



UNIVERSITÀ  
DEGLI STUDI  
DI BERGAMO

Dipartimento  
di Ingegneria Gestionale,  
dell'Informazione e della Produzione



# Linee guida per lo sviluppo del progetto

PROGETTAZIONE, ALGORITMI E  
COMPUTABILITÀ  
(38090-MOD1)

**Corso di laurea  
Magistrale in  
Ingegneria  
Informatica**

RELATORE

Prof.ssa Patrizia  
Scandurra

SEDE

DIGIP

DATA

04-10-2021

# Indicazioni dei passi di sviluppo software e di contenuti del progetto

1. ~~Formare un team di sviluppo e concordare un'idea di progetto insieme al docente~~
2. Seguire il processo di sviluppo agile AMDD
3. Preparare la documentazione da consegnare al docente
4. Consegnare il progetto
  - rilasciare documentazione, codice sorgente, codice eseguibile (archivio Jar, script, e vari artifatti SW) su un repository (Dropbox, Github, ecc..) da condividere con il docente al momento della consegna.

Per i dettagli sulla valutazione del progetto, vedi la presentazione "introduzione.pdf" della prima lezione del corso.

# 1. Formare un team di sviluppo e concordare un'idea di progetto insieme al docente

Stabilire le seguenti cose:

- ~~Idea e raccolta iniziale dei primi requisiti software.~~
- ~~Stabilire una "tool chain" per lo sviluppo agile AMDD del progetto, ovvero decidere quali tool usare per le attività di design, sviluppo collaborativo, analisi statica/dinamica del codice, distribuzione e versionamento.~~
- ~~Stabilire un'agenda e un'organizzazione del team: cadenzare le sessioni/riunioni di lavoro in base alle iterazioni e fasi del processo AMDD e alle best practice agili che si intendono adottare. L'agenda e la suddivisione del gruppo può evolvere nel tempo in base alle esigenze di progetto e competenze di ciascuno.~~ A ogni iterazione di AMDD si fa un brainstorming
- Studiare e prendere come riferimento il "progetto guida" COCOME per modellare in UML e sviluppare il codice delle componenti in modo consistente con il design architetturale in UML. Per quanto riguarda l'implementazione, prendere come riferimento i progetti dell'anno scorso pubblicati su Moodle

## 2. Seguire il processo di sviluppo agile AMDD

- ~~Creare una sottocartella diversa per ogni iterazione (release) del progetto.~~

~~Le iterazioni vanno numerate!~~

A ogni iterazione si clona la cartella dell'iterazione precedente e si riparte da lì.

CHIEDERE RICEVIMENTO SU ITERAZIONE 0 E 1: SE SI SBAGLIANO QUELLE, IL PROGETTO VA SU UNA CATTIVA STRADA

- ~~Iterazione 0 (envisioning): analisi dei requisiti, modellazione dei casi d'uso (requisiti funzionali) iniziali e della configurazione iniziale dell'architettura del sistema a runtime ("topologia" del sistema: free style diagram o UML deployment diagram/component diagram).~~

- Ad ogni iterazione successiva:

- i. Planning: brain storming con scelta e affinamento dei casi d'uso da realizzare come incremento software da produrre nell'iterazione corrente;
- ii. design: modellare/raffinare in modo agile producendo un "modello di analisi" (opzionale) e un "modello di design" dell'architettura SW con componenti e design pattern da realizzare/raffinare per implementare i casi d'uso scelti e mitigare eventuali problemi emersi dall'analisi statica/dinamica;
- iii. Implementazione, analisi statica e dinamica (testing di unità con copertura) del codice.

## 2. Preparare la documentazione da consegnare al docente

L'ITERAZIONE 1

Documentare 3 iterazioni: l'iterazione 0, un'iterazione in mezzo e quella finale. La documentazione di un'iterazione intermedia deve contenere:

- Documento di analisi requisiti (essenzialmente documento raccolta dei requisiti funzionali e non funzionali, user story/casi d'uso UML e descrizione dei passi per i requisiti funzionali)
- Documento di design dell'architettura SW: i. fornire un modello UML 4+1 viste. In particolare, per la vista strutturale: usare UML deployment/component diagram e class diagram per il design gerarchico e i dettagli delle componenti, dei tipi di dato e delle interfacce (le API!).
- Documento "design in piccolo" degli algoritmi usati (pseudocodice + analisi complessità tempo)
- Documento analisi dinamica: casi di test e copertura con Junit/Eclemma
- Documento di analisi statica: report STAN4J
- Realizzare alla fine anche una succinta guida con screenshot sull'installazione ed uso dell'applicazione.

Bisogna presentarsi allo scritto col progetto pronto: gli altri membri del gruppo, se si passa l'orale, presentano il progetto anche se non hanno ancora passato lo scritto.