Documento: BookEasy – “Antico Granaio” – Report Finale

Revisione: 0.2

Dipartimento di Ingegneria e Scienza dell’Informazione

Progetto:

BookEasy – “Antico Granaio”

Titolo del documento:

Report Finale

Informazioni Documento

Nome Doc.	D5-BookEasy-AnticoGranaio_ReportFinale	**Numero Doc.**	RF5
Descrizione	Report finale del progetto: organizzazione del lavoro, ruoli, tempo complessivo e di ciascun membro dedicato al progetto, criticità, autovalutazione.		

FARE IL SOMMARIO!!!!!!!

1. Approcci all’Ingegneria del software

a. BlueTensor

Nella fase iniziale del seminario tenuto in collaborazione con BlueTensor, il relatore Jonni Malacarne ha illustrato le caratteristiche distintive della sua azienda. In particolare, BlueTensor si dedica a fornire servizi basati sull'intelligenza artificiale ad altre aziende. Un esempio concreto di tali servizi è rappresentato da un sistema di riconoscimento delle imperfezioni sviluppato per un'azienda specializzata nella produzione di pneumatici.

Nella seconda parte il relatore ha fornito una dettagliata illustrazione del processo di sviluppo di un sistema. Tale processo inizia con un'analisi di prefattibilità, generalmente condotta nel corso di un periodo di 1-5 giorni. Successivamente, si procede con la fase preliminare che comprende l'analisi di fattibilità, la realizzazione di un POC (Proof Of Concept) e la creazione di un MVP (Minimum Viable Product).

Le fasi appena elencate rivestono un’importanza cruciale sia per l'azienda BlueTensor che per l'azienda cliente. Esse mirano a stabilire le funzionalità del sistema, consentono di valutare la fattibilità e stipulare il contratto.

Una volta completate queste fasi, inizia la fase di Engineering, che consiste nei seguenti passaggi:

1. **Documento dei requisiti:** in questa fase, vengono delineate in modo dettagliato le funzionalità principali del sistema e le funzionalità accessorie;
2. **Specifica dei requisiti:** questo documento espone le User stories, i casi d'uso e la specifica dei singoli componenti del sistema;
3. **Raccolta dati:** durante questa fase, è essenziale raccogliere dataset che saranno successivamente utilizzati per addestrare l'intelligenza artificiale;
4. **Fase di rilascio:** in questa fase, BlueTensor adotta il metodo "Agile", caratterizzato dal rilascio progressivo di diverse parti del sistema. Questo approccio consente di acquisire feedback continuativi e di effettuare revisioni nel tempo.

In sintesi, riguardo a BlueTensor, sebbene questo metodo si dimostri efficace nel migliorare costantemente la valutazione dell'IA, va sottolineato che può rendere più complesso stimare preventivamente il costo e la quantità di lavoro.

Approccio utilizzato da noi in relazione con quello esposto nel seminario:

All’interno del nostro gruppo, questo seminario è risultato particolarmente utile, poiché ci ha permesso di approfondire le diverse fasi dello sviluppo del software e di cogliere appieno le forti interconnessioni tra di esse. Inoltre, il seminario ha sottolineato l’importanza delle User stories, che offrono agli sviluppatori una prospettiva più completa sulle diverse opzioni da considerare durante l’utilizzo del software.

b. Metodo Kanban

Durante il seminario condotto da York Roessler sul metodo Kanban, noi studenti abbiamo partecipato a una simulazione mirata a comprendere al meglio il funzionamento di tale metodo.

La simulazione, svolta in gruppi, prevedeva che ogni studente avesse diverse "opzioni" da sviluppare, le quali dovevano attraversare le fasi di design, implementazione e completamento. Ogni giorno, in base all'esito del lancio di una moneta, ogni studente poteva continuare un lavoro già esistente o iniziarne uno nuovo. Dopo 7 giorni nella simulazione, è emerso che la maggior parte delle opzioni rimaneva bloccata nella fase di design, mentre ne venivano continuamente avviate di nuove.

