

Project Management

SPOSTAMI

Carlo Di Raddo (carlo.diraddo@studio.unibo.it)

Anno accademico 2022-2023 Cesena

Indice

1. Dominio Applicativo	3
2. Scoping	
2.1 Scoping	4
2.2 To-Be	6
2.3 Conditions of satisfaction	7
2.4 Requirement breakdown Structure	8
2.4.1 Facilitated group session	8
2.5 Project Overview Statement (POS)	10
2.6 Project Management Life Cycle	17
3 Planning	
3.1 Project definition statement	17
3.2 Work Breakdown Structure	18
3.3 Gestione dei costi	25
3.3.1 Stima durata e quantità del lavoro	25
3.3.2 Cash flow	26
3.3.3 Project network diagram	27
3.4 Agile sprint	28
4 Launching	
4.1 Developers	29
4.2 Modello organizzativo	30
4.3 Variazioni in corso d'opera	30
4.4 Work package	30
5 Monitoring and Control	
5.1 Sviluppo e Report	32
5.2 Controllo del budget	32
5.3 Meeting e Sal	33
5.4 Tracking dei Bug	34
5.5 Retrospettiva	34

6 Closing

6.1 Collaudo	35
6.2 Rilascio	36
6.3 Accettazione del committente	36
6.4 Documentazione di prodotto	37

Capitolo 1

Dominio Applicativo

SPOSTAMI è un'applicazione mobile focalizzata che nasce con l'obiettivo di creare un servizio concorrenziale rispetto al sistema di taxi locale. Le caratteristiche distintive di SPOSTAMI sono diverse. Inizialmente, l'idea è quella di sviluppare un'applicazione in grado di gestire due tipologie di utenti:

1. Clienti
2. Conducenti

La gestione di questi due gruppi di utenti consente di orientare l'attività di SPOSTAMI esclusivamente verso la creazione e il mantenimento dell'infrastruttura tecnologica, escludendo l'assunzione e la gestione dei conducenti e dei veicoli, poiché questi ultimi sono privati e desiderano offrire un servizio di trasporto su veicoli privati. SPOSTAMI ha l'ambizione di rivoluzionare l'attuale sistema di taxi, che costituisce un monopolio di mercato, riducendo il costo di ogni singola corsa e cercando di garantire che il cliente venga prelevato entro 10 minuti dall'invio della prenotazione.

Capitolo 2

Scoping

La fase di definizione dei requisiti deriva da un'analisi di mercato che ha confermato l'interesse dei cittadini per l'introduzione di una soluzione simile. Dopo un incontro preliminare di definizione dei requisiti, i partecipanti hanno redatto un Project Overview Statement (POS)

2.1 Scoping

L'incontro iniziale di definizione dei requisiti è stato preceduto dalla consegna di una relazione di analisi di mercato da parte di una società di Marketing Analysis, che ha fornito una breve descrizione del campo di applicazione del progetto. Il documento è stato attentamente studiato per agevolare la comprensione del settore di interesse. Il gruppo responsabile della definizione dei requisiti è composto dalle seguenti persone:

- Il CEO di SPOSTAMI.
- Il responsabile delle operazioni informatiche di SPOSTAMI.
- Il project manager della società di sviluppo software IT.
- Il team centrale per lo sviluppo di IT.
- Un facilitatore di IT.

Per arrivare allo sviluppo di un RBS (Requirement Breakdown Structure) approvata dal consiglio di amministrazione, sono state necessarie diverse iterazioni dell'incontro iniziale. Nei paragrafi successivi, verrà fornita una descrizione lineare delle varie fasi dell'incontro, ma i concetti rilevanti saranno ripresi nelle sezioni seguenti.

Primo meeting

Nella prima iterazione, il project manager ha richiesto una spiegazione dettagliata della situazione attuale, definendo in modo preciso i processi aziendali del sistema informativo. È stato anche richiesto di esporre in modo chiaro i desiderata al fine di studiare la gamma di processi che il sistema informativo dovrà implementare, definendo anche un budget per lo sviluppo del progetto. Al termine della prima iterazione, il project manager ha tenuto una riunione privata con il proprio team di sviluppo per analizzare la situazione attuale e studiare la soluzione più efficace ed efficiente possibile.

Secondo meeting

Nella seconda iterazione dell'incontro, il CTO ha proposto lo stack tecnologico e la soluzione ideata insieme al team di sviluppo. L'incontro è proseguito con la stesura del documento per le CoS (condition of satisfaction).

Terzo meeting

La terza iterazione dell'incontro si è focalizzata sull'analisi della nuova gamma di processi aziendali che SPOSTAMI supporterà, e infine sulla raccolta dei requisiti necessari per implementare il nuovo sistema. Questa fase si è conclusa con lo sviluppo concordato della Struttura di Scomposizione dei Requisiti (RBS).

Successivamente, il project manager ha condotto una riunione con i legali della società di sviluppo software per concordare la struttura del POS..

Quarto meeting

La quarta iterazione dell'incontro aveva come obiettivo la presentazione del POS. al consiglio di amministrazione (CdA) e la discussione dei vari dettagli. Infine, il POS è stato approvato e si è proceduto alla fase successiva di scoping.

