

# *SmartMag*

*Progetto di ingegneria del software*

*A.A. 2023/2024*

*Brambilla Davide - Matr. 1080752*

*Brivio Lorenzo - Matr. 1073423*

*Gervasoni Massimiliano - Matr. 1069211*



UNIVERSITÀ  
DEGLI STUDI  
DI BERGAMO

# Project Plan:

## 1. Introduzione:

Lo scopo di questo progetto è quello di mettere in pratica le nozioni apprese durante il corso di ingegneria del software. In particolare, riguardanti l'analisi, la progettazione, la realizzazione, la manutenzione e il test di un sistema informatico. Il nostro progetto consiste nella realizzazione di un software di gestione di un magazzino, che permette di svolgere una serie di attività, tra cui:

- Tenere traccia dei depositi e dei prelievi degli articoli, con le relative destinazioni;
- Monitorare le quantità disponibili, con la possibilità di impostare una soglia minima di scorta a magazzino;
- Tenere traccia delle caratteristiche di ogni singolo prodotto.

Inoltre, è prevista la possibilità di inserimento di utenti con ruoli distinti, ognuno dei quali avrà accesso a funzionalità diverse messe a disposizione del software.

## 2. Process Model:

Per la realizzazione del progetto è stato scelto di utilizzare RAD (Rapid Application Development) come modello di processo.

Questa scelta è stata motivata dalla volontà di avere una maggiore flessibilità e rapidità nello sviluppo del software. Grazie al RAD, infatti, abbiamo la possibilità di apportare rapidamente un gran numero di modifiche al software, senza dover ogni volta ripartire dalle fasi iniziali di sviluppo.

## 3. Organizzazione:

Il progetto verrà svolto da una squadra AGILE. Abbiamo scelto questo tipo di organizzazione perché ha molto in comune con la squadra SWAT (Skilled With Advanced Tools), che viene utilizzata quando si sceglie un modello di processo RAD. Tuttavia, a differenza della squadra SWAT, nella squadra AGILE non è presente una gerarchia e questo è un altro motivo per cui l'abbiamo scelta.

Come in una squadra SWAT, utilizzeremo degli strumenti di supporto per lo sviluppo del software.

## 4. Standard, linee guida, procedure:

- Condivisione codice e documenti: Il codice e la documentazione saranno sempre accessibili ai membri del gruppo e ai professori su un repository GitHub.
- Stesura codice: Lo "scheletro" del codice verrà realizzato mediante il tool REBEL di StarUML, mentre la restante parte, verrà scritta manualmente utilizzando l'IDE Eclipse in linguaggio Java.

In particolare, il codice verrà scritto seguendo le Java Code Conventions: una serie di convenzioni per produrre codice caratterizzato da alta leggibilità, in modo da favorirne la manutenibilità in futuro.



- Modellazione: La modellazione verrà realizzata utilizzando UML (Unified Modeling Language) mediante il software StarUML. Nello specifico, verranno realizzati i seguenti diagrammi:
  - Diagramma delle classi
  - Diagramma dei casi d'uso
  - Diagramma degli stati
  - Diagramma di sequenza
  - Diagramma delle attività
- Qualità del software: Per misurare la qualità del software verranno utilizzati dei tool come ad esempio: Stan4j, Jdepend, Structure 101.
- Consegna: Il project plan deve essere pronto un mese prima dell'esame, nel nostro caso, essendo la data d'esame il 16/02/2024, dovrà essere consegnato entro il 16/01/2024. Mentre il progetto completo dovrà essere completato cinque giorni prima dell'esame orale.

## 5. Gestione delle attività:

Durante tutto il periodo di sviluppo utilizzeremo Slack come piattaforma di comunicazione, dove verranno realizzati degli incontri in cui saranno assegnate le attività da svolgere a tutti i membri del gruppo.

Avendo scelto come modello di processo RAD, i lavori dovranno essere eseguiti entro una certa time-box, il cui tempo potrà variare in base alla complessità dell'attività da svolgere. Lo stato delle attività verrà aggiornato su una Kanban Board in modo tale da permettere ad ogni membro del gruppo di monitorare lo stato di completamento delle varie attività.

