

---

# Ingegneria del Software

III anno Laurea Triennale in Informatica



How the customer explained it



How the Project Leader understood it



How the Analyst designed it



How the Programmer wrote it



How the Business Consultant described it



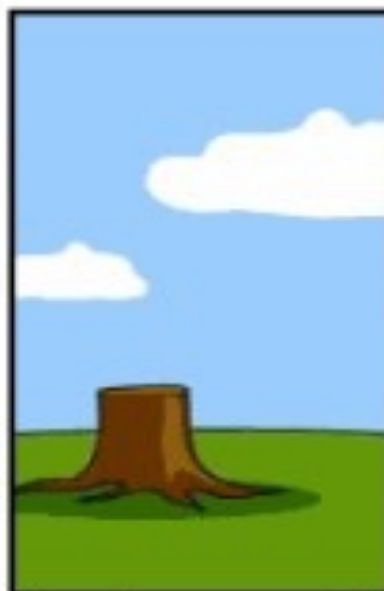
How the project was documented



What operations installed



How the customer was billed



How it was supported



What the customer really needed

# Organizzazione del corso (1)

---

## Corso Semestrale - 9 Crediti

- Concetti basilari di Ing. Del Software
- Ing. del Software Orientata agli Oggetti
- Esercitazioni
- Laboratorio
- ***Prove in itinere... new!***

# Organizzazione del corso (2)

---

- **Parte I**
  - Concetti di base
  - Design Patterns
  - Assignments e Presentazioni
- **Parte II**
  - Analisi dei Requisiti
  - IS Orientata agli Oggetti

# Concetti di Base

---

- Introduzione all'Ingegneria del Software
- Unified Modeling Language
- I Processi Software
- Analisi dei Requisiti
- Progettazione
- Sviluppo
- Verifica e Convalida (Testing & Validation)
- Gestione

# IS Orientata agli Oggetti

---

- Sviluppo Agile
  - Rational Unified Process
- Utilizzo di UML per la progettazione
- Design Patterns

# Esercitazioni

---

- Esercitazioni “classiche”
  - Svolgimento analisi dei requisiti
  - Applicazione dei design patterns
  - Il docente mostra esempi e svolgimenti esercizi
- Esercizi da svolgere e presentare in aula
  - A cura degli studenti....
  - A valore esonerante sull'orale
  - Impatto sulla firma di frequenza

# Presentazioni degli studenti

---

- Esercizi sui pattern assegnati durante le lezioni
  - Almeno 2 esercizi svolti correttamente
    - Per la verifica della presenza attiva
- Presentazione degli esercizi durante le lezioni
  - Diniego = esercizio non svolto
- Consegna su git, verifica offline
  - Esonero da una delle domande dell'orale



# Laboratorio

---

Alcuni strumenti di sviluppo software ricorrenti

- *Versioning del software (GIT)*
- *Building delle applicazioni (Maven)*
- *Testing delle applicazioni (JUnit)*
- *Project Management e Issue Tracking (Trello, JIRA)*
- *Prevede prove intermedie a valore di esonero*

Portate il vostro laptop con voi per le lezioni di laboratorio!

# Prove intermedie

---

- Svolte durante le ore di esercitazione
- Almeno il 50% di presenza
- Matrice Studente X Prova intermedia
- **Esonero sull'esame orale**
- Impatto sulla firma di frequenza

# Mini progetti

---



NEW!!

- Mini Progetto individuale
- Assegnati su richiesta le lezioni
- Visione/correzione (circa) ogni Venerdì
- Da consegnare in formato digitale (git/web)
- **Valore di esonero sul progetto finale**
- Consegna entro la data di assegnamento dei progetti (~Dicembre)

