



UNIVERSITÀ
DEGLI STUDI
DI BERGAMO

Roberto Bertocchi 1060957
Mirko Carrara 1059971
Stefano Cosseddu 1067656
Francesco Peracchi 1057616

SOFTWARE ENGINEERING ASSIGNMENT

Sviluppo di un'applicazione Android "Fitness app"
Per la gestione, creazione e visualizzazione piani di
allenamento

project planning

Changelog

Versione 1 con i 14 punti del capitolo 2.1 "PLANNING A SOFTWARE
DEVELOPMENT PROJECT" V1.1 Corretta matricola

1. Introduzione

Il nostro progetto d'esame di Ingegneria del software consiste nello sviluppo di un applicativo mobile, "**Fitness app**", il cui obiettivo è quello di supportare gli sportivi, offrendo una piattaforma intuitiva e funzionale che permetta di tenere traccia delle varie sedute di allenamento, la loro programmazione e i progressi ottenuti.

In particolare questa applicazione creata per il sistema operativo Android prevede la creazione di schede relative ad ogni specifico allenamento, dove in ogni scheda vengono inseriti i vari esercizi svolti

Le schede poi possono essere visualizzate per tipologia di allenamento in modo da poter monitorare i progressi relativi ad uno specifico allenamento

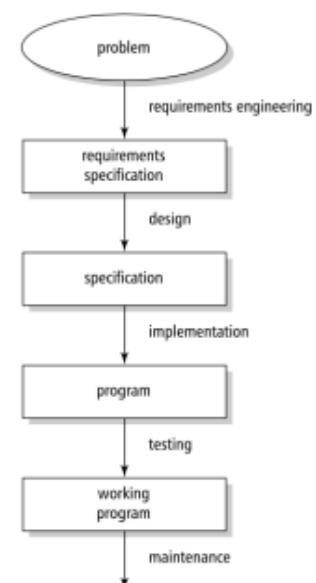
2. Modello di processo

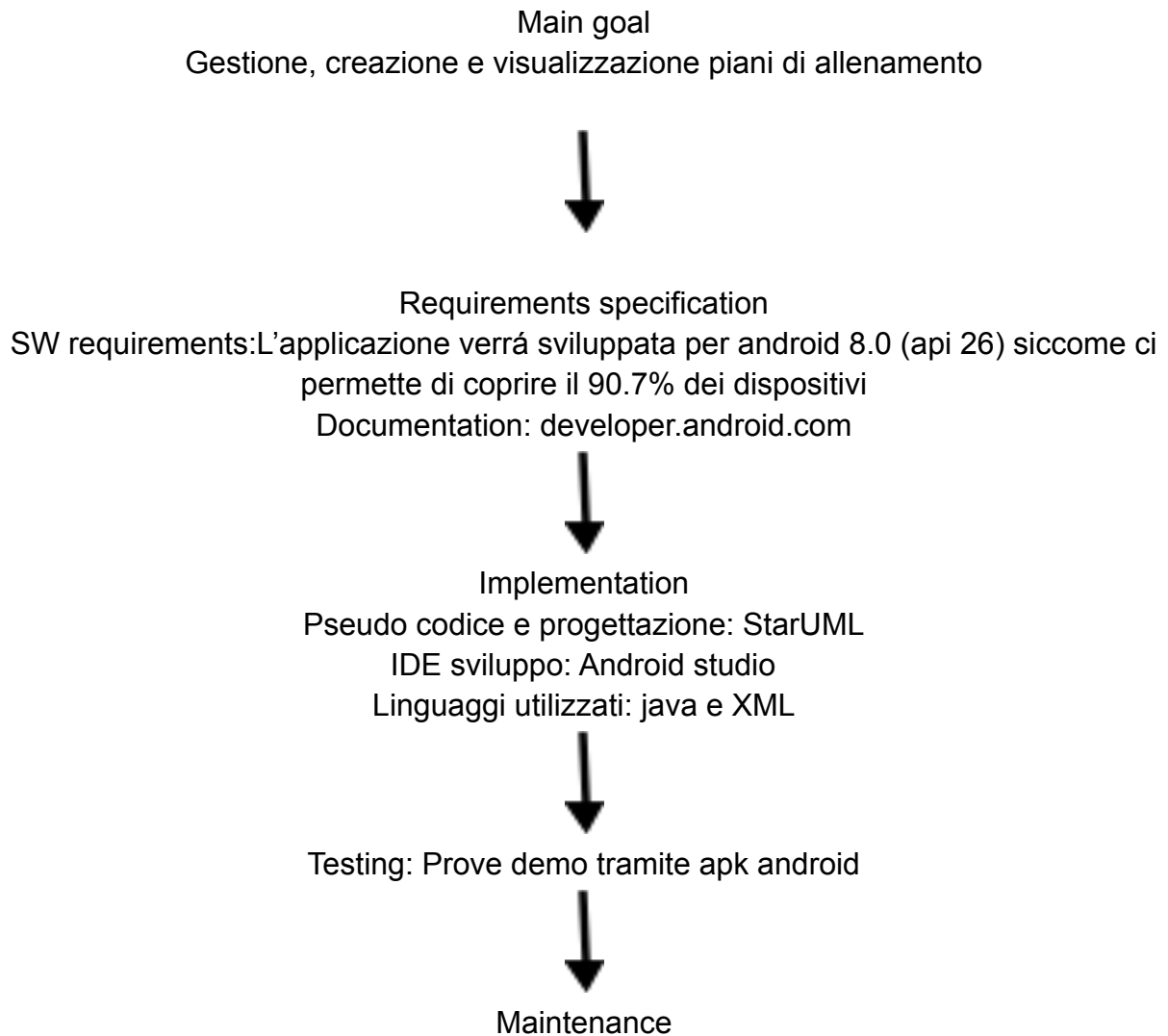
Un' applicazione mobile può variare molto dal punto di vista della complessità; dal tipo di funzionalità che deve implementare, dall'esistenza o meno di una base di dati, dall'accesso ad internet fino al poter sfruttare determinati comportamenti hardware del dispositivo.

