



**UNIVERSITÀ
DI TRENTO**

**Dipartimento di
Ingegneria e Scienza dell'Informazione**

Progetto: TRENTUS

Report finale

INDICE

1. Organizzazione del lavoro.....2

2. Ruoli e attività.....4

3. Carico e distribuzione del lavoro.....6

4. Criticità.....7

5. Autovalutazione.....8

1. Organizzazione del lavoro

Il lavoro è stato pianificato provando ad applicare le metodologie agile viste in classe.

Abbiamo inizialmente provato ad applicare la metodologia kanban, creando un profilo su trello.com. Non ci siamo però trovati bene con questa metodologia, per due ragioni:

- avendo un team piccolo ed un progetto che inizialmente era di dimensioni ridotte, avremmo passato più tempo a programmare ed elaborare la tabella che a lavorare effettivamente
- Molti dei lavori che avevamo erano ancora di documentazione e di brainstorming (D1) su come elaborare il sito, vari grafici, e stesure di idee in cui collaboravamo tutti quanti, trovandoci spesso in biblioteca o dopo lezione, e non applicando una netta divisione degli incarichi, ma solo concentrandoci su certi aspetti.

Nonostante questo abbiamo tentato di implementare nella progettazione del nostro lavoro alcune tecniche della metodologia Kanban:

- In primo luogo limitare il Work In Progress. Nella stesura dei documenti iniziale c'è sempre stata una forte collaborazione, ma ognuno di noi tentava di concentrarsi su un solo aspetto della documentazione alla volta, come Giovanni e Giuseppe ad elaborare la parte grafica e Luca la parte più verbosa, mantenendo comunque acceso il dialogo per avere tutto coerente.
- In secondo luogo mantenere un flow continuo, non pianificando alcuna deadline, ma cercando di mantenere una produzione costante, aggiornando gli altri incontrandosi.

Arrivato però il momento di mettere le mani al codice, abbiamo iniziato ad applicare anche metodologie agile identificabili nello SCRUM:

- Abbiamo definito dei ruoli, vedi capitolo successivo, ma in breve Luca lato server, Giuseppe lato client, e Giovanni il ponte tra i due.
- Abbiamo identificato delle priorità, inizialmente impostare gli strumenti per creare la WebApp, impostare il database, e fornire una base server. Successivamente le priorità sono state identificate come gestione Attori > gestione Servizi > gestione Segnalazioni > gestione Avvisi > gestione Sondaggi.
- Ci siamo concentrati sulla produzione di frequenti incrementi software, questo è rispecchiato dall'approccio orientato alle features nella gestione del git. Tutto questo mantenendo sempre un limite al WIP, tentando di concentrarci su una funzionalità alla volta, rendendola funzionante.
- L'adattabilità ai cambiamenti è stata fondamentale! Ad esempio molto spesso lo swagger è stato riadattato, facendo fronte ad imprevisti tecnici del back-end o all'implementazione di questi nel front-end. Anche la documentazione è variata molto nel corso del progetto, specialmente il D1
- Parlando di questo, la costante attività di documentazione e testing è stata cruciale, specialmente nel back-end, per fornire al front-end un supporto stabile e verificare il funzionamento dell'API

Riassumendo quindi, l'approccio che abbiamo avuto è stato ispirato:

- dal Kanban come scorrimento del lavoro, rivolto verso un flow di produzione continuo, delineando compiti da svolgere ad ogni meeting, senza utilizzare tabelle o cose simili, ma al massimo appuntandoli sulle issues di github.
- dallo Scrum come idee e metodologie sopra elencate, senza utilizzare l'idea degli sprint, non avendo meeting giornalieri e backlog ma secondo necessità.

Abbiamo avuto vari incontri durante il periodo D2/D3, a volte uno ogni due settimane, altre volte due o tre alla settimana, che non rappresentavano una deadline, bensì un momento per controllare i progressi e confrontarci, delineare punti critici nell'andamento del progetto, e pianificare le azioni successive. Questa scelta è principalmente dovuta al fatto che ognuno di noi ha impegni, soprattutto a livello accademico. Si può infatti notare un calo di commit su git attorno ai periodi in cui abbiamo avuto parziali o progetti. C'è stato quindi un "adattamento" rispetto ai bisogni individuali delle persone nel gruppo

Per facilitare la collaborazione tra di noi abbiamo utilizzato principalmente github per la programmazione e google docs per la redazione dei documenti, grazie al suo processare le modifiche in tempo reale. Questo è tornato utile anche su draw.io per la creazione di vari grafici e su figma per il mockup.

