

2024/25

Project Plan

Versione 2.0



Maffeis Riccardo – Mat. 1085706

Zanotti Matteo – Mat. 1085443



Indice

Introduzione.....	2
Modello di processo:	2
Organizzazione del progetto:.....	3
Standard, linee guida, procedure	4
Attività di gestione	4
Rischi	5
Personale:	5
Metodi e tecniche	5
Garanzia di qualità.....	6
Pacchetti di lavoro (workpackages)	6
Risorse	7
Budget e programma	7
Cambiamenti.....	8
Consegna	8

Introduzione

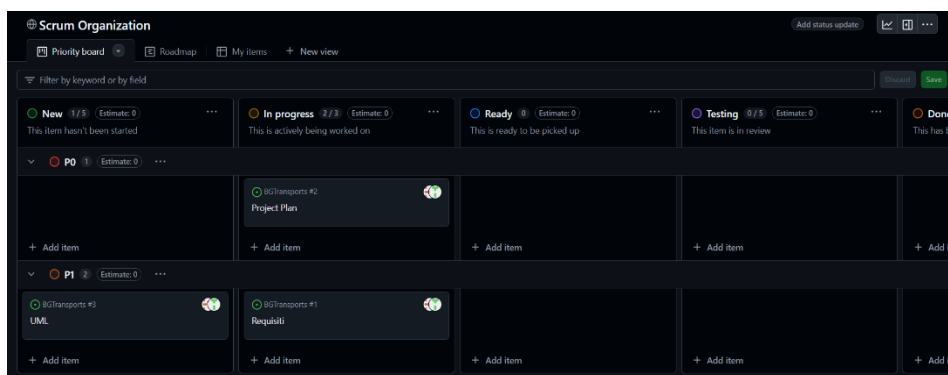
BGTransport è nato per facilitare il ritrovamento di informazioni dei mezzi pubblici nella provincia di Bergamo. Questo progetto permette di visualizzare in modo chiaro, semplice e veloce, all'interno di una pagina web, tutte le linee di pullman, percorsi ferroviari, la funicolare e le linee tranviarie di Bergamo.

Ci sono anche differenti ruoli basati sul tipo di utente: Sviluppatore, che può accedere e modificare tutte le sezioni, specialmente quelle a lui dedicate; Manager, che ha accesso alle sezioni a lui dedicate e può gestire e modificare le informazioni della propria azienda; gli Utenti, che potranno accedere alle funzioni di base e registrandosi possono ottenere dei vantaggi, cioè avere una propria area riservata dove registrare gli abbonamenti o salvare i percorsi, stazioni, fermate preferite.

Modello di processo:

Il modello che è stato scelto per la realizzazione del progetto è di tipologia Agile, in particolare su base SCRUM, poiché ci permette di suddividere il lavoro e di darci degli obiettivi da soddisfare settimanalmente.

Abbiamo utilizzato la sezione Projects all'interno di GitHub per poter osservare e gestire l'avanzamento del progetto, mediante la Kanban Table.



Suddivisione SCRUM

Andando ad approfondire meglio il modello scelto analizziamo gli elementi principali:

- Product Backlog:

Un elenco prioritizzato di tutte le funzionalità, bug e miglioramenti necessari per il prodotto, mediante la compilazione degli issue nella sezione project di Github. Il Product Owner è

responsabile di gestirlo, aggiornandolo e riorganizzandolo in base alle priorità e alle necessità del business. Il team di sviluppo stima ogni elemento per determinarne la difficoltà.

- Sprint Planning:

Durante questa riunione, che si svolge ogni venerdì, il team seleziona gli elementi più importanti dal Product Backlog da completare nel prossimo sprint. Si definisce l'obiettivo dello sprint e si pianifica il lavoro necessario.

- Sprint:

Abbiamo scelto che lo sprint dura una settimana, in ognuno di essi il team sviluppa funzionalità o risolve problemi. Alla fine dello sprint, l'obiettivo è avere un incremento del prodotto pronto, cioè completato secondo la definizione di "Done".

- Daily Scrum:

Il gruppo effettua una riunione quotidiana di circa 15 minuti per aggiornarsi sui progressi giornalieri.

- Sprint Review:

Essendo il team composto da due sole persone abbiamo deciso di organizzare una riunione ogni due settimane per ricontrollare insieme tutto ciò che è stato fatto fino a quel punto.

- Sprint Retrospective:

L'obiettivo di ciò è di identificare i miglioramenti che possono essere apportati in futuro e anch'essa è svolta ogni due settimane.