2.2 To-Be

Rappresenta l'obiettivo in termini realizzativi da raggiungere per portare l'applicazione alla fase di delivery, l'obiettivo principale è quello di realizzare un'applicazione che riesca a scardinare l'attuale mercato monopolistico dei taxi, in seconda battuta gli obiettivi da centrare sono quelli di abbattere i costi per la singola tratta sfruttando l'idea di non dover possedere un parco auto né una flotta di driver, inoltre per incentivare l'uso del servizio si cercherà di garantire il prelevamento del cliente in un arco temporale pari ad un massimo di 10 minuti nella maggior parte dei casi. Elencati i desiderata si è quindi proceduto alla stesura dei dettagli legati ai subsystem da realizzare perchè la piattaforma sia funzionante, si è quindi ipotizzato un sistema software formato dai seguenti sottosistemi:

1. Sistema di registrazione
2. Sistema di selezione veicolo
3. Sistema di localizzazione driver
4. Sistema di prenotazione

5. Sistema di instradamento
6. Sistema di feedback
7. Sistema di pagamento
8. Sistema di scontistica

2.3 Conditions of satisfaction

- SPOSTAMI prevede che il progetto raggiunga lo stato di consegna entro 12 mesi dall'inizio.
- Ottimizzare l'algoritmo di ricerca al fine di garantire tempi di attesa compresi tra 0 e 10 minuti per la maggior parte delle richieste di corsa.
- Assicurarsi che il codice sia correttamente documentato e sviluppato secondo le migliori pratiche attualmente in uso nella programmazione.
- Il sistema deve essere scalabile e in grado di gestire picchi di carico senza subire collassi. È accettabile l'utilizzo di un bilanciatore di carico per distribuire il traffico di rete e prevenire interruzioni del servizio.

2.4 Requirement breakdown Structure

All'interno della Requirement Breakdown Structure si è deciso di inserire i requisiti ad alto livello del sistema. Essi poi saranno arricchiti con ulteriori funzionalità di basso livello, che verranno introdotte all'interno della Work Breakdown Structure. L'approccio utilizzato per la raccolta dei requisiti è stato l'uso dei Facilitated group session

2.4.1 Facilitated group session

L'utilizzo di un PMLC iterativo, in accoppiata con la Continuous delivery garantirà non solo al project ma anche alla direzione di SPOSTAMI di poter tenere sotto controllo l'avanzamento del progetto mediante release mensili effettuate in modalità di cumulative reports. L'obiettivo di questa sezione è stato quello di dettagliare i casi d'uso del sistema in questione cercando di trasporre i requisiti del cliente esposti in linguaggio naturale a requisiti funzionali per il CTO e l'architetto che hanno ideato l'architettura del progetto nella sua interezza

1. Sottosistema

1.1 Sistema di registrazione

1.1.1 Registrazione cliente

1.1.2 Registrazione driver

2. Sistema di selezione del veicolo

2.1 Selezione veicolo cliente

2.2 Inizializzazione sistema di prenotazione

- 3. Sistema di localizzazione driver
 - 3.1 Visualizzazione della mappa generale
 - 3.2 Posizionamento del cliente all'interno della mappa
 - 3.3 Posizionamento del driver all'interno della mappa
- 4. Sistema di prenotazione
 - 4.1 Presa in carico di una prenotazione
 - 4.2 Selezione del driver
 - 4.3 Inizializzazione sistema di pagamento
 - 4.4 Inizializzazione sistema di instradamento
- 5. Sistema di instradamento
 - 5.1 Acquisizione indirizzo di destinazione
 - 5.2 Calcolo percorso verso indirizzo di destinazione
- 6. Sistema di feedback
 - 6.1 Acquisizione feedback cliente
 - 6.2 Acquisizione feedback driver
 - 6.3 Sistema applicazione daspo
 - 6.4 Sistema applicazione ban
- 7. Sistema di pagamento
 - 7.1 Inserimento dati di pagamento
 - 7.2 Elaborazione pagamento
- 8. Sistema di scontistica
 - 8.1 Contatore chilometri percorsi
 - 8.2 Identificazione percentuale di sconto

2.5 Project Overview Statement (POS)

Problem/Opportunity

- Problema: Attualmente, i servizi di taxi nelle grandi città italiane non sono convenienti e i conducenti sono soggetti a restrizioni imposte dalle licenze che limitano il numero di veicoli disponibili, causando ritardi nel servizio.
- Opportunità: Un'analisi di mercato evidenzia la necessità di un sistema di taxi più ampio e distribuito. L'obiettivo è ridurre i costi per singola corsa al fine di renderla più economica rispetto ai taxi locali e garantire tempi di attesa ragionevoli. Inoltre, si mira a consentire ai privati di diventare conducenti senza la necessità di possedere una licenza statale.

Project Goal

Il Project Goal è costruire un'applicazione che sia una diretta concorrente dei taxi locali. Si mira a creare un sistema in grado di scalare rapidamente una volta lanciato, concentrandosi inizialmente su poche città. Dopo la fase di implementazione, il sistema dovrà essere stabile e ben collaudato al fine di ridurre rapidamente i costi di gestione.

