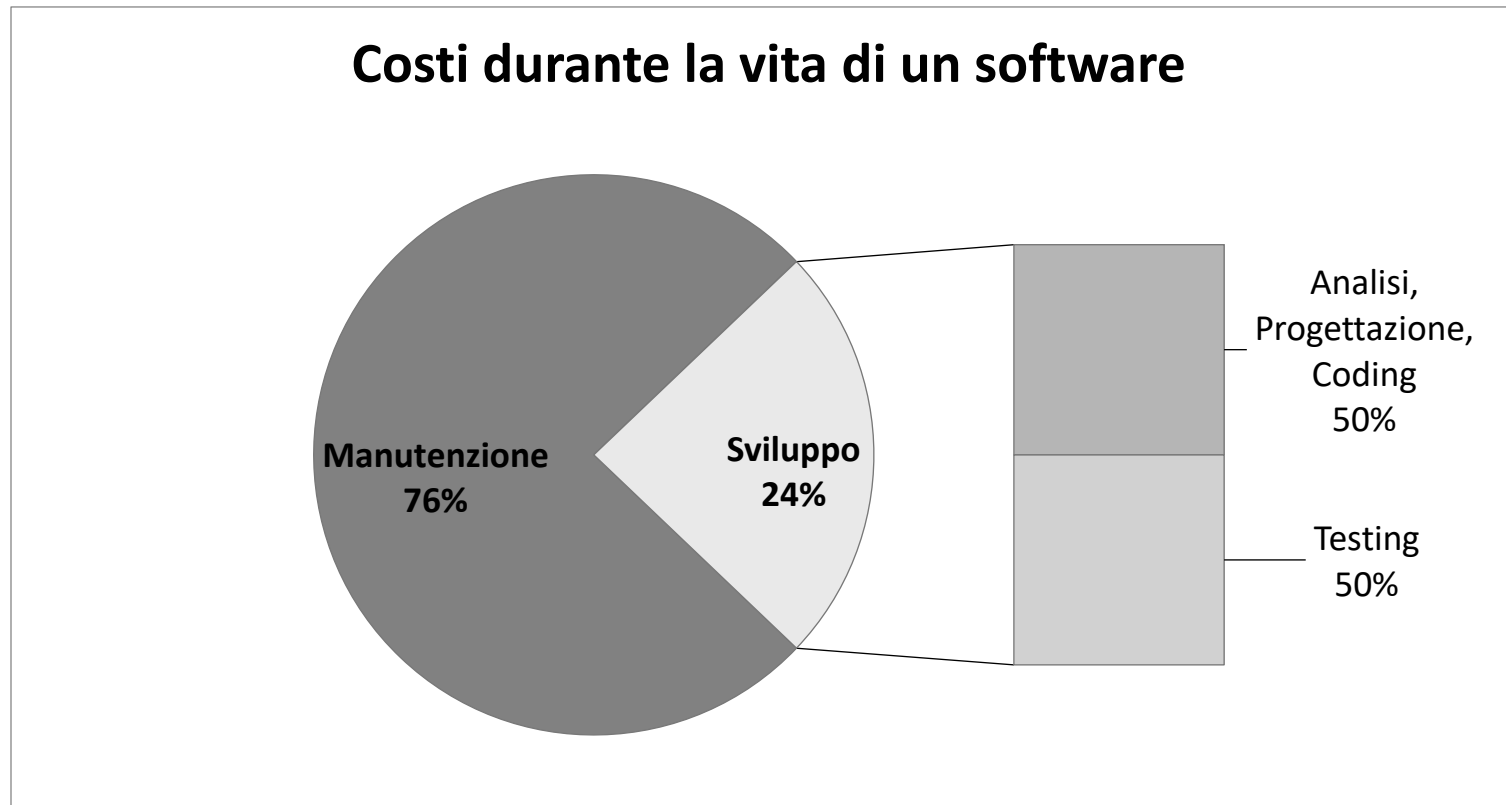
## Introduzione al Corso

Prof. Sergio Di Martino – Prof. Franco Cutugno

# Ciclo di Vita del Software



# Software e costi



# 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	37.132.571
Windows 11	≈50.000.000
Vettura alto di gamma	>100.000.000
Ecosistema Google	>2.000.000.000

