

# **PROJECT PLAN**

**PROGETTO INGEGNERIA DEL SOFTWARE**

**REALISED BY**

**MATTEO NICOLI**

**13/12/2024**

1.0.0

# *Indice*

1. Introduzione
  - 1.1. Background e storia del progetto
  - 1.2. Obbiettivi del progetto
  - 1.3. Risultati attesi
  - 1.4. Management
  - 1.5. Sintesi del progetto
2. Modello di processo
  - 2.1. Modello utilizzato
  - 2.2. Struttura del processo Scrum
  - 2.3. Milestones
3. Organizzazione del progetto
  - 3.1. Organizzazioni coinvolte
4. Standard, linee guida, procedure
  - 4.1. Standard tecnici e linee guida
  - 4.2. Piano di controllo della configurazione
5. Attività di gestione
  - 5.1. Documentazione degli sviluppi operativi
  - 5.2. Bilanciamento delle priorità
  - 5.3. Gestione delle tempistiche
6. Rischi
  - 6.1. Rischi relativi alla coerenza con i requisiti
  - 6.2. Rischi relativi al rilascio della piattaforma
  - 6.3. Rischi dovuti a ritardi nello sviluppo
7. Personale

7.1. Struttura del management

7.2. Struttura del team di sviluppo

8. Metodi e tecniche

8.1. Ingegneria dei Requisiti

8.2. Progettazione

8.3. Controllo della configurazione

8.4. Piano di Documentazione

9. Garanzia di qualità

9.1. Requisiti di qualità

10. Pacchetti di lavoro (workpackages)

10.1. Descrizione dei pacchetti di lavoro

11. Risorse

11.1. Personale

11.2. Risorse software

11.3. Risorse hardware

12. Budget

13. Cambiamenti

14. Consegna

# 1. *Introduzione*

## 1.1 *Background e storia del progetto*

Negli ultimi anni, c'è stata un'impennata nella domanda di trattamenti personalizzati nel campo della chirurgia estetica. Di conseguenza, molti professionisti di questo settore si trovano a lottare per far fronte a crescenti quantità di dati relativi ai pazienti, ai trattamenti e ai follow-up, spesso ricorrendo a strumenti rudimentali o non adeguati alle loro esigenze.

Questo progetto mira a colmare questa lacuna fornendo una soluzione innovativa.

Modernità e semplicità sono le parole chiave che contraddistinguono questo progetto.

## 1.2 *Obbiettivi del progetto*

L'obiettivo del progetto è realizzare una piattaforma web che permetta ai chirurghi estetici di:

- Gestire in modo sistematico i dati personali e sanitari dei pazienti.
- Visualizzare una storia dei trattamenti estetici effettuati, la loro efficacia e i risultati ottenuti, attraverso foto delle procedure, note esplicative su ciò che viene trattato e definizione dei prodotti utilizzati durante ciascuna visita.
- Fornire un monitoraggio delle visite mediche.

## 1.3 *Risultati attesi*

Il progetto ha l'obiettivo di consegnare una piattaforma con le seguenti funzionalità:

- Modulo di registrazione e accesso per medici e chirurghi alla piattaforma.
- Dashboard per la gestione dei pazienti.
- Dashboard per la gestione e la visualizzazione delle visite e dei trattamenti.
- Modulo per il caricamento delle foto relative ai trattamenti

## *1.4 Management*

- Project Manager: Matteo Nicoli
- Team di sviluppo: Matteo Nicoli – full-stack developer
- Referenti accademici: Angelo Gargantini, Silvia Bonfanti

## *1.5 Sintesi del progetto*

La piattaforma web ambisce a essere l'innovazione per i chirurghi estetici, facilitando la gestione dei dati e migliorando nel complesso tale esperienza per i professionisti del settore, digitalizzando tutti quei processi che venivano eseguiti in modo cartaceo.

# *2. Modello di processo*

## *2.1 Modello utilizzato*

Per questo progetto è stato scelto il modello di processo Agile SCRUM per lo sviluppo della piattaforma web. In quanto è previsto che il progetto necessiti flessibilità e rilasci incrementali.

## *2.2 Struttura del processo Scrum*

- Pianificazione dello sprint: Ogni sprint avrà una durata di 3 giorni, durante le quali il team dovrà completare funzionalità specificate nel Product Backlog.
- Revisione dello sprint: alla fine di ogni sprint, verrà caricato un una beta funzionante del prodotto per effettuare test e verificare la coerenza con le richieste iniziali.
- Retrospective Sprint: Il team effettuerà un resoconto sulle performance del proprio sprint e individuerà le possibilità di miglioramento.