Project Objectives

Sistema di registrazione

Sviluppare un sistema di registrazione che permette di registrare clienti e driver:

- lato cliente, effettuare una nuova registrazione inserendo anche una carta di credito univoca nel sistema per i pagamenti;
- lato driver, di effettuare una nuova registrazione inserendo anche un conto di appoggio univoco nel sistema per gli accrediti;
- visualizzare il nuovo utente al back office;

Sistema selezione veicolo

Sviluppare un sistema di selezione del veicolo in funzione della prestigiosità del veicolo (luxury o standard), numero di posti e la sua classe ecologica:

- Veicolo endotermico: veicolo con una tariffa maggiore al km;
- Veicolo ibrido: veicolo con una tariffa media al km;
- Veicolo elettrico: veicolo con una tariffa minore al km;

Sistema di localizzazione driver

Sviluppare un sistema per la localizzazione dei driver allo scopo di garantire:

- l'ottimale tracciamento della posizione del driver in caso di inconvenienti o incidenti e per permettere al cliente di sapere dove si trova il driver prima del prelevamento;
- la scalabilità del sistema nel momento in cui il numero di clienti e driver va ad aumentare; Sistema di prenotazione driver Sviluppare un sistema lato cliente allo scopo di effettuare le prenotazioni garantendo:
- la possibilità di inserire la destinazione attraverso l'applicazione.
- la selezione del driver più vicino al cliente che non superi i 10 minuti di tempo di attesa.
- un'ottimale tracciamento della posizione del cliente in termini di GPS unito al tracciamento dell'IP migliorando la precisione di prelevamento.
- l'accettazione o meno della richiesta da parte del driver tramite un'azione sull'applicazione, in caso positivo rimandando al cliente i dati utili all'identificazione della corsa.
- l'annullamento da parte del cliente entro 30 secondi dalla prenotazione della corsa.

- nel caso che il sistema di instradamento non riesca a garantire i 10 minuti massimi di attesa, la possibilità da parte del cliente di annullare la prenotazione per una troppa attesa.

Inoltre è da prevedere un sottosistema di interscambio di contatti tra driver e cliente per un contatto informale in caso di irrintracciabilità una volta raggiunta la posizione di prelevamento.

Sistema di instradamento

Sviluppare un sistema in grado di determinare:

- il percorso ottimale tra il driver e il cliente considerando tutti i parametri tra cui il traffico attuale, il prelevamento deve essere garantito nella maggior parte dei casi in massimo 10 minuti di tempo.
- il percorso ottimale dalla posizione in cui viene prelevato il cliente alla posizione di destinazione.

Sistema di feedback

Sviluppare un sistema di feedback interno che permette:

- l'invio facoltativo da parte del driver tramite l'applicazione di un feedback riguardante il cliente e il suo comportamento dal prelevamento al raggiungimento della destinazione.
- di aggiungere una nota di demerito in caso di comportamento negativo da parte del cliente da inviare come notifica al cliente.

- il controllo sul numero di note di demerito dei clienti, al raggiungimento di 10 note negative nell'arco temporale di 6 mesi applicare una tariffa maggiore.
- l'eventuale ban di un cliente per comportamenti dannosi/pericolosi.

Sviluppare un sistema di feedback per i clienti:

- l'invio facoltativo da parte del cliente tramite l'applicazione di un feedback riguardante il driver e il suo comportamento dal prelevamento al raggiungimento della destinazione.
- di aggiungere una nota di demerito in caso di comportamento negativo da parte del driver da inviare come notifica al driver.
- il controllo sul numero di note di demerito dei driver, al raggiungimento di 10 note negative nell'arco temporale di 6 mesi applicare una trattenuta al minuto maggiore rispetto allo standard.
- l'eventuale daspo per 2 mesi in caso di superamento del numero di note.

Sistema di pagamento

Sviluppare un sistema di pagamento in grado di:

- lato cliente, poter pagare la corsa attraverso la carta registrata non appena la corsa viene prenotata, in particolare nel momento in cui il cliente non può più annullare la corsa.

- lato driver, poter ricevere i pagamenti dai clienti su un conto di appoggio fornito al momento della registrazione, viene applicata una trattenuta su l'accredito del pagamento.

Sistema di scontistica

Sviluppare un sistema di scontistica in grado di applicare uno sconto progressivo del 5% fino al 30% ogni 50km viaggiati anche non continuativamente. L'utente è inoltre in grado di visualizzare l'attuale sconto e gli eventuali km rimanenti allo sconto successivo.

Success Criteria

- Prelevare in massimo 10 minuti dalla prenotazione nella maggior parte delle corse.
- I risultati di eventuali indagini e interviste figurano una soddisfazione nella maggior parte dei clienti in termini di tempistiche e costi del servizio.
- Gestione corretta del carico di ampie flotte di driver nonché di clienti.
- Scalare velocemente invogliando i driver nel sottoscrivere al servizio.

Assumptions

- Si considera il budget totale per il progetto stanziato pari a 800.000,00 euro sufficiente per la delivery dell'applicazione e del servizio.
- Si assume che il servizio non avrà grandi concorrenti ma un monopolio locale. Nonostante ciò inizialmente verrà avviata una campagna di marketing per portare conoscenza del servizio.
- Si assume che il servizio venga erogato in egual modo indipendentemente dalla città.
- La software house è in grado di portare a termine un progetto di grandi dimensioni.

Risks

- Impiegare troppo tempo nello sviluppo del servizio nonché il collaudo, questo comporterebbe il rischio che nel mercato subentri un nuovo competitor oltre all'attuale servizio di taxi locale.
- Non è possibile coprire la domanda dei clienti o l'offerta dei driver.
- La campagna di marketing non è così incisiva da portare nuovi driver e nuovi clienti a sufficienza. Obstacles Il mancato supporto da parte dell'applicazione di tutto il parco device attualmente presente.

Obstacles

Il mancato supporto da parte dell'applicazione di tutto il parco device attualmente presente.