Nel nostro caso il modello di processo che più si adatta ad un applicativo come il nostro, è sicuramente un modello lightweight agile, in quanto pensiamo che i feedback degli utenti saranno fondamentali per migliorare il servizio offerto, e che sicuramente comporteranno dei cambiamenti sia lato UI che lato funzionalità.

Infatti il modello agile presenta alcuni valori chiave che si adattano perfettamente al software che vogliamo creare, quali

- Collaborazioni con il cliente, più importanti di aspetti contrattuali
- Risposta al cambiamento, più importante rispetto al planning
- individui e interazioni, più importanti rispetto a processi e strumenti





3. Organizzazione progetto

Un progetto simile sicuramente richiede vari campi da coprire: dalla progettazione dell'interfaccia grafica, la gestione dell'eventuale base di dati e ovviamente il codice dell'applicazione che comprende una parte dedicata all'interfaccia e una orientata verso il codice java, puramente algoritmica, in cui si definiscono classi, oggetti ecc. I ruoli come Project manager, programmatore, analista, vengono definiti in fasi successive di sviluppo del progetto, in quanto non abbiamo schemi rigidi in base anche al modello di processo agile. Ci sembrano appropriati approcci come XP o Extreme Programming, metodo agile per eccellenza.

4. Standard, linee guida, procedure

Nel nostro caso per aiutare l'utente in caso di difficoltà nell'utilizzo dell'app abbiamo inserito una sezione di "domande frequenti" dove saranno descritti i problemi più comuni che si possono riscontrare nell'utilizzo dell'applicazione.

Per poter rendere l'applicazione più "user friendly" possibile, è buona norma fare riferimento alle Google Guide Lines, presenti nel sito Material.io, in questo modo l'utente è già "istruito" sulla posizione di molti elementi di UI; questo perché molte delle applicazioni che si usano nella vita di tutti i giorni, vengono sviluppate con le medesime meccaniche sia di visualizzazione che di interazione.

<https://material.io/resources/tutorials#android-java>

5. Attività di gestione

Il repository presente su github, ci permette, oltre che a lavorare al progetto, di poter rapidamente segnalare problemi, eventuali parti mancanti e l'accesso contemporaneo ai file con il salvataggio dei progressi.

6. Rischi

I rischi principali a cui si va incontro, nello sviluppo di un software tale, sono da rapportare alla concorrenza e alla sfruttabilità dell'applicazione, in quanto possono già essere presenti altre applicazioni che presentano funzioni simili, per questo motivo è importante implementare le stesse funzioni, ma in maniera più semplice e rapida. Oltre a questo, più nello specifico, è importante evidenziare quali possono essere gli aspetti più critici dell'applicazione, quali funzioni possono dare problemi e la difficoltà di risoluzione di essi.

7. Staff e skills richieste

Per poter portare avanti questo progetto, sono necessarie determinate competenze di Progettazione, quali l'utilizzo del software StarUML per la realizzazione di Casi d'uso, Activity diagrams ecc. L'applicazione Android, in quanto tale, richiede anche la conoscenza dell'IDE Android Studio, in cui si sfrutterà il linguaggio ad oggetti Java, per la parte di codice riguardante la logica, e XML per la parte di interfaccia e componenti grafici e realizzazione di template di schermate e icone (ad esempio qualche stilizzata icona relativa al fitness)

8. Metodi e tecniche

In questa sezione utilizziamo un ambiente object oriented(java) per lo sviluppo del codice e tramite l'utilizzo di Github viene utilizzato un metodo agile in cui ogni componente del gruppo condivide una parte del codice con relativa documentazione. Successivamente viene utilizzato Android Studio per la creazione e le specifiche dell'applicazione android "**Fitness app**".

