



Corso di Laurea in Informatica e Tecnologie per la Produzione del Software (*Track B*) - A.A. 2018/2019

Laboratorio di Informatica

Presentazione Casi di Studio

docente: Veronica Rossano

veronica.rossano@uniba.it

Caso di Studio

Progetto da sviluppare in totale **autonomia** a partire da una traccia contenente delle specifiche

- **Specifiche** = funzionalità che il sistema dovrà implementare, vincoli che dovrà rispettare
- Il caso di studio è da sviluppare **in coppia**
 - E' possibile sviluppare il caso di studio anche singolarmente, ma le modalità di valutazione restano invariate. E' dunque consigliata la formazione di gruppi

Caso di Studio

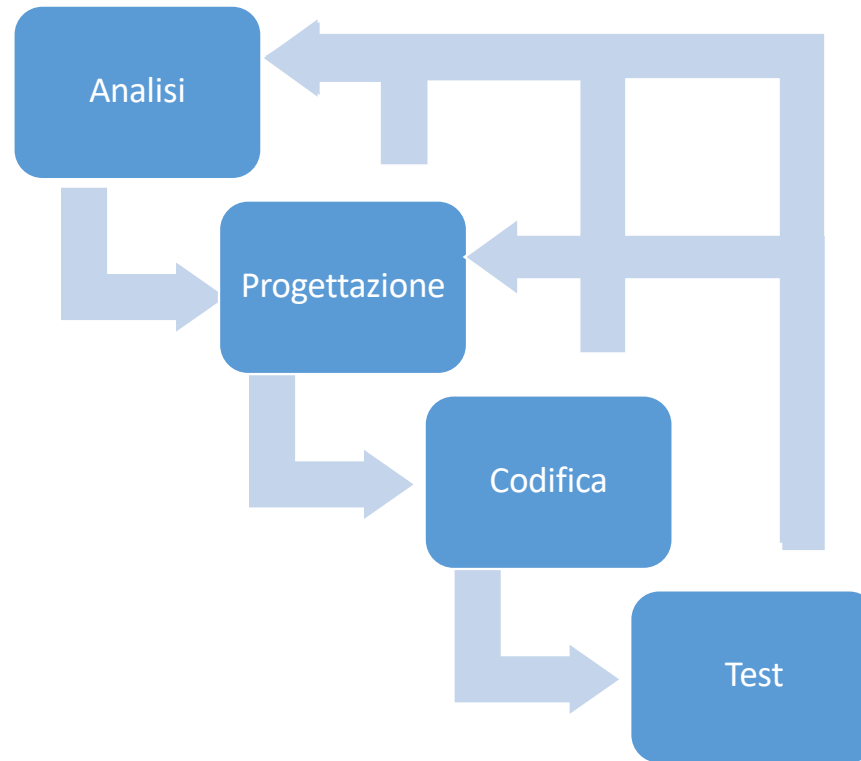
- **Traccia d'esame**
 - Una traccia per ogni appello
 - **Sessione 1: appelli Giugno, Luglio 2019**
 - Tracce comunicate ad Aprile 2019
 - **Sessione 2: appelli Settembre, Novembre 2019**
 - Tracce comunicate a Luglio 2019
 - **Sessione 3: appelli Febbraio, Aprile 2020**
 - Tracce comunicate a Novembre 2019

Caso di Studio

Scelte della traccia e composizione dei gruppi **non possono essere cambiate, a meno di motivate (e giustificate) richieste**

Caso di Studio – Linee Guida

Realizzare il caso di studio
Seguendo i quattro passi
Standard del **Ciclo di Vita
del Software**



Caso di Studio – Linee Guida

- **Analisi**

- **Chiarifica del problema**

- Qual è il problema che vogliamo affrontare?

- **Specifica dei Requisiti**

- Quali sono le funzionalità da implementare?

- **Progettazione**

- Individuazione di una strategia di soluzione

- COME raggiungere l'obiettivo?

