Documentation for Software engineering project

Nicolas Anselmi, David Guzman Piedrahita and Marco Vinciguerra

24 gennaio 2022

1 Project Plan

1.1 Introduction

Il progetto prevede lo sviluppo di una mobile app per gestire la prenotazione di un negozio di parruccheria. C'è la possibilità di avere due tipi di utente (anche in entrambe le mdoalità): un cliente e un padrone di negozio. La novità di questo progetto consiste nel fatto che è un tipo di sistema P2P in cui un utente può essere contemporaneamente cliente e (se vuole) gestore. Questo modello di business è fatto anche da Uber in cui un autista può essere sia cliente che autista.
Il target della clientela è market driven in quanto il core business dell'azienda si basa su percentuali delle transazioni.
Il cliente ha la possibilità di prenotare diversi tipi di acconciatura direttamente senza interfacciarsi /chiamare direttamente il proprietario del negozio ma tramite l'applicazione.Ogni tipologia di taglio selezionabile ha una durata e in base al tipo dei macchinari e delle risorse che devono essere utilizzate e può consentire un orario customizzabile di prenotazione da parte del cliente.
I membri del team sono: Nicolas Anselmi, David Guzman Piedrahita e Marco Vinciguerra.

1.2 Process model

Il life cycle del progetto è agile, in particolare la tecnica utilizzata è SCRUM con sprint di 5 giorni in quanto il tempo necessario per lo sviluppo è breve. Inoltre viene utilizzato un triage per gestire i compiti (MoSCoW).
Inoltre viene applicato un modello di prototipazione incrementale in cui ad ogni git viene aggiunto una funzionalità al sito e deve essere sempre disponibile su Github una versione funzionante del progetto.
Per accelerare il processo di apprendimento viene applicato anche il processo di pair programming, pratica molto utilizzata nei metodi AGILE.
Per quanto riguarda i requirements si utilizza il Kano Model. Il periodo di sviluppo parte poco prima di Natale. Ogni giorno verso le 9 30 del mattino c'è un daily scrum tenuto dallo scrum master in cui si discutono le problematiche riscontrate durante il giorno precedente e le possibili soluzioni a queste.

1.3 Organization of the project

Il progetto, dovuto alla sua natura, deve intefacciarsi sia con utenti che usufruiscono del servizio di prenotazione, sia da utenti che mettono a disposizioni i loro servizi commerciali. Il team di sviluppo è composto dai succitati integranti. Per portare a termine l'applicazione, ci sono dei knowledge-gap che dovranno essere colmati tramite la lettura di documentazione e l'uso di risorse online: particolarmente nel caso del framework Flutter per il frontend development.

1.4 Standards, guidelines, procedures

I principali linguaggi di programmazione del progetto saranno: python e dart, quest'ultimo viene esteso tramite flutter. Si usa i coding standards di flutter.
Per gestire l'assegnazione e il corretto sviluppo si usa un template di Notion per gestire i compiti.
Gli IDE che vengono utilizzati sono: VSC e terminale con VIM.

1.5 Management activities

Ogni settimana viene fatto un report sui progressi in corso fatti dal team di sviluppo per avere un'idea sull'avanzamento del progetto.
Le modifiche critiche del progetto devono essere accettate dal padrone della cartella (Marco Vinciguerra), le altre possono essere fatte liberamente.
Questi report, insieme alle decisioni definite negli scrum meeting, rappresentano la principale strategia per valutare

velocemente lo status del progetto, e, di conseguenza, sono utili a fare course-correction.
Difatti, è proprio in questo modo che è previsto bilanciare l'equilibrio tra requirements e l'impegno necessario per soddisfarli.

1.6 Risks

Il rischio principale è di non consegnare in tempo il progetto o di non consegnare un progetto perfettamente funzionante.