Per tenerci in contatto abbiamo creato un gruppo telegram, tornato utile specialmente per fare domande e richiedere piccole modifiche. Alcuni meeting, quelli più brevi, li abbiamo tenuti per comodità su discord.

2. Ruoli e attività

Per tenere traccia delle attività di ognuno di noi abbiamo utilizzato un foglio excel come log. Abbiamo trascritto orari di lavoro ed attività svolte. In seguito riassumiamo i compiti svolti da ognuno di noi nel progetto

Giovanni Sbalchiero (Leader)

- **Progettista/architetto del software**
- **UI/UX designer**
- **Sviluppatore front-end principale**
- **Sviluppatore back-end secondario**
- **Tester**

All'inizio del progetto Giovanni si è occupato, assieme a Giuseppe, di ideare la parte grafica del software su Figma, cioè il mockup, e gestendo alcuni schemi per i documenti, come le user stories, user flow.

Successivamente si è dedicato, con Luca, alla progettazione del database e dei relativi schemi per ogni entità della webapp, ha poi implementato parzialmente la parte di back-end dedicata all'autenticazione.

In seguito ha lavorato al front-end con Giuseppe, creando principalmente i componenti per le aree dedicate all'admin e al GdS, inoltre ha agito da mediatore tra front-end e back-end, implementando le chiamate alle API e dialogando con Luca per la rilevazione delle API necessarie.

Ha inoltre implementato alcuni fix minori del back-end in contemporanea allo sviluppo del front-end.

Si è poi occupato dell'individuazione dei componenti con relativa descrizione del D2, e del deploy della web app.

Infine ha collaborato con gli altri membri del team nella revisione di tutti i deliverable.

Giuseppe Conte

- **Analista di sistema**
- **Responsabile del deploy**
- **UI/UX designer**
- **Sviluppatore front-end principale**
- **Redactor**

All'inizio del progetto Giuseppe si è occupato, assieme a Giovanni, di ideare la parte grafica del software su Figma, cioè il mockup, e gestendo alcuni schemi per i documenti, come le user stories e user flow.

Successivamente si è dedicato alla finalizzazione del D1, mettendo per iscritto il mockup, ed al D2, elaborando gli use case.

Giuseppe si è poi preso carico del front end, implementando i grafici e curando la parte stilistica e formale della piattaforma.

Infine, ha lavorato alla stesura del D3, concentrandosi sugli User Flow, sul Front-End e sul Deployment.

Luca Da Riz

- **Progettista/architetto del software**
- **Sviluppatore back-end principale**
- **Technical writer**
- **Tester**

All'inizio del progetto, Luca si è dedicato alla descrizione del progetto D1, creando un vocabolario di progetto, ed elaborando le idee del gruppo, definendo per iscritto lo scopo di Trentus, vantaggi e criticità, utenti e ruoli, e requisiti funzionali e non.

Successivamente si è dedicato, assieme a Giovanni, alla progettazione del database e dei relativi schemi per ogni entità della webapp.

Luca si è poi preso carico del back-end, rilevando le API necessarie con Giovanni e poi elaborando uno swagger, che ha aggiornato e mantenuto per il resto del progetto. Ha creato e strutturato il backend usando node.js ed express, gestendo la comunicazione con il database con mongoose, producendo tutte le api necessarie.

Ha testato il funzionamento delle API creando un ambiente su insomnia. Sotto necessità dei front-end developers ha applicato modifiche al back-end.

Infine, ha lavorato alla stesura del D2, descrivendo il diagramma delle classi ed il passaggio alle API, alla stesura del D3, componendo principalmente la parte delle Web API, del database e del testing, ed al D4.

3. Carico e distribuzione del lavoro

	D1	D2	D3	D4	TOT
Giovanni Sbalchiero	28	25	140	3	~196
Giuseppe Conte	36	14	113	3	~166
Luca Da Riz	16	42	138	10	~206
TOT	80	81	391	16	~568

Poiché nel nostro log non abbiamo diviso le ore in deliverables, ma per azione svolta, distinguiamo qui sotto come abbiamo calcolato le ore.