- [https://www.openhub.net/p/firefox/analyses/latest/languages\\_summary](https://www.openhub.net/p/firefox/analyses/latest/languages_summary)

# Lo sviluppo del software

- Approccio Naive:



- Da dove vengono le specifiche?
- Le specifiche corrispondono alle vere esigenze del cliente?
- Come si decide l'architettura/struttura del Programma?
- Come suddividiamo il lavoro tra più programmatori?
- Come sappiamo se il programma davvero implementa le specifiche?
- Il programma funziona sempre correttamente?
- Come gestiamo modifiche delle specifiche?

# L'Industria del Software

- Produrre software non è (solo) un'arte e neppure (solo) una scienza: è un'*industria*
  - Si lavora sempre in un contesto di gruppo e di azienda
  - vincoli economici e requisiti di qualità
- Come in ogni industria, per produrre software sono state sviluppate *metodologie* di progetto, di sviluppo e di verifica
- Il laureato in Informatica *deve* conoscerle!
  - Non basta essere i migliori programmatori possibili
  - Si deve sapere come ANALIZZARE, PROGETTARE e VALIDARE un software nella sua interezza, con particolare enfasi anche sulle interfacce grafiche

# 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
- 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

# L'Ingegneria del Software

- Alcune definizioni
  - *"The application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software; that is, the application of engineering to software " [IEEE]*
  - *"State of the art of developing **quality** software on time and within budget"*



# Il Concetto di Qualità del Software

# Qualità del Software

- Quality (IEEE SE Glossary).
  1. *The degree to which a system, component, or process meets specified requirements.*
  2. *The degree to which a system, component, or process meets customer or user needs or expectations.*
- Gran parte della ricerca nel campo dell'ingegneria del software è dedicata, direttamente o indirettamente, al tema della **qualità del prodotto software**.

# Qualità Esterna e Qualità Interna

- I fattori rispetto a cui si può misurare la qualità del software vengono classificati in:
  - **Fattori Interni** - la qualità del software percepita dagli sviluppatori
  - **Fattori Esterni** - la qualità del software percepita dagli utenti

# (Alcune) Qualità Interne del SW

- **Riparabilità**

- Un sistema è riparabile se la correzione degli errori è agevole. La riparabilità si persegue attraverso la modularizzazione e opportune metodologie di progettazione.

- **Manutenibilità**

- Facilità di apportare modifiche a sistema realizzato.

- **Riusabilità**

- Facilità con cui è possibile riutilizzare parti di sistema per realizzare un prodotto diverso.

- **Verificabilità**

- Un sistema è verificabile se le sue proprietà di correttezza e di affidabilità sono facilmente validabili.

# (Alcune) Qualità Esterne del SW

- **Usabilità**

- Un sistema è facile da usare se un essere umano lo reputa tale.

- **Robustezza**

- La robustezza di un sistema è la misura in cui il sistema si comporta in modo ragionevole in situazioni impreviste, non contemplate dalle specifiche.