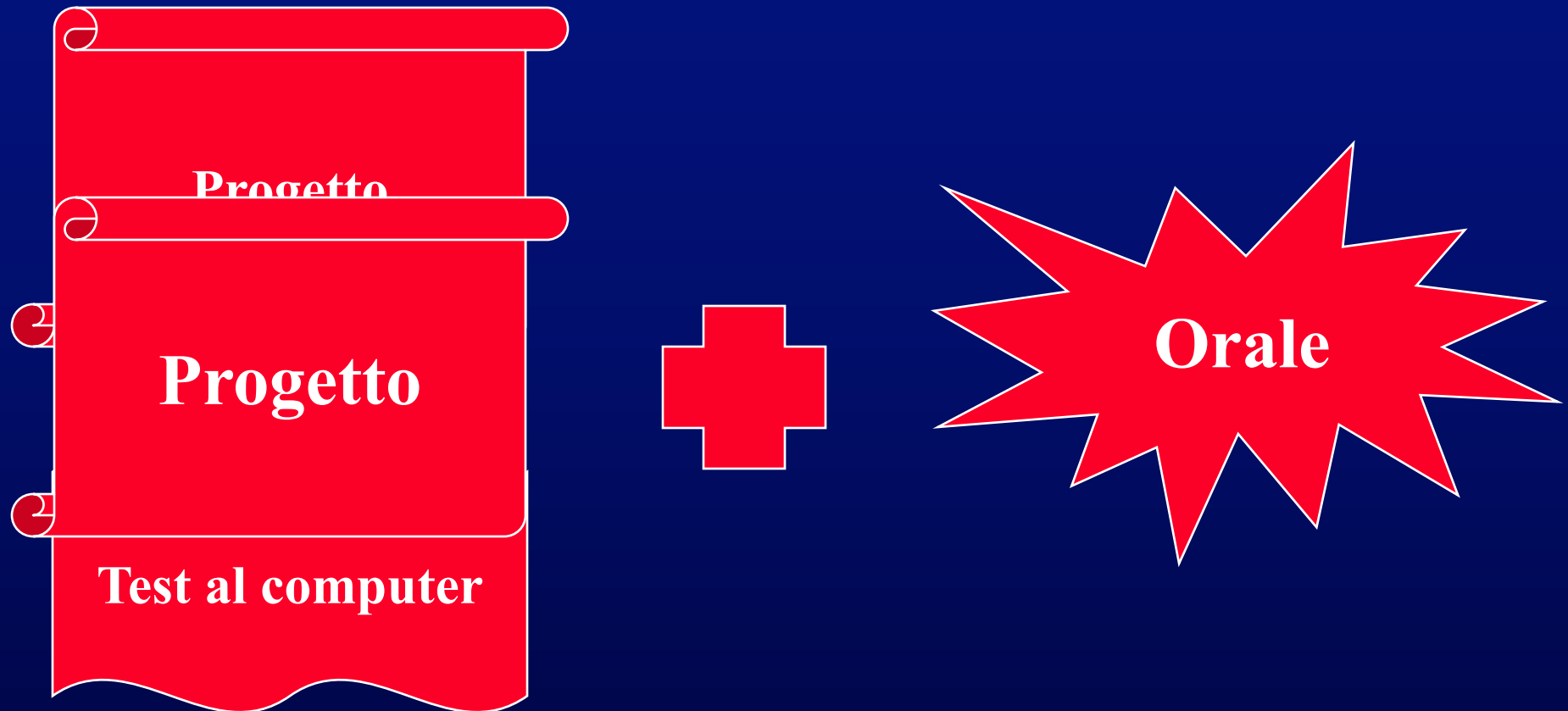
# Progetto Finale

---

- Progetto e realizzazione di un piccolo software
  - Analisi dei requisiti
  - Progettazione
  - Implementazione
- Nessun limite alla tecnologia da usare
- Tipicamente proposto dagli studenti (concordato)
  - Gruppi da 4 (esattamente 4 + resto)
- Consegna relazione primi di Gennaio
- Consegna software all'esame

# Come funziona l'esame?

---



# Come funziona l'esame?

---

- **Modalità** (progetto + orale)
  - Progetto (fino al 80%) + Orale
  - **Consentito solo al I appello**
  - **Orale o Esercizi con Presentazione**
- **Modalità** (test + orale)
  - Test (60%) + Orale (40%)
- **Massimo dei voti raggiungibile in entrambe le modalità**

# Funzionamento esoneri

---

- Esempi:
  1. Prova intermedia “Basics” corretta
    1. Nessuna domande su “basics” all’orale
  2. 2 esercizi corretti + 1 presentazione durante il Corso
    1. 1 domanda in meno (es. sui pattern) all’orale
  3. Mini Progetto su analisi dei requisiti
    1. No elaborato casi d’uso al Progetto
  4. Mini Progetto pattern
    1. No pattern nel Progetto
- Cumulabili....

# Libri e Materiale

---

- **Libri di riferimento:**
  - C. Larman - "Applying UML and Patterns..." - Prentice Hall
  - I. Sommerville - "Software Engineering" 8th ed. - Pearson Education
  - M. Fowler, K. Scott - "UML Distilled" - Addison-Wesley.
- **...e per approfondire:**
  - E. Gamma, R. Helm, R. Johnson, J. Vlissides, - "Design Patterns" - Addison-Wesley.
  - M. Fowler, K. Beck, J. Brant, W. Opdyke, Don Roberts – "Refactoring: Improving the Design of Existing Code" - Addison-Wesley
  - Brown, William J.; Raphael C. Malveau, H.W. McCormick, S. W. Thomas, T. Hudson - "Anti-Patterns in Project Management." John Wiley & Sons
  - J. Kerievsky - "Refactoring to Patterns" - Addison-Wesley
- **Dispense e lucidi sul team del corso**
  - Team code: ocu99vx



# Accedi subito

---



# Ricevimento Studenti

---

Cubo 30b - V Piano

Aula A (dopo lezione)

**Venerdì 17:30 – 18:30**

---

# COMINCIAMO?!

Introduzione all'Ingegneria del Software