2.6 Project Management Life Cycle

In funzione dei requisiti si è deciso di utilizzare un PMLC di tipo Iterativo. La scelta è derivata dall'ambiguità di alcuni requisiti perché un approccio iterativo dà la possibilità di cambiare e raffinare i task durante la fase di sviluppo del progetto.

Capitolo 3

Planning

All'interno di questa sezione, verranno descritte tutte le attività svolte durante la Sessione Congiunta di Pianificazione del Progetto (JPPS). Durante questa fase, sono state pianificate le attività necessarie per raggiungere il valore aziendale atteso dal cliente.

La fase di pianificazione si è basata su una serie di incontri iniziali di Kick-off, durante i quali è stato ripresentato l'output prodotto nella fase iniziale di definizione dei requisiti, descritta nel capitolo 2. Una volta completata questa fase, si è proceduto effettivamente alla pianificazione del lavoro, chiamata Working Session, la quale sarà descritta in dettaglio nelle seguenti sezioni.

Un modello iterativo guiderà il processo in un continuo processo di pianificazione. Il modello iterativo prevede di avere fasi ripetute di pianificazione durante l'intero arco del progetto, durante le quali saranno definiti e inseriti nella pianificazione mensile alcuni task prioritari, per poi procedere con lo sviluppo di tali task. Una volta completata l'iterazione, si avrà a disposizione un deliverable arricchito di funzionalità che verrà ulteriormente sviluppato nelle successive iterazioni.

3.1 Project definition statement

Durante questa fase, l'obiettivo principale è stato la scrittura del PDS (Project Definition Statement), che sarà consegnato al team di sviluppo e agli stakeholder. Questo documento avrà il compito di dettagliare gli obiettivi definiti in precedenza nel POS, al fine di chiarire ulteriormente l'obiettivo del progetto e risolvere eventuali problematiche o incomprensioni che possono emergere. Il PDS servirà come riferimento per garantire una migliore comprensione del progetto da parte di tutte le parti interessate.

3.2 Work Breakdown Structure

La WBS (Work Breakdown Structure) completerà la RBS descritta precedentemente nel capitolo 2 e definirà in modo gerarchico le attività e i compiti che il sistema dovrà soddisfare. L'approccio iterativo utilizzato ha permesso di sviluppare una WBS che verrà arricchita durante la pianificazione di ogni iterazione, aggiungendo alle foglie le attività e i compiti necessari per sviluppare le funzionalità corrispondenti. In questo modo, la WBS si evolverà nel tempo, includendo dettagli

specifici delle attività da svolgere per il completamento delle diverse funzionalità.