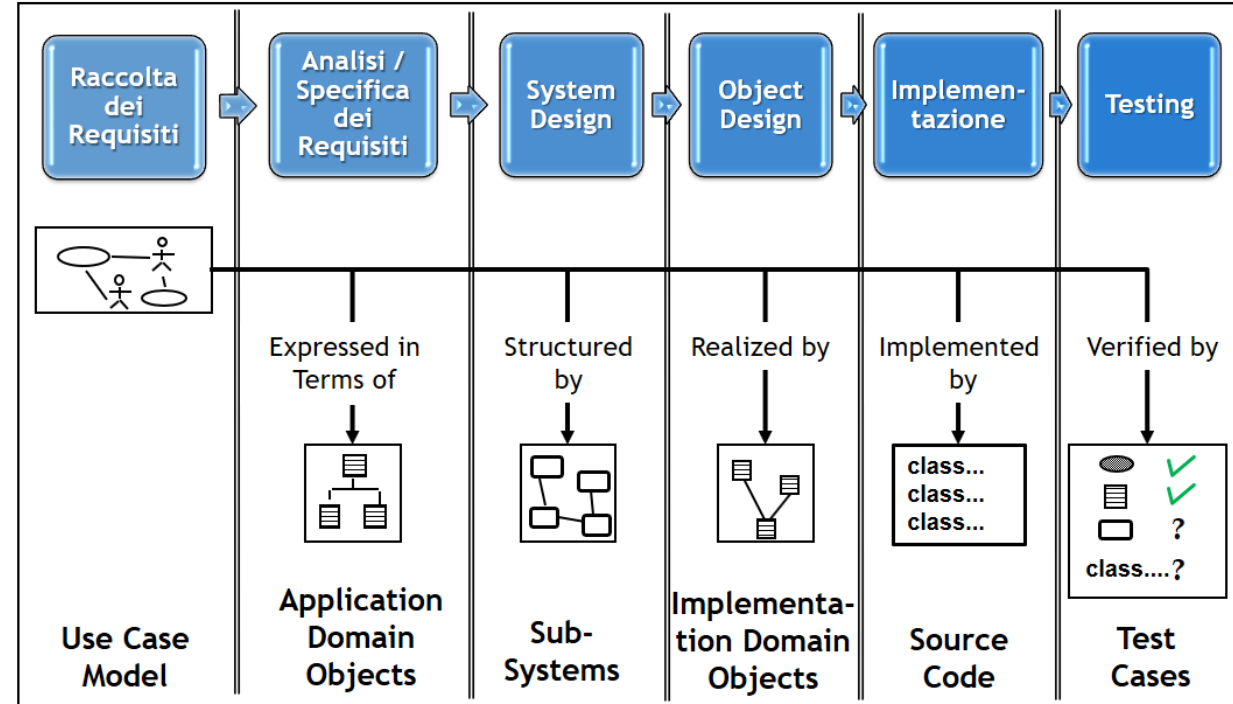
- **Affidabilità**

- Un sistema è tanto più affidabile quanto più raramente, durante l'uso del sistema, si manifestano malfunzionamenti.

# Informazioni sul corso

# Organizzazione del corso

- ~40 lezioni:
  - 4-5 lezioni per argomento
  - 3-4 seminari di aziende
  - Esercitazioni



# Obiettivi del Corso di Ingegneria del Software

- Conoscenze che si intendono trasmettere (sapere):
  - Metodi e tecniche per sviluppare software di migliore qualità
    - Modulo A: Focus su qualità INTERNE del codice
    - Modulo B: Focus su qualità ESTERNE del codice
  - Concetti di base dell'ingegneria del software, dei processi di ingegneria del software e delle relative fasi, attività e deliverable;
  - Concetti di base dell'interazione uomo-macchina, e dei paradigmi di interazione
  - Metodi di analisi e progettazione di un sistema software e della sua GUI. 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 e tecniche di valutazione dell'usabilità di un software
  - Principali problematiche e metodologie per la gestione dei progetti software



# Comunicazioni Docente → Studenti

- Sito Istituzionale su [docenti.unina](http://docenti.unina.it)
- Siti dei docenti:  
    [www.docenti.unina.it/sergio.dimartino](http://www.docenti.unina.it/sergio.dimartino)  
    [www.docenti.unina.it/francesco.cutugno](http://www.docenti.unina.it/francesco.cutugno)

# Comunicazioni Studenti → Docente

- **NO Chat di Teams**
- Mail:
  - Solo per quesiti brevi! **Per quesiti articolati esiste il ricevimento**
  - [sergio.dimartino@unina.it](mailto:sergio.dimartino@unina.it) oppure [cutugno@unina.it](mailto:cutugno@unina.it)
  - Subject: [INGSW] *e poi l'oggetto*
  - Firmare SEMPRE le mail
  - Non mandare mail per quesiti su aspetti già descritti nel sito istituzionale e/o nel materiale didattico.
  - 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!