## 6. Rischi:

Il rischio principale è quello di non riuscire a rispettare i tempi prestabiliti. Ciò può essere dovuto a due fattori principali:

- Mancanza di disponibilità dei membri del gruppo;
- Inesperienza nell'utilizzo dei vari tool di sviluppo e controllo del codice.

La prima è dovuta alla corrispondenza temporale tra consegna del progetto e sessione d'esame che potrebbe portare a dei ritardi di consegna. Una possibile soluzione è quella di tenere conto dei possibili ritardi nelle time-box in modo tale verificare la fattibilità del progetto.

Mentre per quanto riguarda la seconda, si è deciso di adottare la tecnica MoSCoW per poter dare una priorità ai requisiti del progetto.

## 7. Personale:

Il gruppo di sviluppo è composto da Brambilla Davide, Brivio Lorenzo e Gervasoni Massimiliano. Nel team non si individuano dei ruoli precisi e i membri sono coinvolti in ugual misura durante le fasi di sviluppo del progetto.



## 8. Metodi e tecniche:

Nella fase di ingegneria dei requisiti, verranno utilizzate le seguenti tecniche:

- Intervista
- analisi delle attività
- analisi degli scenari

Queste tecniche saranno impiegate per l'elicitazione dei requisiti a cui successivamente, verrà assegnata una priorità utilizzando il metodo MoSCoW.

Per le fasi di modellazione, architettura del software e design, verrà utilizzato il software StarUML con il quale realizzeremo i diagrammi in UML.

Per il controllo della qualità del software, saranno utilizzati strumenti software come jdepend.

Per quanto riguarda le attività di test, principalmente verranno effettuate manualmente tramite simulazioni di utilizzo del programma. Inoltre, verrà utilizzato JUnit per testare le sezioni più importanti del codice in modo automatizzato.

## 9. Garanzia di qualità:

Poiché il nostro progetto non fa parte della categoria dei software critici, non è necessario applicare standard rigidi come l'IEEE 730. Pertanto, con l'aiuto dei tool già menzionati, verranno valutati i criteri di McCall tra cui:

- Manutenibilità
- Flessibilità
- Testabilità
- Riutilizzabilità

Inoltre, la portabilità verrà garantita dal linguaggio Java stesso (bytecode e JVM).

## 10. Work Packages:

Utilizzando come modello di processo RAD, il progetto verrà suddiviso in attività alle quali corrisponde una time-box.

Time-box	Attività
1	Organizzazione e scelta del progetto da realizzare
2	Project plan
3	Specifica dei requisiti
4	Architettura del software
5	Design del software
6	Realizzazione dei diagrammi UML
7	Scrittura del codice
8	Test

## 11. Risorse:

Ad ogni fase del progetto parteciperanno tutti i membri del team utilizzando le seguenti risorse:

- Eclipse, GitHub, REBEL, StarUML, etc.
- Materiale didattico fornito durante il corso di ingegneria del software in particolare il libro Software Engineering di H. V. Vliet e gli esercizi svolti a lezione.
- Documentazione online dei vari tool/librerie

## 12. Budget:

Il budget di tempo a disposizione per la realizzazione del progetto è di circa 40 ore per ogni membro del gruppo, per un totale di 120 ore.

Queste ore sono state assegnate a ciascuna time-box nel seguente modo:

Nr. ore	Attività
2	Organizzazione e scelta del progetto da realizzare
10	Project plan
8	Specifica dei requisiti
14	Architettura del software
14	Design del software
12	Realizzazione dei diagrammi UML
50	Scrittura del codice
10	Test
120	TOTALE ORE

## 13. Modifiche:

L'organizzazione della squadra è di tipo AGILE, pertanto le modifiche potrebbero non essere viste come cambiamenti significativi. Nonostante ciò, il nostro gruppo terrà traccia di ogni modifica che verrà apportata al codice grazie all'utilizzo di GitHub, dove ad ogni aggiornamento rilevante del programma, corrisponderà un'azione di commit con il relativo commento.

Di GitHub verranno anche utilizzati gli Issues che permetteranno di tenere traccia dei problemi che si saranno riscontrati durante lo sviluppo del software e delle modifiche messe in atto per risolverli.

## 14. Consegna:

La consegna del progetto avverrà attraverso un repository GitHub entro i limiti di tempo specificati. Nel repository sarà presente il software e tutta la documentazione ad esso relativa.