Dopo un momento di discussione, è stato imposto un limite massimo al numero di opzioni in fase di design e implementazione, chiamati Work In Progress Limits.

Questa decisione si è rivelata estremamente efficace, in quanto ha prodotto, oltre ad una maggiore coesione e collaborazione all'interno del gruppo, un miglioramento drastico nel numero di opzioni completate.

Nell'analisi statistica degli intervalli di tempo necessari per la conclusione di un'opzione, il metodo Kanban si è dimostrato un efficace strumento per predire la durata di un progetto e per mantenere la catena di lavoro stabile, evitando di sovraccaricarla di nuove opzioni.

Nei commenti finali, il relatore ha anche fatto notare come, in alcune giornate della simulazione, alcune persone non potessero fare nulla per migliorare la produttività complessiva. Questo approccio contrasta direttamente con l'idea classica di "lavoro", in cui il tempo di tutti i dipendenti deve essere completamente impiegato.

Approccio utilizzato da noi in relazione con quello esposto nel seminario:

Nelle attività del nostro lavoro, abbiamo applicato le conoscenze acquisite durante questo seminario evitando la stesura di diversi deliverable in contemporanea. Infatti, come dimostrato dalla simulazione del metodo Kanban, abbiamo riscontrato maggiore efficienza mantenendo lo sviluppo di poche caratteristiche o fasi “aperto”. Questo stesso approccio è stato adottato nello sviluppo del codice relativo al D4, dove Alessandro Busola ha concentrato l’attenzione su poche funzionalità contemporaneamente, privilegiando la qualità e la completezza rispetto alla quantità di feature avviate.

c. IBM

Durante il seminario tenuto da Ferdinando Gorga, rappresentante di IBM, è stato illustrato un approccio innovativo alla progettazione di software orientato al cloud.

Come sottolineato dal relatore, la piattaforma cloud di IBM offre al cliente l’opportunità di sfruttare macchine ad alte prestazioni a costi convenienti, garantendo allo stesso tempo un livello di sicurezza e di protezione estremamente elevato.

In particolare, il cloud IBM si distingue per offrire potenza computazionale attraverso quattro modalità: Bare Metal Servers, Virtual Machines, Containers e Serverless Code Engines.

I primi concedono al cliente un maggiore controllo, consentendogli di gestire direttamente le risorse, mentre gli ultimi forniscono una soluzione più automatizzata, permettendo di concentrarsi sulla scrittura del codice senza doversi preoccupare dell'implementazione dell'infrastruttura sottostante. Questa varietà di opzioni consente di adattare la soluzione alle specifiche esigenze e preferenze dell'utente, offrendo flessibilità e versatilità nel processo di sviluppo e gestione del software basato su cloud.

Approccio utilizzato da noi in relazione con quello esposto nel seminario:

Nella fase di pianificazione del nostro progetto, abbiamo ritenuto non necessario l’utilizzo di macchine ad alte prestazioni. Tale decisione è derivata dalla definizione dei requisiti di sistema, i quali indicano un numero limitato di utenti in simultanea e un utilizzo relativamente basso di memoria, ad esempio, per quanto riguarda la memorizzazione delle prenotazioni.

d. Meta

Durante il seminario tenuto da Heorhi Raik per conto di META, ci è stato presentato un approccio all'ingegneria del software molto più flessibile rispetto alla concezione "classica" dello sviluppo software. META, infatti, consente ai suoi sviluppatori di scegliere il gruppo in cui desiderano lavorare e offre la possibilità di cambiare team, promuovendo così un ambiente dinamico e adattabile. La flessibilità viene ulteriormente evidenziata dalla libertà concessa ai vari team di organizzarsi in base alle loro preferenze: all'interno di META, esistono team che conducono riunioni e meeting giornalieri e altri che comunicano principalmente attraverso documenti su Google Documents.