## *2.3 Milestones*

Le milestone del progetto sono le seguenti:

1. Elicazione dei requisiti
2. Modellazione del diagramma dei casi d'uso
3. Modellazione delle classi Java per la parte di backend
4. Modellazione di altri diagrammi utili
5. Definizione dell'architettura
6. Codifica della parte relativa all'autenticazione della piattaforma
7. Realizzazione delle prime API: necessarie per la gestione dei pazienti
8. Realizzazione della dashboard pazienti
9. Realizzazione delle API relative a gestione di visite e trattamenti
10. Realizzazione della dashboard visite e trattamenti
11. Realizzazione delle funzionalità per il caricamento delle foto
12. Implementazione funzionalità per il caricamento delle foto

## 12.2 *Rischi principali*

I rischi principali che possono presentarsi nella realizzazione del progetto sono i seguenti:

- Creazione di una struttura funzionale in accordo con i requisiti richiesti:

La creazione di una struttura che soddisfi pienamente i requisiti del può risultare complessa, rischiando di portare a discrepanze tra le aspettative e il prodotto finale.

Per ovviare a questo rischio, si adotta un approccio Agile che permette piccoli cicli di sviluppo incrementali.

- Realizzazione di un'interfaccia non abbastanza user-friendly:

La piattaforma potrebbe avere un'interfaccia non abbastanza intuitiva per i medici che la utilizzeranno, i quali probabilmente non sono abituati all'utilizzo di tecnologie di questo tipo.

Questo potrebbe portare a una perdita di utenti.

Per affrontare questo rischio, verranno realizzati prototipi e test di usabilità.

- ritardi dovuti alla qualità del sistema:

per la realizzazione di un sistema mantenendo alti standard di qualità sia durante la fase di sviluppo che nei risultati è necessaria una grande mole temporale, il che può portare a ritardi rispetto alle date prefissate nel piano di progetto e nei singoli sprint SCRUM.

## 3. *Organizzazione del progetto*

### 3.1 *Organizzazioni coinvolte*

Il progetto coinvolge le seguenti entità organizzative principali:

- **Organizzazione Utente - chirurghi estetici e cliniche**

Gli stakeholder finali parteciperanno attivamente alla raccolta dei requisiti e ai test di accettazione. Forniranno:

- Requisiti funzionali specifici.
- Feedback sull'uso e sulle funzionalità implementate.

- **Organizzazione madre - team di sviluppo**

Il team di sviluppo si occuperà dello sviluppo del software, dei test e della gestione delle versioni. Si assicurerà che lo sviluppo avvenga in corrispondenza delle specifiche richieste

- **Altri enti**

Se necessario, saranno coinvolte figure esterne come i docenti universitari, per le linee guida relative alla realizzazione del progetto.

## 4. *Standard, linee guida e procedure*

## 4.1 Standard tecnici e linee guida

Per quanto riguarda la parte di codifica verranno applicati i seguenti standard:

- Backend (Spring Boot con Java):

Si seguiranno le convenzioni di codifica per Java definite in Oracle Code Conventions e in specifiche per Spring Boot. Saranno rispettate le best practices per la gestione delle dipendenze tramite Maven per la configurazione del framework.

- Frontend (React):

Per il frontend si seguiranno gli standard di sviluppo per React, la gestione dello stato tramite React Hooks e Stores, oltre a una separazione chiara tra componenti. Sarà utilizzato ESLint per mantenere uno stile di codifica adeguato.

- Framework:

Il backend utilizzerà Spring Boot per l'architettura RESTful e la gestione del database, mentre il frontend si baserà su React. Entrambi seguiranno le linee guida ufficiali di ciascun framework.

- Database

Il database sarà embedded e utilizzerà il framework H2

## 4.2 standard

Tutto il codice sarà gestito tramite i sistemi di controllo della versione forniti da Git e GitHub, utilizzando un modello di branch e issue utilizzando una Kanban Board.

Ogni task del progetto seguirà il flusso operativo passando attraverso gli steps definiti da un classico modello Kanban: BackLog, Ready, In progress, In review e Done.

Per ogni task verrà creato un nuovo branch che dovrà essere sincronizzato durante lo sviluppo.



Sarà utilizzato Swagger per generare documentazione automatica delle API backend.

Verranno realizzati diagrammi UML per fornire una documentazione di tutti i processi di backend e i flussi seguiti.