- Scelta delle strutture dati

- Scelta degli algoritmi più adeguati

Caso di Studio – Linee Guida

- **Codifica**

- Scrittura del programma

- **Verifica (e correzione)**

- (Test) del programma
 - Rimanda ad una delle fasi precedenti

- **Manutenzione**

- Correttiva
 - Adattativa
 - Migliorativa

Caso di Studio – Linee Guida

1) Analisi

- **Chiarifica in linguaggio naturale del problema**
 - Chiarisce il problema (chi userà il sistema? Come lo userà? Per quanto tempo?) e il dominio applicativo in cui si lavora
- **Definisce le funzionalità (requisiti funzionali)**
 - Specifica le condizioni che devono verificarsi per poter eseguire le operazioni
 - Specifica dati di input (valori accettati/non accettati) e dati di output (valori attesi)
 - Definisce eventuali assunzioni ed ipotesi semplificative
- **Definisce i vincoli (requisiti non funzionali)**
 - Caratteristiche che il programma deve garantire (in merito ad esempio all'efficienza o alla semplicità d'uso)
- **Produce un documenti di Analisi dei Requisiti**

Caso di Studio – Linee Guida

2) Progettazione

- **Dal Problema alla Soluzione**
 - Progettazione Top-Down / Bottom-Up
 - Utilizzo di formalismi standard (pseudo-linguaggio/flow-chart)
 - Esame di Programmazione 😊
- **Scelta delle Strutture Dati**
 - Definizione dei **tipi di dato** necessari a risolvere il problema (tipicamente delle **struct**) e dei relativi campi, valutazione Pro/Contro

Caso di Studio – Linee Guida

2) Progettazione (cont.)

- **Progettazione Modulare**
 - Quali funzioni? Quali procedure? Quali parametri?
Concetti: Astrazione Dati e Information Hiding
- **Scelta degli Algoritmi**
 - Ordinamento, Ricerca, etc.
- **Produce un documento di Progettazione**

Caso di Studio – Linee Guida

3) Codifica

- **Utilizzo Corretto del Linguaggio C**
 - Conoscenza e utilizzo delle librerie (es. `<string.h>`)
 - Utilizzo corretto di procedure e funzioni
 - Utilizzo adeguato di file e puntatori, quando necessario
- **Programmazione «difensiva»**
 - Corretta gestione degli errori
 - Corretta gestione dei casi «limite»

Caso di Studio – Linee Guida

3) Codifica (cont.)

- **Corretto Stile di Programmazione**
 - Indentazione Adeguata
 - Identificatori significativi
 - Commenti opportuni
- **Documentazione con Tool Standard (es. Doxygen)**
- **Produce il file eseguibile del progetto**

Caso di Studio – Linee Guida

4) Testing

- **Metodo Empirico**

- Ispezione del codice per individuare errori
- Tracing (simula l'esecuzione passo dopo passo)
- Metodo delle stampe (stampare a video nei punti critici)

- **Metodo Analitico**

- Correttezza semantica mediante asserzioni logiche (CUnit)
- Test a scatola nera dei valori di input e output
 - Casi tipici (con valori attesi)
 - Casi limite (con valori particolari)

- **Produce un piano di test (e la relativa esecuzione)**

Caso di Studio – La Scelta

Una volta selezionato lo scenario del caso di studio e formato (eventualmente il gruppo, procedere seguendo questo protocollo:

1. Rispondere al sondaggio sulla piattaforma di e-learning

- Ogni componente deve farlo

2. Inviare una mail all'indirizzo veronica.rossano@uniba.it con oggetto:

- [LabInf1819] Scelta Caso di Studio

2. Nel corpo della mail indicare:

- Nome, Cognome, Matricola, Anno di Corso (se diverso dal primo) di entrambi i membri del gruppo e **indicazione della traccia scelta.**

Caso di Studio – La Scelta

3. In risposta alla mail riceverete:

- Una conferma di avvenuta prenotazione, e un identificativo del gruppo da utilizzare per tutte le comunicazioni successive.
 - Es) [LabInf1819-Gruppo12] Richiesta Informazioni
- **In caso di mancata risposta entro 48h, scrivere nuovamente.**
- Comunicazioni con oggetto errato saranno ignorate

Caso di Studio – La Valutazione

A ogni caso di studio sarà assegnato **un punteggio da 0 a 33 punti**, distribuiti come segue.

Metrica di Valutazione	Rosso	Giallo	Verde
Correttezza	0-2 punti	3-4 punti	5-7 punti
Programmazione Modulare	0-2 punti	3-4 punti	5-7 punti
Programmazione Difensiva	0-2 punti	3 punti	4 punti
Stile di Programmazione	0-2 punti	3 punti	4 punti
Documentazione e Commenti	0-2 punti	3 punti	4 punti
Testing del Codice	0-2 punti	3 punti	4 punti
Bonus	3 punti (max.)		
TOTALE	33 punti		

Caso di Studio – La Valutazione

- **Correttezza**

- **Il programma implementa correttamente tutte le funzioni indicate nelle specifiche.** La correttezza riguarda anche l'utilizzo adeguato di puntatori e file, quando necessario. Anche la corretta progettazione delle strutture e dei tipi di dato più adeguati atte a risolvere il problema rientra in questa voce. Sarà valutata anche la scelta degli algoritmi di ricerca e ordinamento implementati. **La correttezza del programma riguarda anche la quantità di dati inseriti nei file di esempio.**

- **Programmazione Modulare**

- **Il programma è implementato seguendo correttamente i principi della programmazione modulare.** Procedure e funzioni sono progettate correttamente, sia in termini di parametri di input che in termini di tipi di ritorno. E' corretto implementare il progetto sia come libreria statica che come file .h /.c separati.