Un altro aspetto distintivo di META è la richiesta di capacità di adattamento a ruoli non strettamente correlati. Come sottolineato dal relatore, è accaduto in diversi casi che uno sviluppatore si sia trovato a ricoprire il ruolo di Data Scientist, imparando così nuove tecniche e acquisendo nuove conoscenze a riguardo in maniera efficace.

In sintesi, l'approccio di META all'ingegneria del software promuove la flessibilità, l'adattabilità e la diversificazione dei ruoli, creando un ambiente in cui gli sviluppatori sono liberi di esplorare e affrontare nuove sfide, contribuendo così alla crescita e alla versatilità del team.

Approccio utilizzato da noi in relazione con quello esposto nel seminario:

All’interno del nostro gruppo di lavoro, come illustrato nel capitolo “Organizzazione del lavoro”, abbiamo ottenuto notevoli vantaggi dall’utilizzo di Google Documents. Questo strumento ha consentito di comunicare efficacemente anche senza la necessità di incontri sincroni, adattandosi così alle esigenze e agli impegni di ciascun membro.

e. U-Hopper

Durante il seminario condotto da Daniele Miorandi per conto di U-Hopper, è stata presentata un’interessante modalità di organizzazione del lavoro che si distingue leggermente dagli approcci precedentemente esposti. All'interno di U-Hopper, si adotta l’utilizzo degli sprint, ovvero periodi di tempo relativamente brevi durante i quali un team si dedica allo sviluppo di una specifica funzionalità. La definizione di questi sprint è divisa in tre step:

- Descrizione dell'obiettivo;
- Specifica dei quesiti;
- Definizione della scadenza.

La definizione degli sprint avviene periodicamente, poiché è possibile che una data scadenza si estenda o si accorci in base alle circostanze.

Nel corso degli sprint, si susseguono ciclicamente le fasi di scrittura del codice, di testing e di revisione.

Riguardo alla fase di revisione, uno strumento molto utilizzato sono le merge requests, che facilitano lo spostamento del codice nelle singole branch per poi riunirlo nella main branch.

Per quanto riguarda i deployments, invece, le fasi sono 4:

- Deploy in produzione;
- Deploy in staging;
- Test;
- Rilascio.

Queste fasi consentono di testare le nuove funzionalità in diversi ambienti e di individuare eventuali problematiche prima del rilascio effettivo.

Approccio utilizzato da noi in relazione con quello esposto nel seminario:

Nel nostro gruppo, abbiamo sfruttato le conoscenze apprese da questo seminario nella programmazione delle release. Infatti, invece che adottare un approccio di rilascio continuo, abbiamo scelto di mantenere le diverse componenti completate nella fase di test, in modo da verificare il corretto funzionamento tra le varie parti.

f. RedHat

Nel seminario condotto da Mario Fusco riguardo al mondo dell'Open Source, è stato presentato un approccio allo sviluppo del software che si distingue per la sua apertura rispetto all’ambiente aziendale tradizionale. Il relatore, infatti, ha evidenziato le potenzialità del codice Open Source, sottolineando come, dal punto di vista della community, esso offra una qualità superiore grazie alla possibilità per qualsiasi sviluppatore di contribuire al miglioramento del lavoro altrui.

Mentre, dal punto di vista personale, i progressi effettuati su quel determinato codice possono essere un ottimo "biglietto da visita" per persone interessate a inserirsi in un ambiente aziendale che richiede questa tipologia di skills.

Tuttavia, è importante notare che l’Open Source presenta anche aspetti negativi. Uno di questi è rappresentato, non solo dalla sfida nel trarre profitto dalle proprie azioni, ma anche dalla necessità di sviluppare la giusta quantità di documentazione e ottenere le certificazioni per acquisire la proprietà di specifici file.