1. Sistema di registrazione

1.1. Client

1.1.1. Sviluppo interfaccia registrazione cliente

1.1.1.1. Inserimento dati personali

1.1.1.2. Inserimento carta di credito

1.1.2. Sviluppo interfaccia registrazione driver

1.1.2.1. Inserimento dati personali

1.1.2.2. Inserimento dati veicolo

1.1.2.3. Inserimento conto di appoggio

1.1.3. Sviluppo interfaccia visualizzazione back office

1.1.3.1. Visualizzazione informazioni sui clienti

1.1.3.2. Visualizzazione informazioni sui driver

1.2. Server

1.2.1. Sviluppo DB

1.2.1.1. Tabella clienti

1.2.1.2. Tabella driver

1.2.1.3. Tabella carte di credito

1.2.1.4. Tabella conti di appoggio

1.2.2. Scrittura nel DB

1.2.2.1. Scrittura clienti

1.2.2.2. Scrittura driver

1.2.2.3. Scrittura carte di credito (crittografando)

1.2.2.4. Scrittura conti di appoggio

1.2.3. Lettura dal DB

1.2.3.1. Lettura clienti

1.2.3.2. Lettura driver

2. Sistema di selezione del veicolo

2.1. Client

2.1.1. Sviluppo interfaccia di selezione veicolo

2.1.1.1. Visualizzare i veicoli per classe ecologica (endotermico, ibrido, elettrico)

2.1.1.2. Visualizzare i veicoli per prestigio

2.1.1.3. Visualizzare il numero di posti di ciascun veicolo

2.1.1.4. Visualizzare la tariffa al km di ogni veicolo

2.1.2. Integrazione con il sistema di prenotazione

2.2. Server

2.2.1. Lettura dal DB

2.2.1.1. Lettura veicoli e informazioni relative dal DB

3. Sistema di localizzazione driver

3.1. Client

3.1.1. Sviluppo interfaccia di localizzazione del driver (clienti e backoffice)

3.1.1.1. Visualizzazione della mappa

3.1.1.2. Visualizzazione del driver all'interno della mappa in tempo reale

3.1.1.3. Visualizzazione del tempo di attesa di arrivo stimato

3.1.2. Integrazione con il sistema di prenotazione

3.2. Server

3.2.1. Sviluppo DB

3.2.1.1. Tabella posizioni dei driver

3.2.2. Scrittura nel DB

3.2.2.1. Scrittura frequente delle posizioni dei driver

3.2.2.2. Scrittura storico posizioni per la prenotazione

3.2.3. Integrazione con il DB delle prenotazioni

3.2.4. Localizzazione del driver più vicino

4. Sistema di prenotazione

4.1. Client

4.1.1. Sviluppo interfaccia di prenotazione

4.1.1.1. Inserimento della destinazione sia dalla mappa o dall'indirizzo

4.1.1.2. Visualizzazione costo della corsa

4.1.1.3. Visualizzazione tempo della corsa

4.1.1.4. Visualizzazione modello e targa del veicolo della corsa

4.1.1.5. Notifica all'arrivo del driver

4.1.1.6. Possibilità di annullare la prenotazione entro 30 secondi

4.1.2. Integrazione sistema di selezione veicolo

4.1.3. Integrazione sistema di localizzazione driver

4.1.4. Integrazione sistema di pagamento

4.1.5. Sviluppo interfaccia di gestione prenotazioni dei driver.

4.1.5.1. Gestione delle prenotazioni (accettare o rifiutare)

4.1.5.2. Visualizzazione della posizione del cliente

4.1.6. Sviluppo interfaccia di contatto tra cliente e driver

4.1.6.1. Visualizzare numero di telefono del driver/cliente

4.2. Server

4.2.1. Sviluppo DB

4.2.1.1. Tabella prenotazioni

4.2.2. Scrittura DB

4.2.2.1. Scrittura prenotazioni nel DB

4.2.3. Integrazione sistema di instradamento

4.2.4. Localizzazione della posizione del cliente

5. Sistema di instradamento

5.1. Client

5.1.1. Sviluppo interfaccia di instradamento

5.1.1.1. Visualizzazione del percorso ottimale tra driver e cliente

5.1.1.2. Visualizzazione del percorso ottimale tra cliente e destinazione

5.1.1.3. Visualizzare i km rimanenti

5.1.1.4. Visualizzare il tempo rimanente

5.1.2. Integrazione sistema di prenotazione

5.2. Server

5.2.1. Calcolo percorso ottimale

5.2.1.1. Calcolo percorso minimo tra sorgente e destinazione

5.2.1.2. Calcolo del tempo rimanente

5.2.2. Integrazione DB prenotazione

6. Sistema di feedback

6.1. Client

6.1.1. Sviluppo interfaccia di feedback lato driver

6.1.1.1. Inserire un feedback facoltativo per il cliente

6.1.1.2. Aggiungere una nota di demerito in caso di comportamento negativo

6.1.1.3. Notificare il cliente in caso di feedback

6.1.2. Sviluppo interfaccia di feedback lato cliente

6.1.2.1. Inserire un feedback facoltativo per il driver

6.1.2.2. Aggiungere una nota di demerito in caso di comportamento negativo

6.1.2.3. Notificare il driver in caso di feedback

6.1.3. Sviluppo interfaccia di controllo feedback back office

- 6.1.3.1. Controllo numero di note di demerito clienti/driver
- 6.1.3.2. Possibilità di aumentare la tariffa per il cliente
- 6.1.3.3. Possibilità di aumentare la trattenuta per il driver
- 6.1.3.4. Possibilità di bannare il cliente dal sistema
- 6.1.3.5. Possibilità di daspo del driver dal sistema

6.2. Server

6.2.1. Sviluppo DB

6.2.1.1. Tabella feedback

6.2.2. Scrittura nel DB

6.2.2.1. Scrittura feedback e note

6.2.3. Aggiornamento nel DB

6.2.3.1. Aggiornamento tariffa utente

6.2.3.2. Aggiornamento trattenuta driver

6.2.3.3. Aggiornamento ban utente

6.2.3.4. Aggiornamento daspo utente

6.2.4. Lettura dal DB

6.2.4.1. Lettura feedback e note

6.2.5. Integrazione DB prenotazioni

7. Sistema di pagamento

7.1. Client

7.1.1. Sviluppo interfaccia di pagamento

7.1.1.1. Visualizzazione dei km effettuati

7.1.1.2. Visualizzazione dell'importo

7.1.1.3. Pagamento

7.1.1.4. Notifica di pagamento al driver

7.1.2. Integrazione sistema di prenotazione

7.1.3. Integrazione sistema di scontistica

7.2. Server

- 7.2.1. Sviluppo DB
 - 7.2.1.1. Tabella pagamenti
- 7.2.2. Scrittura nel DB
 - 7.2.2.1. Scrittura pagamenti nel DB
- 7.2.3. Servizio di pagamento di MoveMe
 - 7.2.3.1. Addebito dalla carta del cliente
 - 7.2.3.2. Accredito al conto del driver
 - 7.2.3.3. Calcolo della trattenuta
- 7.2.4. Integrazione DB prenotazioni
- 7.2.5. Integrazione DB di scontistica

8. Sistema di scontistica

8.1. Client

- 8.1.1. Sviluppo interfaccia di visualizzazione sconto
 - 8.1.1.1. Visualizzare lo sconto attuale
 - 8.1.1.2. Visualizzare i km rimanenti per il successivo sconto (se presente)

8.2. Server

- 8.2.1. Sviluppo DB
 - 8.2.1.1. Tabella sconti
- 8.2.2. Scrittura nel DB
 - 8.2.2.1. Scrittura sconti
- 8.2.3. Lettura dal DB
 - 8.2.3.1. Lettura sconti
 - 8.2.3.2. Lettura km rimanenti
- 8.2.4. Calcolo sconto
 - 8.2.4.1. Calcolo sconto per i km effettuati
 - 8.2.4.2. Aumentare lo sconto del 5% ogni 50km fino al 30%

Oltre alle attività elencate nella WBS, vanno considerate le attività relative al testing del sistema e di tutti i vari sottosistemi presenti all'interno. Come ultima attività va rilasciata l'applicazione sugli store dando per assodato che la qualità del codice sia tale da non ricevere un diniego.