Organizzazione del progetto:

A svolgere il progetto siamo in due quindi la divisione dei compiti è abbastanza immediata:

- Product Owner: Zanotti Matteo --> si occupa di definire le funzionalità del prodotto; gestisce il Product Backlog, le funzionalità e le funzionalità da sviluppare
- Scrum Master: Maffeis Riccardo --> si occupa di far seguire le pratiche Scrum, mantiene il Development Team e il Product Owner focalizzati sugli obiettivi elencati nello Sprint Backlog e risolve eventuali problemi.
- Development team: Maffeis Riccardo, Zanotti Matteo --> si occupano di portare a termine i compiti assegnati durante lo Sprint

Standard, linee guida, procedure

Per poter svolgere al meglio le attività abbiamo delineato delle regole da seguire, in modo tale che non si possano verificare situazioni in cui i componenti del team non sanno cosa fare quando si verificano delle problematiche.

Tutto il team deve seguire le seguenti linee guida:

- Tutti i bug/implementazioni devono essere segnati nella sezione Projects di GitHub e associati ad un Issue
- Quando vengono fatte delle modifiche a documenti deve essere creato un Issue per far sì che ogni componente del gruppo possa vedere le modifiche e approvare i cambiamenti
- Il software deve essere sempre testato dopo qualsiasi modifica/implementazione per accertarsi che le modifiche/implementazioni siano corrette
- Per qualsiasi idea mirata al miglioramento del software bisogna discuterne insieme a tutto il team
- Per qualsiasi problema bisogna discuterne insieme a tutto il team per risolvere le problematiche nel minor tempo possibile

Attività di gestione

Gli obiettivi del progetto sono:

- Obiettivo principale:
Lanciare una nuova versione della piattaforma con nuove funzionalità primarie ogni all'incirca due settimane.
- Obiettivi secondari: Aggiungere funzionalità di ricerca avanzata, migliorare la sicurezza e correggere i bug noti.

Per la gestione degli obiettivi abbiamo deciso di assegnare ad ogni issue un grado di priorità, soprattutto per stabilire un ordine di lavoro. Per quanto riguarda i tempi di lavoro, in caso di ritardi si discute dei problemi riscontrati e si riorganizzano le priorità in base agli obiettivi dello sprint.

Rischi

I rischi vengono tenuti sotto controllo dal Product Owner, sia discutendo con il resto del team, sia effettuando una previsione di essi. Alcuni dei rischi previsionali, su cui dovremmo prestare particolare attenzione, sono i seguenti:

- Scrittura della documentazione in maniera dettagliata, come la specifica dei requisiti, ma anche la descrizione accurata delle classi sviluppate, poiché possono creare incomprensioni;
- Tempo per l'implementazione del software non sufficiente;
- Difficoltà di realizzazione di certe parti applicative del software;
- Problemi di suddivisione dei compiti.

Personale

Il team che si occupa dello sviluppo del progetto è composto da due persone: Maffeis Riccardo Zanotti Matteo. Entrambi hanno buone capacità di programmazione in java, derivanti da altri progetti sviluppati nello stesso ambiente di programmazione e una collaborazione perfezionata col tempo, derivante da altri progetti svolti con lo stesso gruppo. La data di inizio del progetto è stata il 18 ottobre 2024, mentre la fine di essa è stimata per gennaio 2025.

Metodi e tecniche

Inizialmente il team ha eseguito delle riunioni, definite come “Sprint”, nel quale sono stati stilati i principali obiettivi e sono stati realizzate le prime versioni dei diagrammi per la rappresentazione generale del progetto, mediante diagrammi UML.

Per la realizzazione del codice sono state utilizzate tecniche in grado di limitare al minimo i difetti, o eventuali errori, del programma, per questo le operazioni di sviluppo si dividono in 5 fasi: La prima fase, definita come “New”, viene utilizzata per descrivere con cura la parte del programma da sviluppare, per far sì che debba essere il più completa possibile e vada a escludere mancanze e problemi; una seconda, detta “In progress”, riguarda la vera e propria scrittura del codice, cercando di renderla utilizzabile per più parti e di facile comprensione mediante l'aggiunta di appropriati commenti e descrizione delle parti realizzate; una terza fase, che rappresenta il passaggio da “in sviluppo” a “Ready”, che segna la fine della scrittura del codice e che dovrà essere esaminata mediante dei test, ed un'ultima fase che viene definita come “Done”, nel quale verranno

considerate, di comune accordo, le differenti parti del progetto.