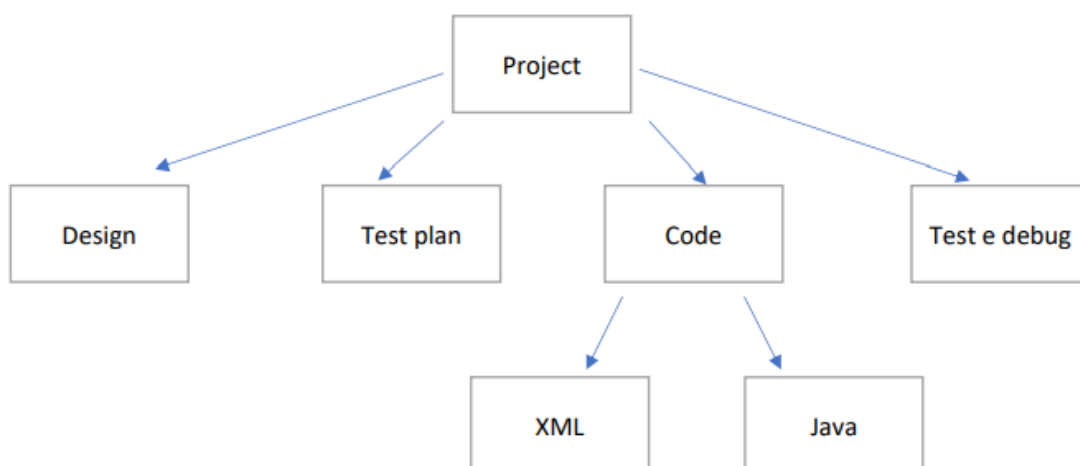
9. Garanzia di funzionamento e qualità

Per poter garantire il corretto funzionamento dell'applicazione, è necessario che la versione del sistema operativo su cui girerà il software, sia android.

Si dovrà quindi indicare nella creazione del progetto questa caratteristica. Inoltre è importante garantire il corretto funzionamento anche in base ai diversi DPI (densità di pixel per inch) dei display in quanti possono esserci problemi di visualizzazione dell'interfaccia.

Si potrebbero anche implementare delle impostazioni relative alla lingua di sistema, oppure alle unità di misura utilizzata negli esercizi (come Kg o Libbre)

10. Pacchetti di lavoro (workpackages)



11/12. Requisiti minimi e risorse + costi e budget

Requisiti minimi e consigliati per l'IDE Android Studio e la macchina virtuale per emulare android ed eseguire l'applicazione e fare debug

System requirements			
Windows	Mac	Linux	Chrome OS
<ul style="list-style-type: none">64-bit Microsoft® Windows® 8/10x86_64 CPU architecture; 2nd generation Intel Core or newer, or AMD CPU with support for a Windows Hypervisor8 GB RAM or more8 GB of available disk space minimum (IDE + Android SDK + Android Emulator)1280 x 800 minimum screen resolution	<ul style="list-style-type: none">MacOS® 10.14 (Mojave) or higherARM-based chips, or 2nd generation Intel Core or newer with support for Hypervisor.Framework8 GB RAM or more8 GB of available disk space minimum (IDE + Android SDK + Android Emulator)1280 x 800 minimum screen resolution	<ul style="list-style-type: none">Any 64-bit Linux distribution that supports Gnome, KDE, or Unity DE; GNU C Library (glibc) 2.31 or later.x86_64 CPU architecture; 2nd generation Intel Core or newer, or AMD processor with support for AMD Virtualization (AMD-V) and SSE38 GB RAM or more8 GB of available disk space minimum (IDE + Android SDK + Android Emulator)1280 x 800 minimum screen resolution	<p>For information on recommended devices and specifications, as well as Android Emulator support, visit chromeos.dev.</p>

Requisiti minimi per il software StarUML.

Il software in questione richiederebbe una licenza commerciale, dal costo di 129\$/utente

System Requirements

The following are the minimum system requirements for running StarUML™.

Fonti:

- Intel® Pentium® 233MHz or higher
- Windows® 2000, Windows XP™, or higher
- Microsoft® Internet Explorer 5.0 or higher
- 128 MB RAM (256MB recommended)
- 110 MB hard disc space (150MB space recommended)
- CD-ROM drive
- SVGA or higher resolution monitor (1024x768 recommended)
- Mouse or other pointing device

<https://developer.android.com/studio>

[http://staruml.sourceforge.net/docs/user-guide\(en\)/ch01.html](http://staruml.sourceforge.net/docs/user-guide(en)/ch01.html)

Costi delle relative macchine per poter far girare i software (1000€ CA.).

Licenza account Google Developer (25\$)

La licenza di Android Studio è open source.

13. Cambiamenti

Google, attraverso la sua piattaforma di pubblicazione, permette agli utenti di recensire e commentare l'app, in base alla loro esperienza utente, sottolineando criticità e possibili funzionalità.

In seguito, grazie a questi commenti, valuteremo quali modifiche apportare all'applicazione.

14. Rilascio applicazione.

L'applicazione verrà rilasciata su Google Play Store in maniera gratuita.

Per farlo, occorre un account developer per poter fare accesso a Google Play Console, dal costo di 25\$.

Si può valutare in base ai refactoring, l'implementazione di alcune funzionalità nuove, utilizzabili solo acquistando una "versione completa", implementando un "in-app purchase" attraverso una libreria messa a disposizione direttamente da Google, chiamata Billing Library version 3.