

# UniBg Notes

Gestione del Progetto

Progetto di Ingegneria del software

*Bilal Drissi 1087161*

*Daniele Iania 1085982*

*Mattia Nicastro 1086903*

*a.a 2024/2025*



# INDICE

Introduzione	1
Software Life Cycle	2
Configuration Management	3
Team Management	4

## 1. Introduzione

Questo documento specifica e approfondisce come verrà gestito il progetto software *UniBg Notes* con lo scopo di far capire il modo in cui il team di sviluppo procederà.

## 2. Software Life Cycle

Il *team* adotterà il framework **SCRUM**, un modello di sviluppo agile progettato per facilitare la gestione e lo sviluppo di prodotti complessi. Questo approccio si basa su iterazioni incrementali, volte a fornire valore in modo rapido ed efficiente.

Il processo inizia con la selezione di un insieme di attività dal **Product Backlog**, che vengono inserite nello **Sprint Backlog** e rappresentano gli obiettivi dello Sprint successivo. Durante lo Sprint, il team partecipa al **Daily Scrum**, una breve riunione giornaliera finalizzata a monitorare i progressi, identificare eventuali ostacoli e allineare le attività dei membri.

Alla conclusione di ogni Sprint, si tiene una **Sprint Retrospective**, durante la quale il *team* analizza i risultati ottenuti confrontandoli con gli obiettivi stabiliti nello Sprint Backlog, identificando aree di miglioramento. Questo ciclo iterativo si ripete fino a completare tutte le attività presenti nel Product Backlog, garantendo la consegna incrementale di un prodotto di valore.

### 3. Configuration Management

Nel corso dello sviluppo del progetto, l'organizzazione del lavoro si baserà su strumenti di versionamento e collaborazione, con **GitHub** e **GitHub Desktop** come piattaforme principali, e per quanto riguarda la documentazione verranno utilizzate le piattaforme di **Google Drive** e **Google Docs** per poter collaborare alla stesura del Project Plan, Specifica dei Requisiti e Gestione del progetto.

Questi strumenti permetteranno ai membri del team di operare in modo indipendente attraverso l'eventuale uso di branch separati, garantendo al contempo la stabilità del branch principale. Ogni attività sarà gestita tramite **issue** dettagliate, la cui assegnazione e avanzamento verranno monitorati tramite **Kanban Board** aggiornata in tempo reale. Questa metodologia offrirà una visione chiara del progresso, agevolando l'integrazione continua delle modifiche ed una gestione efficiente delle risorse e della documentazione.

Il **Repository** del progetto sarà strutturato in modo da facilitare l'accesso e l'organizzazione dei diversi componenti. Le principali sezioni includeranno:

- **Codice**: conterrà tutti i file sorgente del sistema sviluppo;
- **Documentazione**: raccoglierà i materiali relativi alla pianificazione, ai requisiti, al design, ai test e alla manutenzione del progetto;
- **Diagrammi UML**: includerà modelli come il diagramma delle classi, delle sequenze e delle attività, fondamentali per rappresentare visivamente le funzionalità e la struttura del software;
- **Images**: conterrà immagini (come il logo del software) da poter usare in fase di documentazione e design;
- **Test**: ospiterà script e report relativi ai test effettivi per verificare il corretto funzionamento del sistema.

Questa suddivisione consentirà una gestione chiara dei contenuti e faciliterà il lavoro collaborativo. Le modifiche al codice saranno sottoposte a revisione attraverso il flusso di **Pull Request**, con l'obiettivo di mantenere un'elevata qualità del prodotto finale.

### 4. Team Management

Il *team*, composto da tre membri, seguirà un approccio basato sulla metodologia Agile **SCRUM**, che permette una pianificazione flessibile ed un adattamento continuo alle esigenze del progetto. Il lavoro è suddiviso, come specificato nel **Capitolo 3. Configuration Management**, in sprint, ognuno con una durata predefinita, durante i quali vengono completate attività specifiche secondo le priorità stabilite nel **Product Backlog**.

Ognuno dei membri assumerà contemporaneamente i ruoli di **Product Owner**, **Scrum Master** e **Development Team**. Questa distribuzione permette a tutti i membri

di contribuire attivamente allo sviluppo, alla supervisione delle attività e al mantenimento del focus sugli obiettivi, discutendo eventuali difficoltà durante i **Daily Scrum** ed effettuando **Sprint Review** e **Retrospective** per individuare miglioramenti e ottimizzare il processo di lavoro.

Strumenti come **GitHub**, **Kanban Board**, come spiegato nel **Capitolo 3. Configuration Management**, e piattaforme di comunicazione favoriranno la collaborazione e la trasparenza, assicurando che ogni membro sia pienamente coinvolto in tutte le fasi del progetto.