Per la gestione delle versioni il team ha optato ad un meccanismo semplice, ma funzionale: alle versioni iniziali è stato dato un nome, che resterà invariato nel corso del progetto, per far in modo che mediante un titolo si sa a quale parte o documento ci si riferisca. Per la numerazione si è deciso di comune accordo di assegnare un numero, a partire da 1, a tutte le nuove versioni che riguardano introduzioni di nuove parti o cambiamenti di notevole importanza, mentre per l'aggiunta di piccole parti, correzione di errori (bugfix) o sistemazioni varie, verrà aggiunto al numero della versione una sotto-versione, indicata mediante un "punto" e l'aggiunta di un altro numero (es. v1.1; v1.2; ...).

Garanzia di qualità

La Garanzia della Qualità nel nostro progetto coinvolge tutti i membri del team. Le procedure di qualità vengono integrate nel ciclo di vita del progetto, specialmente sulla sezione "Done".

Abbiamo scelto di optare per qualche test automatizzato, ma soprattutto basarci sul feedback continuo e sulla revisione costante del codice. Questa scelta è stata fatta per un motivo abbastanza semplice: i test automatizzati verranno usati per trovare i problemi veri e propri del software, quindi ci permetteranno di sistemare le parti mal funzionanti, ma i feedback delle persone, ci permetteranno di rendere più comprensibile il nostro progetto, che deve essere accessibile a tutti e il più semplice possibile da utilizzare.

Pacchetti di lavoro (workpackages)

Il progetto si può suddividere in quattro grandi attività da svolgere basandosi sulle tre parti generali nel quale esso si può scomporre: vi è una prima parte riguardante la gestione delle Aziende di trasporto, con la relativa suddivisione e gestione dei mezzi e dei luoghi di deposito di essi; è presente anche una seconda parte che riguarda la gestione degli utenti, i quali avranno funzionalità differenti a seconda del loro ruolo assegnatogli e in base alle funzioni che dovranno svolgere. Possiamo considerare anche una terza parte che riguarda la visualizzazione su mappa dei percorsi dei mezzi secondo la scelta degli utenti ed infine, una quarta parte riguardante lo sviluppo della parte grafica del progetto, che permetterà agli utenti finali di poter interagire con il programma senza difficoltà.

Risorse

Software

Ambiente di sviluppo:

Eclipse IDE per lo sviluppo in Java (versione 2024-09)

Java Development Kit (JDK):

Kit di sviluppo Java (versione JDK 23)

Papyrus: Tool per la creazione delle rappresentazioni UML (2024-06 release (6.7.0))

Hardware

Computer personali che dispongono delle caratteristiche sufficienti per lavorare con Eclipse, Papyrus, GitHub, ... (quantità: 2)

Programma

Lo Sprint Plannig del progetto è strutturato nel seguente modo:

Sprint	Data	Modalità	Decisioni
1	18/10/2024	Riunione in presenza	Miglioramenti al Project Plan e creazione del file che contiene i requisiti
2	25/10/2024	Riunione in presenza	Miglioramenti al Project Plan, miglioramenti dei requisiti
3	01/11/2024	Riunione in presenza	Termine prima versione ufficiale Project Plan e creazione primi schemi UML
4	08/11/2024	Riunione a distanza	Avanzamento creazione schemi UML e inizio scrittura codice

Ogni settimana viene definito un nuovo sprint:

Sprint	Data inizio	Data fine	Durata
1	21/10/2024	25/10/2024	1 settimana
2	28/10/2024	01/11/2024	1 settimana
3	04/11/2024	08/11/2024	1 settimana

Cambiamenti

I cambiamenti, in questo progetto, devono essere studiati da tutto il gruppo e approvati. In particolare, deve essere aperto un issue e assegnato a tutti, ma deve essere anche convocata una riunione per discutere dei cambiamenti e per l'approvazione di esso, sia che i cambiamenti derivino da un problema in ambiente di sviluppo, sia dai test, sia come richiesta degli utenti finali. Tutto ciò deve essere documentato e inserito nella specifica dei requisiti, ma soprattutto ci deve essere una specifica ben precisa della problematica, una motivazione argomentata del cambiamento effettuato e del tempo che occorre per sviluppare la soluzione.

Consegna

La consegna prevista del progetto è stimata per la prima settimana di gennaio del 2025 e prevede la completa raggiunta degli obiettivi concordati ed un'eventuale risoluzione dei problemi affrontati in corso d'opera.

Si stima che il progetto permetta a tutti gli utenti un utilizzo facilitato dell'interfaccia per semplificare la comprensione e l'utilizzo dei mezzi di spostamento pubblico nella provincia.