Il D1 considera il tempo che ci abbiamo messo a ideare il progetto, metterlo per iscritto e per grafica, ed anche alcune delle ore che abbiamo impiegato per la creazione delle slides da presentare al comune, che comunque sono tornate utili. Qui Giovanni e Giuseppe hanno più ore. Questo è dovuto alla creazione mockup su Figma, processo oneroso, soprattutto non avendolo mai usato prima.

Il D2 considera il tempo che ci abbiamo messo ad architettare il backend, individuando i componenti e le classi, elaborare lo swagger, oltre a creare i grafici necessari al documento. Qui Luca ha più ore, questo è dovuto al suo ruolo da gestore del lato server, tenendo conto anche dei vari cambiamenti progettuali che il back-end ha affrontato e le varie riscritture dello swagger.

Il D3 considera il tempo dedito alla vera e propria programmazione codice della WebApp, back-end di Luca e front-end di Giovanni e Giuseppe, e dalla redazione del documento: dalle user stories, alla realizzazione vera e propria delle API, fino all'implementazione e deployment. Queste ore sarebbero sicuramente potute essere ridotte, ma noi tutti abbiamo messo le mani su questi programmi e linguaggi per la prima volta, ed abbiamo anche dovuto apprendere il loro funzionamento.

Il D4 considera la stesura del documento ed il confronto di gruppo per scriverlo.

La distribuzione delle ore è abbastanza sparsa, ma questo è corretto, e dovuto al fatto che ognuno di noi ha intrapreso ruoli diversi, che rispecchiano necessità diverse in momenti diversi della progettazione. Dal totale delle ore per ogni persona si può vedere come tutti abbiano lavorato duramente. In particolare, essendoci tutti concentrati molto sulla parte di sviluppo codice, la maggior parte delle ore viene concentrata nel D3.

Nota: ognuno ha stilato nel documento le proprie ore, divise secondo i criteri sopraelencati, e si è confrontato con gli altri colleghi del gruppo.

Questa stima è stata quindi condivisa e approvata

4. Criticità

Nel lavoro di gruppo non abbiamo incontrato criticità. Si è venuto a formare un team ben coeso ed unito. Le criticità riscontrate riguardano per lo più il lato implementativo della WebApp. Questo è principalmente dovuto al fatto che il progetto era partito in maniera molto ambiziosa, prevedendo diverse funzioni.

Non avendo esperienza con i linguaggi e le tecnologie utilizzate, abbiamo deciso di scartarne alcune più avanzate che all'inizio avevamo previsto nel nostro sistema, per difficoltà nell'implementazione.

Tra queste feature scartate troviamo:

- **Ricerca del servizio tramite casella di testo:**

Non siamo riusciti a implementare questa funzione a causa di come abbiamo gestito inizialmente lo store e i componenti di visualizzazione dei servizi. Infatti, in seguito la nostra implementazione si è rivelata di difficile gestione in questo specifico campo, e rifare tutto con le conoscenze apprese avrebbe richiesto troppo tempo.

- **Email:**

Abbiamo riscontrato dei problemi nell'invio automatico delle e-mail alla pubblicazione di un avviso, poiché è necessario un server SMTP, ma non siamo riusciti a trovarne nessuno appropriato. Optando per un protocollo sendmail (come quello che usa php) abbiamo risolto il problema in locale, ma non siamo riusciti ad applicarlo nella parte di deployment.

- **Login con Google:**

Inizialmente questo componente era stato previsto, ma per gestire al meglio i vari ruoli del sistema, abbiamo preferito gestire l'autenticazione internamente.

- **Aggiornamento in tempo reale delle informazioni e notifiche push:**

Non c'è stato il tempo di approfondire le caratteristiche della comunicazione client-server IRT, di conseguenza abbiamo scartato questa feature.

5. Autovalutazione

Nel complesso abbiamo lavorato tutti con impegno e costanza nell'arco di tutto il progetto, salvo periodi di altri impegni didattici. Tra membri del gruppo c'è sempre stata una buona intesa ed un efficace teamwork. Non abbiamo mai avuto problemi nel ricoprire i ruoli assegnati e nessun problema nell'aiutarci a vicenda, imparando anche a uscire dalla propria comfort zone. C'è stata anche una fiducia reciproca nelle capacità di tutti, ognuno ha ampiamente soddisfatto le responsabilità che si è preso, per questo motivo, pensiamo di meritare tutti un voto pari.

	Voto
Giovanni Sbalchiero	29
Giuseppe Conte	29
Luca Da Riz	29