### ***4.3 Piano di controllo della configurazione***

Per mantenere una coerenza durante lo sviluppo verrà applicato un sistema di controllo di

Gestione delle versioni, dove ogni versione della piattaforma sarà identificata con un sistema di versioning incrementale (es. v1.0.0).

Ogni modifica significativa richiederà l'approvazione da parte del Project Manager.

## ***5. Attività di gestione***

### ***5.1 Documentazione degli sviluppi operativi***

Lo stato di avanzamento del progetto sarà organizzato all'interno di una Kanban board realizzata sfruttando i template predefiniti di GitHub.

All'interno di essa saranno inserite le varie task funzionali che seguiranno un workflow durante il loro sviluppo avanzando all'interno degli steps predefiniti.

### ***5.2 Bilanciamento delle priorità***

I requisiti saranno suddivisi in prioritari secondo uno schema MoSCoW

Verrà adottato un approccio iterativo tramite l'approccio agile SCRUM che consenta di rilasciare progressivamente le funzionalità chiave, per poi proseguire con le funzionalità secondarie e se il tempo lo permette con le funzionalità aggiuntive.

### ***5.3 Gestione delle tempistiche***

Le tempistiche saranno definite all'interno della Kanban board stessa basandosi sulle priorità pianificate.

Eventuali ritardi saranno gestiti con piani di recupero, ridefinendo le priorità.

## 6. *Rischi*

### *6.1 Rischi relativi alla coerenza con i requisiti*

è possibile che siano presenti disallineamenti con le aspettative dei clienti finali e con i requisiti iniziali, soprattutto quando questi ultimi non sono chiari o cambiano durante le fasi di sviluppo.

Appunto per questo motivo vengono applicate delle procedure Agili, che permettono sviluppi incrementali, in modo tale da permettere un riallineamento veloce tra le parti e la ripresa della realizzazione di un progetto conforme alle aspettative

### *6.2 Rischi relativi al rilascio della piattaforma*

è necessario che la piattaforma realizzata sia il più user-friendly possibile, in quanto la maggior parte dei clienti finali molto probabilmente non è abituata all'utilizzo di tali tecnologie e potrebbe trovarsi spaesata di fronte a una piattaforma di questo tipo.

Questo potrebbe tradursi in uno scarso successo della piattaforma e una perdita sostanziale di clienti e ricavi futuri.

è necessario, perciò effettuare studi di UI/UX che permettano di progettare il sistema nel miglior modo possibile.

### *6.3 Rischi dovuti a ritardi nello sviluppo*

è possibile che nelle fasi di sviluppo si presentino ritardi rispetto al piano previsto, in quanto un sistema ben sviluppato, conforme ai requisiti di qualità sia di sviluppo che finali, richiede molto più tempo per essere sviluppato.

Tuttavia, secondo le linee guida aziendali è un requisito fondamentale, la cosa importante è riorganizzare le priorità tempestivamente in modo tale da rispettare i requisiti essenziali nei tempi prestabiliti.

## **7. Personale**

### **7.1 Struttura del management**

Il team seguirà un approccio cross-funzionale con ruoli specifici definiti in precedenza:

- **Project Manager**

Avrà il compito di pianificare e supervisionare l'intero progetto, gestire le finestre temporali e le comunicazioni con gli stakeholder.

Competenze Richieste: Capacità di gestione del progetto, conoscenza delle metodologie Agili e SCRUM.

### **7.2 Struttura del team di sviluppo**

- **Analista**

Avrà il compito di Identificare e analizzare i requisiti funzionali oltre a scrivere le specifiche dettagliate.

Competenze Richieste: Familiarità con i processi clinici e capacità di elicicazione dei requisiti.

- **Sviluppatori (Frontend, Backend, Modellazione)**

Avranno il compito di codificare la piattaforma, implementare l'interfaccia utente

(frontend), gestire il database e le API (backend) oltre a documentare il progetto.

Competenze Richieste:

- Frontend: conoscenza del linguaggio JavaScript, framework React.
- Backend: Conoscenza del linguaggio Java, dei framework SpringBoot e Maven, degli ambienti di sviluppo Eclipse e IntelliJ.
- Documentazione: Conoscenza di Papyrus e dei diagrammi UML.

## *8. Metodi e tecniche utilizzate*

Durante lo sviluppo della piattaforma, saranno adottati metodi Agile con iterazioni brevi e feedback costanti dagli utenti per garantire dei risultati ottimali.