# Modalità di esame

- Progetto obbligatorio di gruppo
  - Analisi, Progettazione, Implementazione e Testing di un piccolo sistema software
  - Presentazione di gruppo ai docenti
    - Documentazione + Demo/Powerpoint max 20 min.
- Scritto:
  - Esercizi e domande aperte sull'intero programma.
  - Obbligatorio
  - Può essere sostenuto nei 12 mesi successivi alla consegna del progetto
- Orale obbligatorio se il voto del progetto è superiore di almeno 5 punti a quello dello scritto
- Voto: Media di Progetto e Scritto (ed eventuale orale)

# Progetto come gioco di ruolo

- Un'unica traccia, declinata differentemente per i vari gruppi
- Committente: Docenti
  - 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

# Presentazione Progetto

- Ogni gruppo, a valle della consegna del progetto, dovrà concordare con i docenti una data per la presentazione (in orario di ricevimento, se non diversamente specificato).
- La presentazione si articola in 3 fasi:
  1. Presentazione **tecnica** del progetto (Slides)
    - **15 minuti per convincerci di aver sviluppato un prodotto di alta qualità interna/esterna**
  2. Demo dell'applicativo dopo la presentazione
  3. Discussione del codice
- Vale il concetto di Collective Ownership

# Valutazione del Progetto

- Qualità della progettazione, presentazione e demo.
- **Valutazione dell'intera interazione committente-contraente.**
  - Interazione e uso degli strumenti di comunicazione.
  - Qualità grafica dei documenti prodotti.
  - Qualità della presentazione finale.
  - Capacità di rispettare le indicazioni del docente nella consegna
- E' necessaria una valutazione almeno sufficiente su tutti gli artefatti prodotti.
- Il progetto dell'a.a. 2023-24 può essere consegnato entro fine Ottobre 2024.
  - Traccia semplificata per chi consegna entro la prima sessione

# Progetto e Scritto

- Lo scritto può essere sostenuto solo \*dopo\* aver consegnato il progetto.
- Dalla data di consegna del progetto, si ha un anno di tempo per sostenere lo scritto.

# Formazione dei gruppi

- Consistenza numerica: 2/3 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

- Viene usato uno strumento automatico di Cheating Detection per il confronto di tutto ciò che viene consegnato.
- In caso di due o più progetti siano ritenuti troppo simili, ad insindacabile giudizio dei docenti, i progetti saranno annullati ad entrambi i gruppi e sarà data una nuova traccia, più estesa e complessa di quella originaria

# Materiali di studio (Mod. A)

- Libri di testo generali consigliati
- Parti generali
  - I. Sommerville. Software Engineering, Pearson.
- Progettazione ad oggetti
  - C. Larman, Applicare UML e i Pattern - Analisi e Progettazione orientata agli Oggetti, V ed. Prentice-Hall, 2020.
  - B. Bruegge, A. Dutoit. Object-Oriented Software Engineering, Pearson, 2008. (Alternativo a C. Larman).

# Materiali di studio (2)

- 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
  - E. Gamma, R. Helm, R.Johnson, J.Vissides. Design patterns, Addison Wesley

# Materiali di studio (3)

- Interaction Design: Oltre l'interazione uomo-macchina, di Helen Sharp, Yvonne Rogers, Jennifer Preece. 2022 Apogeo Edizioni
- 
- J. Lazar, J. . Feng, H. Hochheiser. Research Methods in Human-Computer Interaction. Wiley 2010
- A. Cooper, R. Reimann, D. Cronin. About Face: The Essentials of Interaction Design. Wiley 2017
- Materiali vari indicati di volta in volta al corso.

# Materiali di studio (4)

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

# Contatti

- Ufficio: Palazzina 1, Il piano, via Claudio 21
- Ricevimento Di Martino:
  - Durante il I semestre, Mercoledì 10:00 – 12:00, in Presenza/Teams.
  - Dalla fine del semestre, indicazioni sul sito del docente.
- Sito del docente:  
[www.docenti.unina.it/sergio.dimartino](http://www.docenti.unina.it/sergio.dimartino)