3.3 Gestione dei costi

In questa sezione, si affronteranno le modalità utilizzate per gestire i costi relativi al progetto. Durante la fase di pianificazione, verrà eseguita la procedura di Cost Budgeting, in cui verrà assegnato un costo provvisorio a ogni attività nella WBS. Queste stime saranno affinate durante la pianificazione di uno sprint, quando sarà possibile avere una migliore comprensione delle attività da svolgere. Il project manager, mediante queste operazioni di stima, ha verificato che il budget proposto dal CEO di SPOSTAMI sia sufficiente per coprire tutte le spese del progetto. Saranno quindi prese in considerazione le risorse necessarie, le attività pianificate e i relativi costi al fine di garantire la sostenibilità finanziaria del progetto.

3.3.1 Stima durata e quantità del lavoro

Per stimare la durata del lavoro e la quantità di lavoro necessaria, sono state utilizzate tecniche basate sul consenso. Il project manager, insieme al core team di sviluppo, ha condotto iterazioni per determinare la mole di lavoro per le diverse attività, prendendo come riferimento i task presenti nella WBS. Una volta raggiunto un consenso, le stime ottenute sono state registrate e successivamente considerate durante le fasi di sprint planning per gestire la distribuzione delle risorse nelle attività. La stima della complessità della WBS ha consentito al project manager di valutare le risorse necessarie per completare il progetto. Nella sezione successiva, verranno prese decisioni

relative al flusso di cassa, pertanto questa attività assume un ruolo cruciale per evitare errori che potrebbero compromettere il successo del progetto. Sarà quindi fondamentale gestire accuratamente le stime di durata e quantità di lavoro per garantire una corretta allocazione delle risorse e mantenere il progetto finanziariamente sostenibile.

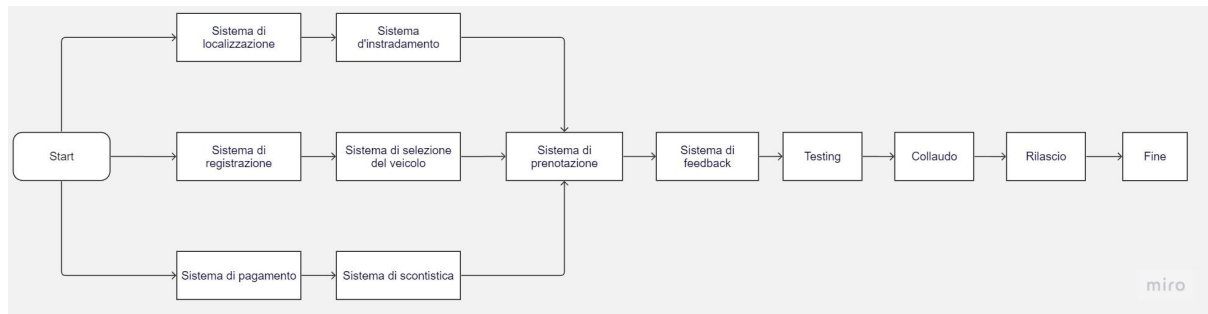
3.3.2 Cash flow

Per la gestione del flusso di cassa, il project manager ha concordato con il CEO l'adozione di un contratto a consuntivo. Dopo il completamento di due sprint, il prodotto sarà consegnato agli stakeholder per ottenere un feedback immediato sul lavoro svolto e garantire un flusso costante. Questo approccio è particolarmente adatto per la gestione dei costi di un progetto condotto in modo iterativo, poiché consente al cliente di comprendere lo sviluppo dell'intero progetto nel tempo. L'erogazione graduale del prodotto e la possibilità di ottenere feedback tempestivi permettono di adattare le attività successive in base alle esigenze e alle priorità emerse durante il processo di sviluppo. Ciò contribuisce a garantire una gestione finanziaria efficace e una maggiore soddisfazione del cliente.

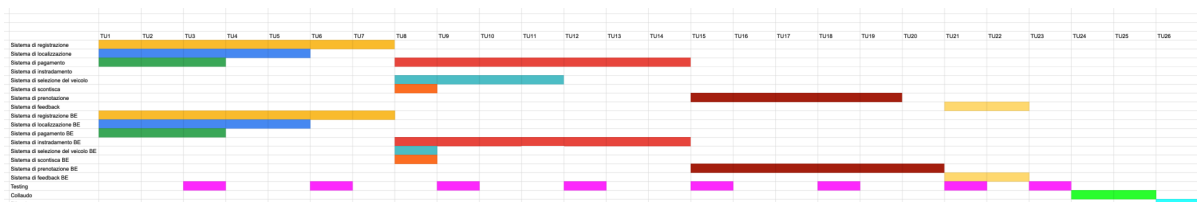
3.3.3 Project network diagram

All'interno di questa sezione si cercherà di delineare, in base a quanto stabilito nella sezione di stima del lavoro, uno schema temporale in grado di descrivere quali attività dovranno essere completate entro quale periodo. Utilizzando un approccio di tipo iterativo non sarà possibile effettuare in fase iniziale un

diagramma che definisca con esattezza una schedula delle attività. Si ritiene comunque opportuno derivare quali macro attività potranno essere effettuate e con quale ordine, per poi poter definire il Gantt.



Il diagramma rappresenta sulle righe le attività, e sulle colonne le unità di tempo per sprint bisettimanali, definite time unit (TU). Ogni TU sarà l'equivalente di 7 giorni lavorativi circa. Un'attività per lo sviluppo richiede tante TU quanti sono i quadratini associati ad essa. Le attività non potranno variare il loro ordine di esecuzione in quanto esso è stato definito nel diagramma precedente. Il diagramma Gantt svolgerà un ruolo fondamentale nella effettiva allocazione delle risorse, dando la possibilità di individuare i path critici, ed assegnando a loro un numero maggiore di risorse, dove possibile, riducendo le fasi nelle quali verrà eseguita una unica attività, favorendo al massimo il parallelismo.



