

Progetto di ingegneria del Software

Anno accademico 2022-2023

Roberto Bertocchi 1060957

Mirko Carrara 1059971

Stefano Cosseddu 1067656

Francesco Peracchi 1057616

Indice

* **0. Il progetto**
* **1. Project plan**
  + 1.1 Introduzione
  + 1.2 Modello di processo
  + 1.3 Organizzazione del progetto
  + 1.4 Standards, linee guida e procedure
  + 1.5 Attività di gestione
  + 1.6 Rischi
  + 1.7 Membri
  + 1.8 Metodi e tecniche
  + 1.9 Garanzie di qualità
  + 1.10 Package di lavoro
  + 1.11 Risorse
  + 1.12 Budget e pianificazione
  + 1.13 Cambiamenti
  + 1.14 Consegna
* **2. Software lifecycle**
* **3. Configuration management**
  + 3.1 Struttura del progetto
  + 3.2 Issues
* **4. People management**
* **5. Software quality**
  + 5.1 Operatività del software
  + 5.2 Revisione del software
  + 5.3 Transizione verso un nuovo ambiente
* **6. Requirement engineering**
* **7. Modelling**
  + 7.1 Grafica
  + 7.2 Funzionamento del back-end
* **8. Software architecture**
* **9. Software testing e manutenibilità**

Il progetto

L’applicazione Fitness app viene sviluppata in java tramite l’utilizzo del software Android Studio.

Questa applicazione permette di creare una scheda di allenamento personalizzata da visualizzare su qualsiasi dispositivo tramite l’uso di un database.

Grazie a questa applicazione l’atleta potrà tener traccia di tutti i suoi allenamenti svolti e potrà monitorare i miglioramenti nel lungo periodo dei vari esercizi.

**Software lifecycle**

Per lo sviluppo del progetto, il team ha scelto un approccio **agile ,** questo perché la metodologia agile si adatta meglio alla metodologia di lavoro desiderata, ovvero:

1. Si considera importante il gruppo, le abilità dei suoi membri e le loro interazioni.

I lavori vengono assegnati in base alle capacità personali riducendo il tempo necessario allo studio di una competenza non conosciuta.

Nel caso in cui un componente della squadra non sia in grado di portare a termine un task, il team si riunisce per trovare una soluzione.

1. Nel team non c'è una struttura di tipo gerarchico, tutti i membri hanno la stessa importanza e possono esprimere le loro opinioni.
2. Si considera più importante un prodotto software funzionante piuttosto che una documentazione estremamente elaborata. Ciò consente di dedicare più tempo allo sviluppo e ottenere un prodotto pubblicabile in minore tempo con un risparmio di risorse. Nel caso sia necessario, la documentazione potrà essere espansa o modificata in seguito.
3. team è organizzato secondo la filosofia dell'extreme programming e durante lo sviluppo si è sfruttato spesso il pair programming come metodo di verifica.

I programmatori in coppia hanno lavorato sullo stesso frammento di codice in modo tale da avere una verifica in tempo reale sul lavoro svolto.

1. Si è anche sfruttata la tecnica del timeboxing per la suddivisione dello sviluppo a intervalli temporali entro i quali determinate funzionalità devono essere implementate ciò è avvenuto tramite l’utilizzo di branch e di pull request(funzioni contenute all’interno di github).
2. Si è scelto di dare molta importanza alla collaborazione con gli utenti, infatti dopo aver messo appunto le funzionalità base dell'applicazione, si è deciso di dare la possibilità agli utenti di collaborare mediante suggerimenti e feedback o anche attivamente allo sviluppo degli aggiornamenti.
3. Il team è favorevole al cambiamento e si impegna a dedicare il giusto tempo alla pianificazione del futuro del software.

Per la progettazione del software è stato utilizzato il model-driven architecture, ovvero prima di procedere alla scrittura del codice sono stati costruiti dei modelli in UML che rappresentano gli obiettivi e il funzionamento dell’applicazione fitness app.

Configuration managment

Il lavoro svolto, che si tratti di documentazione o di codice, viene regolarmente salvato nel repository di GitHub in condivisione con tutti i membri del team.

Utilizzo di github

Il repository è strutturato nel seguente modo:

utilizziamo 2 diversi branch, i quali ci permettono di sviluppare il progetto senza modificare oggetti essenziali all’interno del main del progettp

branches:

* main: Contiene le versioni stabili del codice e la documentazione
* dev: Contiene le versioni in via di sviluppo del codice

Successivamente suddividiamo il progetto in cartelle per aver una gestione migliore

cartelle:

code: Contiene il codice sorgente del progetto.

docs: Contiene la documentazione del progetto

Nei vari incontri tra i componenti del progetto agile vengono creati i temi che dovranno essere sviluppati.

Le varie attività sono create come issue, aggiungendo una breve descrizione del lavoro da svolgere.

Durante la settimana i membri del team sono liberi di creare nuove issue in base alle necessità.

Le issue sono suddivise in 3 categorie principali:

task: attività da svolgere

bug: segnalazione di un bug

enhancement: proposta di miglioramento

Gli stati che può assumere il progetto possono essere:

* Not started: L'attività non è ancora stata iniziata
* Working on it: L'attività è in fase di sviluppo
* To test: L'attività è stata completata e deve essere testata
* Testing: L'attività è in fase di testing che comprende la scrittura di test JUnit o la verifica manuale
* To fix: L'attività è stata testata ma non funziona correttamente
* Done: L'attività è stata completata e funziona correttamente