- **Programmazione Difensiva**

- **Il programma implementa i controlli di correttezza sugli input / output del programma (tipo di input corretto, date, nomi, etc.)**

Caso di Studio – La Valutazione

- **Stile di Programmazione**

- Il codice è scritto seguendo le linee guida corrette in termini di nomenclatura delle variabili e delle funzioni, scelta delle espressioni, indentazione, spaziatura e leggibilità.

- **Testing**

- I test sono progettati nel modo corretto e coprono tutti i casi (es. casi limite, situazioni di errore, etc.) che è necessario valutare. **L'esecuzione dei test corrisponde a quanto indicato nel piano di test.**

- **Documentazione e Commenti**

- Il progetto è correttamente documentato e utilizza le librerie per la generazione automatica della documentazione dei metodi implementati. **Ciascun elemento del progetto è documentato in modo preciso e completo seguendo le linee guida mostrate a lezione. I commenti sono inseriti in modo significativo**

Caso di Studio – La Valutazione

- **Bonus (max. 3 punti)**

- Invio delle esercitazioni durante il corso
- Partecipazione alle attività online
- Implementazione estensioni alle funzionalità del progetto

Caso di Studio – La Consegna

- **Consegna:** 10 giorni prima della data d'appello
 - Es: Primo Appello = 11 Giugno 2019 → Consegna = 1 Giugno

Caso di Studio – La Consegna

- **Consegna: max. 10 giorni prima della data d'appello**
 - Es: Primo Appello = 11 Giugno 2019 → Consegna = 1 Giugno
- **Come?**
 - Inviare i file utilizzando GDrive
 - Link: <http://tiny.cc/vixu5y>
 - Creare un archivio zip che deve utilizzare la seguente nomenclatura:
<matricola>_<cognome>_<appello>_<versione>.zip
 - **Es) 432941_musto_luglio2017_v1.zip**
 - In caso di consegne di gruppo, **sostituire il numero del gruppo a <cognome> e usare la matricola di chi consegna.**

Caso di Studio – La Consegna

- Cosa metto nel file .zip?

1. Documentazione del Caso di Studio

- Documento di Analisi + Documento di Progetto + Piano di Test e Relativa Esecuzione

2. Progetto Eclipse con i Sorgenti (verificare che l'import funzioni!)

3. Eseguitibile funzionante (verificare che l'eseguitibile funzioni!)

4. Compilazione ed esecuzione devono essere riproducibili sul computer del docente, pena la non ammissione all'orale.

- IMPORTANTE: fare attenzione al nome del file .zip

- Creare un archivio zip che deve utilizzare la seguente nomenclatura:

<matricola>_<cognome>_<appello>_<versione>.zip oppure

<matricola>_<gruppo>_<appello>_<versione>.zip

Es) 432941_musto_luglio2017_v1.zip

Caso di Studio – Prova d'esame

- **Come è organizzato l'esame? L'esame** consiste in:

1. una discussione del caso di studio
2. una prova orale che **prevede delle domande di teoria relative ai contenuti del corso** (programmazione modulare, testing, debugging, stile di programmazione, etc.)

Caso di Studio – Prova d'esame

- **Cosa si intende per discussione?**

- **Fase 1: Demo**

- Lo studente o gli studenti mostrano le funzionalità implementate. Evidenziano i punti di forza e i punti di debolezza del programma. Si mostra se il programma risponde (o meno) ai requisiti funzionali e non funzionali.

- **Fase 2: Documentazione**

- **Si entra nel merito di quanto implementato.** E' importante mostrare consapevolezza delle funzionalità implementate. Bisogna saper giustificare le scelte progettuali effettuate e conoscere gli algoritmi.

- **Fase 3: Codice Sorgente**

- **Eventualmente, si guarda insieme il codice sorgente** in caso di dubbi relativi alla 'paternità' del codice