3.4 Agile sprint

Per lo sviluppo del progetto si è utilizzato il framework Scrum. Questa sezione sarà dedicata alla pianificazione dello sprint iniziale del progetto, i cui task faranno fede alla WBS. Con il susseguirsi degli sprint la gerarchia WBS verrà arricchita, aggiungendo ad ogni pianificazione per ogni sprint i nuovi task da sviluppare che non sono stati considerati durante la pianificazione iniziale, aggiungendo nuovi layer alla gerarchia prodotta, in maniera tale da arrivare agli sprint finali con la WBS completa in ogni dettaglio.

Capitolo 4

Launching

Durante questa fase le risorse che verranno maggiormente coinvolte saranno gli sviluppatori che si approcceranno al progetto. Il project manager dovrà completare alcune formalità tralasciate durante la fase di planning. Saranno necessarie attività di :

- Selezione del team di sviluppo
- Definizione dei pattern di sviluppo
- Fornitura delle risorse a corredo del progetto

Completata questa fase inizierà la fase di sviluppo del progetto, facendo riferimento alle strutture prodotte in fase di planning.

4.1 Developers

In questa sezione verranno forniti dettagli su come è avvenuta la gestione del team da parte della software house. La software house incorpora già developers con seniority variabile da junior a senior passando per regular e middle, al termine della fase di selezione il team risulta così composto:

- **PO (Product Owner)** responsabile del delivery del progetto, sarà suo compito monitorare l'andamento e verificare la velocity di sviluppo dei task
- **Sviluppatori** il team di sviluppo sarà composto da 6 sviluppatori tra cui due senior (core team) e 4 junior allo scopo di fornire a questi ultimi una formazione in merito.
- **Analista Funzionale** responsabile della stesura delle storie, preparazione dei planning e all'occorrenza scrum master durante la fase di stima degli sprint, stima effettuata con modello scrum poker secondo la serie di Fibonacci

Verrà eseguita in maniera iterativa ad ogni sprint planning la gestione delle risorse. Si terrà sempre in considerazione il diagramma Gantt sviluppato in fase di planning che descriverà la distribuzione del lavoro nell'arco temporale.

4.2 Modello organizzativo

Tutto il progetto seguirà un approccio Scrum che quindi prevede la gestione del progetto a sprint bisettimanali, preceduti da uno sprint planning con relativa stima delle storie in story point, da svilupparsi e una retrospettiva finale con l'obiettivo di

espletare gli aspetti negativi/positivi dello sprint appena ultimato.

4.3 Variazioni in corso d'opera

Il project manager ha adottato una modalità per gestire i cambiamenti di scope del progetto, in conformità agli standard del settore. Grazie all'approccio iterativo del PMLC, è possibile apportare modifiche dinamiche allo scope del progetto mediante la pianificazione regolare degli sprint e l'aggiunta o la modifica delle attività nella WBS. Questo rende più facile affrontare i cambiamenti di scope rispetto a un progetto sviluppato in modo lineare. Tuttavia, è previsto che le eventuali richieste di cambio che emergono durante lo sviluppo saranno valutate e implementate alla fine degli sprint pianificati, in modo da non influire sullo sviluppo in corso dell'applicazione e non ritardare la consegna.

4.4 Work package

Il project manager ha previsto lo sviluppo di un Work Package per monitorare i task ad alto rischio nel progetto. Sono stati identificati i seguenti task:

1. Sviluppo del client e del server: Questi task richiedono una significativa allocazione di risorse umane in quanto fanno parte dei percorsi critici del progetto. Saranno necessarie risorse altamente esperte per garantire prestazioni ottimali. Il core team si occuperà di portare avanti questi task, ma saranno affiancati anche da sviluppatori junior per ridurre i tempi di sviluppo.

2. Integrazione delle componenti: Una volta completati i task di sviluppo del client e del server, si procederà con l'integrazione delle diverse componenti. Sarà fondamentale una buona progettazione e l'utilizzo di tecniche di Continuous Integration e Continuous Delivery.

3. Testing: Parallelamente all'integrazione dei moduli, si avvieranno i task relativi al testing. Saranno utilizzati ambienti di test sviluppati in parallelo al client e server, realizzati dai sviluppatori junior con competenze in ingegneria del software.

4. Fase di collaudo: Una volta completata la fase di testing, si entrerà nella fase di collaudo, dove potrebbe essere necessario un cambio di risorse. Il team di sviluppo avrà un ruolo fondamentale solo nella fase iniziale di installazione.

Questo approccio permette di gestire in modo efficace le risorse e di affrontare le sfide più complesse del progetto, garantendo un controllo adeguato e un progresso continuo.

Capitolo 5

Monitoring and Control

In questa sezione verranno considerati diversi risk drivers e verranno delineate diverse strategie per la loro gestione.

5.1 Sviluppo e Report

L'utilizzo di un PMLC iterativo, in accoppiata con la Continuous delivery garantirà al project di poter tenere sotto controllo l'avanzamento del progetto mediante release mensili effettuate in modalità di cumulative reports.