### *8.1 Ingegneria dei Requisiti*

Interviste e questionari saranno utilizzati per raccogliere informazioni dai chirurghi estetici.

I requisiti saranno descritti tramite use case diagram secondo gli standard UML e aggiornati in base ai feedback ricevuti.

### *8.2 Progettazione*

Verrà adottato un approccio incrementale, con l'uso di diagrammi UML per rappresentare le interazioni tra i componenti. Inoltre, verranno deployate versioni parziali dell'applicativo, sui quali verranno raccolti feedback prima di completare lo sviluppo completo.

### *8.3 Controllo della configurazione*

La codifica seguirà gli standard di qualità definiti nel piano del progetto.

Il versioning del codice avverrà tramite Git, e saranno rispettate le convenzioni di codifica per garantire la leggibilità e la manutenibilità del codice.

Utilizzando Git come strumento di controllo versione, tutte le modifiche al codice saranno tracciabili e monitorabili.

## ***8.4 Piano di Documentazione:***

Saranno prodotti diagrammi UML durante tutto lo sviluppo del software,

Sarà prodotta la documentazione necessaria per un corretto sviluppo SW.

## **9. Qualità**

Come riferimento per i requisiti di qualità verrà utilizzata la tassonomia di McCall:

- Funzionamento del prodotto
  - Correttezza
  - Affidabilità
  - Efficienza
  - Integrità
  - Usabilità
- Revisione del prodotto
  - Manutenibilità
  - Testabilità
  - Flessibilità
- Transizione del prodotto
  - Portabilità
  - Riutilizzabilità
  - Interoperabilità

## ***10. Workpackages***

I pacchetti di lavoro relativi allo sviluppo saranno suddivisi come segue

1. Modellazione delle classi Java per la parte di backend con Papyrus

2. Modellazione di altri diagrammi utili utilizzando Papyrus o altro plugin per la realizzazione di diagrammi UML.
3. Creazione della root del progetto suddivisa nelle seguenti 3 sezioni:
  - frontend
  - backend
  - documentazione
4. Realizzazione dell'interfaccia e delle API per l'autenticazione e per la sicurezza del sistema.
5. Realizzazione delle prime API necessarie per la gestione dei pazienti
6. Realizzazione della dashboard pazienti
7. Realizzazione delle API relative a gestione di visite e trattamenti
8. Realizzazione della dashboard visite e trattamenti
9. Realizzazione delle funzionalità per il caricamento delle foto
10. Implementazione funzionalità per il caricamento delle foto

## ***11. Risorse***

Le risorse sono una parte fondamentale per la riuscita del progetto, la scelta di risorse adeguate e il loro migliore sfruttamento sono fondamentali per la realizzazione di un applicativo di qualità.

### ***11.1 Personale***

Il team di sviluppo sarà composto da diverse figure professionali, ognuna con compiti specifici determinati accuratamente durante la fase di progettazione.

Si riporta alla sezione 7.2 per la descrizione dei ruoli.

### ***11.2 Risorse software***

Il progetto richiede diversi strumenti software tra cui:

- strumenti di sviluppo: i principali editor di codice utilizzati saranno Visual Studio Code per la parte frontend, Eclipse per lo sviluppo di diagrammi UML e IntelliJ per lo sviluppo della parte Java backend
- Strumenti di gestione del progetto: per la pianificazione saranno utilizzate la kanban board di github con l'aiuto di un software gestionale sempre in stile Kanban come Trello, per facilitare la visualizzazione delle task da svolgere.

### *11.3 Risorse hardware*

Il progetto sarà sviluppato in locale tramite localhost quindi l'unica risorsa hardware necessaria è un pc per ogni membro del team abbastanza efficiente per permettere una programmazione agevole e rapida.

## *12. Budget*

Trattandosi di un progetto universitario il budget è nullo.

## *13. Cambiamenti*

I cambiamenti durante lo sviluppo saranno inevitabili, la parte fondamentale sarà gestirli in maniera ordinata ed efficace soprattutto in un processo Agile come quello che verrà applicato. tutte le modifiche saranno realizzate applicando un sistema di versioning (es. 1.0.0).

La manutenzione del sistema verrà riportata in un documento apposito.

## *14. Consegna*

La consegna avverrà tramite GitHub, condivisa con l'utente finale, il quale potrà scaricare il sistema finale, con l'assistenza di un tecnico incaricato e portarlo su un proprio hosting dedicato.

La consegna di piccole versioni di test incrementali, saranno invece mostrate al cliente in ambienti di demo dagli sviluppatori.