1.7 Staffing

I membri del team sono: Nicolas Anselmi, David Guzman Piedrahita e Marco.
Per provare a vedere come funziona il mestiere il ruolo dello SCRUM master cambia a rotazione e si parte dalla settimana che inizia col 13 dicembre. Il primo SCRUM master sarà David, la settimana dopo Nicolas e quella dopo Marco Vinciguerra e così via...
Sono 3 studenti di ingegneria del terzo anno.

1.8 Methods and techiniques

Per gestire gli sprint è stato utilizzato un template di Notion in quanto ha la possibilità di schedulare i task in base alla scadenza e in base ad un ordine gerarchico.
Per gestire la fase di testing si usa il tool better flutter tests che fa lui il testing sul framework Flutter. Il test viene scritto automaticamente dal tool e quindi non si applica fin da principio.
Per quanto riguarda la specifica dei requisiti si utilizza lo standard IEEE 830.
Il design pattern utilizzato è il delegation pattern.
Si utilizza Flutlab.io per convertire il modello fatto con Figma in codice sorgente Flutter, questa è una strategia model driven.
Si prevede l'uso di una strategia COTS per la scelta di diversi moduli o, più precisamente, widget di Flutter, che consentono di implementare velocemente elementi UI classici, senza scrivergli da zero.
Per quanto riguarda i test di Dart si utilizza Dart unit testing e si utilizza il TDD e per garantire la continuos integration si usa Travis CI che partirà ad ogni Git e ogni 24 ore.
Per i database non si fa nessun tipo di testing e si utilizza Firebase per creare e gestire il database delle prenotazioni.
La gestione delle modifiche viene svolta tenendo in conto le considerazione del punto 5 e, soprattutto, il punto 13.

1.9 Quality assurance

Per garantire la qualità del prodotto viene utilizzato lo standard ISO 9001.

1.10 Work packages

Alcuni dei sottoprogetti (work package) che sono stati definiti a priori.
In viste della natura agile del progetto, questi work-package saranno estesi e modificati o evoluti nelle diverse iterazioni del life-cycle:

- Fase di design di schemi UML
- Imparare ad utilizzare Flutter e dart
- Implementare l'applicazione con un'interfaccia grafica (front-end)
- Implementare la domain logic (back-end)
- Uso di un database
- Fare il testing sul prodotto
- Fare testing usando l'applicazioni in telefoni reali (non simulator)

1.11 Resources

Gli obiettivi di prototipazione proposti dal progetto in questione richiedono solo l'uso di computer adatti alla programmazione nei suddetti linguaggi e framework. Dopo diverse iterazioni di prototipazione è prevista la possibilità di usare dei cloud-server che ricevano e gestiscano le richieste degli utenti tramite i loro client/app.

1.12 Budget and schedule

Il tempo preventivato per il progetto è di circa 55 ore a testa, quindi in totale saranno richieste 150 ore.
In particolare si prevede che saranno richieste 30 ore per fare la documentazione, 10 ore per imparare Flutter e dart, 10 ore per implementarla e le restanti ore i database e il testing.

1.13 Changes

Col tempo potrebbero cambiare le richieste da parte del cliente durante la fase di validazione di ogni processo.
Al supporto delle attività di change management vengono utilizzati i due tool usati anche per altri aspetti del progetto, ovvero:
Github, che in questo caso funge da piattaforma per accettare o rifiutare le modifiche version-oriented e per consentire di usare il forked develpment;
e Notion che, non essendo un tool specifico per lo sviluppo, serve invece a gestire l'organizzazione delle tempistiche e dei sotto-progetti a un livello di astrazione più alto.
I contenuti di questi tool, e i diversi report dei punti precedenti, servono come guida per la compilazione di un eventuale configuration management plan che conterrà una management section e activities. I documenti generati come risultato del punto 5 sono particolarmente utili per la management section.

1.14 Delivery

Il project plan verrà consegnato entro il 27 dicembre 2021
La consegna verrà fatta 5 giorni prima dell'esame orale di gennaio.

1.15 People Management and Team Organization

1.16 Softare Quality

1.17 Requirement Engenieering-IEE830

1.18 Software Architecture

1.19 Software Design

1.20 Software Maintenance