5.2 Controllo del budget

Anche considerando sufficiente il budget stanziato per lo sviluppo del progetto totale si cercherà di ottimizzare le risorse e i relativi costi sempre cercando di ottenere il miglior prodotto possibile che segua i più attuali standard di produzione del software

5.3 Meeting e Sal

Gli stand-up meeting saranno introdotti fin dal primo giorno del progetto per garantire un adeguato tracciamento dello sviluppo del codice. Questi meeting avranno luogo ogni giorno lavorativo alle 9:10 e avranno uno scopo sintetico e informale. Durante lo

stand-up meeting, i membri del team descriveranno brevemente le attività svolte nella giornata lavorativa precedente e quelle pianificate per la giornata odierna.

La partecipazione agli stand-up è obbligatoria, ma nel caso in cui ci siano ritardi dei trasporti o problemi di collegamento per i membri del team che lavorano da remoto, non sarà formalmente obbligatoria. Tuttavia, è comunque fortemente consigliato partecipare per garantire una comunicazione efficace e un coordinamento adeguato tra i membri del team.

Durante ogni stand-up meeting, verrà prodotta una minuta che riassume le principali discussioni e attività svolte. Questo documento servirà a creare uno storico delle attività svolte nel corso del progetto. Inoltre, gli stand-up meeting serviranno anche come forum per discutere eventuali problematiche comuni che riguardano l'intero team. Tuttavia, se ci sono problematiche individuali, queste verranno affrontate in modo separato e offline, al fine di evitare di allontanarsi dallo scopo principale dello stand-up.

Per garantire un focus adeguato durante gli stand-up meeting, sarà introdotta la figura del "Buzz Man". Questa persona, scelta a campione all'interno del team, avrà il compito di dichiarare il "BUZZ" nel caso in cui si discutono argomenti non rilevanti per l'intero team. In tal caso, la discussione verrà interrotta e l'argomento sarà trattato offline solo con i membri interessati.

In questo modo, gli stand-up meeting saranno un'importante pratica per mantenere il team allineato, condividere informazioni importanti e affrontare le problematiche comuni,

senza dilungarsi in discussioni non rilevanti per tutti i membri del team.

5.4 Tracking dei Bug

Durante gli sviluppi sarà possibile tracciare gli eventuali bug riscontrati, a tal proposito si utilizzerà un formalismo comune per la stesura non ambigua delle issues il tutto gestito tramite l'apposita sezione di issues tracking fornita del software di project management.

5.5 Retrospettiva

La retrospettiva è l'istituzione di un momento di riflessione a termine degli sprint dove i componenti del team hanno il diritto di esporre cosa ha funzionato, cosa è necessario mantenere e cosa si vorrebbe cambiasse nello sprint successivo. Lo scopo di questo momento di condivisione e relativa minuta è quello di cercare di migliorare il modello di sviluppo del software nonchè cercare di disinnescare eventuali dissapori che potrebbero essere nati tra team di sviluppo e altre figure accessorie al progetto.

Capitolo 6

Closing

In questa fase, verranno definite le procedure che porteranno alla conclusione del progetto. Sarà infatti necessario completare i task finali per l'effettiva messa in produzione del software. Si avrà una fase iniziale nella quale il sistema verrà eseguito in modalità di collaudo.

6.1 Collaudo

Durante la fase di collaudo, il nuovo sistema sarà eseguito in una piccola e specifica area della città di Milano, destinato a una clientela limitata. Prima dell'avvio del collaudo, si terrà un incontro dedicato alla formazione dei dipendenti di MoveMe, al fine di prepararli all'utilizzo del sistema.

È stata identificata una durata precisa per il periodo di collaudo, ma si prevede di affrontarlo in modo iterativo. Inizialmente, si stima che il sistema richiederà alcuni aggiustamenti e ottimizzazioni per funzionare correttamente. Tuttavia, nel corso del tempo, si prevede che raggiungerà uno stato di auto-mantenimento, dove non saranno necessari interventi tecnici frequenti.

Nonostante ciò, è previsto che una percentuale del team di sviluppo rimarrà a disposizione per affrontare eventuali problemi tecnici o bug che potrebbero emergere durante il periodo di collaudo. Questo garantisce un supporto continuo per il corretto funzionamento del sistema e per risolvere eventuali inconvenienti che possano sorgere.

In sintesi, il periodo di collaudo del sistema sarà gestito in modo iterativo, con una fase iniziale di aggiustamenti seguita da un progressivo stato di auto-mantenimento. Il team di sviluppo rimarrà disponibile per affrontare eventuali problemi tecnici o bug che potrebbero richiedere interventi durante il collaudo.

6.2 Rilascio

Ultimata la fase di collaudo quindi garantito che il sistema sia stabile sarà effettuato l'effettivo rilascio sulle piattaforme pubbliche delle app sviluppate.

6.3 Accettazione del committente

Coincide con la definizione formale del fine progetto quindi si assume che il progetto ha soddisfatto tutti i requisiti inizialmente esposti. La fase di accettazione prevede anche un accordo per la definizione delle attività operative che potrebbero essere effettuate una volta consegnato il progetto. Tra queste possiamo considerare la notifica e conseguente correzione di eventuali bug o l'aggiornamento per cause di sicurezza delle librerie usate, per quanto riguarda la modifica di determinate parti dell'applicazione ed eventualmente anche l'aggiunta di nuove funzionalità queste saranno trattate come CR quindi sottoposte a preventivo e stima da parte del team di sviluppo.

6.4 Documentazione di prodotto

L'obiettivo di questa fase sarà consegnare la documentazione annessa al software prodotto.