Approccio utilizzato da noi in relazione con quello esposto nel seminario:

Nonostante la rilevanza di questo seminario per prospettive di lavoro future, non abbiamo trovato modo di applicare quanto appreso all’interno del nostro lavoro.

g. Microsoft

Nel seminario tenuto da Diego Colombo per conto di Microsoft, è stato presentato un approccio all'ingegneria del software noto come Test-Driven Development (TDD). In questo metodo, i requisiti del software vengono trasformati in test case particolarmente specifici. Ciò consente agli sviluppatori di avere una comprensione molto più precisa del problema da risolvere e della soluzione da adottare, riducendo anche il rischio di "Over-engineering", ovvero lo sviluppo di un prodotto in maniera eccessivamente complessa.

Tuttavia, c’è un lato negativo: l’approccio TDD potrebbe andare ad escludere il coinvolgimento del cliente, dato che richiede l’uso di linguaggio estremamente specifico, e ciò potrebbe portare a un rischio di scarsa comunicazione tra il cliente e lo sviluppatore.

Approccio utilizzato da noi in relazione con quello esposto nel seminario:

All’interno del nostro team, è stato privilegiato un approccio BDD (Behavior-Driven Development), poiché questo metodo favorisce una maggiore collaborazione con un ipotetico cliente. Questo approccio ci consente di adattare lo sviluppo delle diverse componenti in base ai requisiti richiesti e al feedback ricevuto, promuovendo una maggiore interazione e comprensione reciproca.

h. Molinari

Nel seminario tenuto da Andrea Molinari, è stato affrontato il problema di come contrastare i sistemi legacy e, più in generale, individuare la migliore soluzione per mantenere un sistema aggiornato con tecnologie e linguaggi in costante evoluzione.

Gli approcci principali sono due: il primo, molto radicale, consiste nella sostituzione totale del sistema, comportando la sostituzione di software e hardware e la migrazione dei dati. Il secondo, invece, richiede l'interazione o addirittura l’integrazione del vecchio software con quello nuovo, consentendo al sistema precedente di comunicare o coesistere esternamente o internamente al nuovo sistema.

La prima opzione risulta relativamente semplice, ma molto costosa in termini economici, Al contrario, la seconda opzione risulta più economica, ma richiede un periodo di tempo più prolungato e complesso in termini di tempo.

Egli, inoltre, ha sottolineato che una metodologia Agile non è efficace per gestire i sistemi legacy, in quanto la transizione da tali sistemi si concentra maggiormente su processi e tools, trascurando l'interazione con il cliente e con l'utente.

Al contrario, un approccio predittivo, in cui i requirements sono pianificati con estrema attenzione fin dall'inizio e raramente subiscono modifiche, risulta essere più solido e funzionale per il raggiungimento dell’obiettivo prefissato.

Approccio utilizzato da noi in relazione con quello esposto nel seminario:

Nella fase di pianificazione del nostro progetto, abbiamo preferito utilizzare una metodologia Agile. Tale scelta ci ha permesso di aggiornare i requisiti dopo la conclusione della pianificazione iniziale. Questa decisione è stata motivata anche dalla consapevolezza della possibilità di modifiche ai requisiti da parte di un ipotetico cliente, opzione non praticabile con un approccio predittivo.

i. Seminario: Marsiglia

Nel seminario tenuto da Gerardo Marsiglia, è stato introdotto un approccio leggermente diverso rispetto alla metodologia Agile, denominato DevOps.

Attraverso questo metodo, un team riesce a mantenere un approccio basato sullo sviluppo iterativo e sul feedback degli utenti, permettendo contemporaneamente di sostenere un ritmo di delivery molto elevato.

Il concetto chiave del DevOps è la stretta interconnessione tra la parte Dev (sviluppo, che comprende le fasi di pianificazione, di scrittura del codice, di build e di test) e la parte Ops (operations, che coinvolge deployment e monitoraggio).

Un altro concetto strategicamente rilevante è l'automazione dei processi, specialmente nelle fasi di build, test e deployment, che rende possibile sostenere la frequenza di delivery prefissata.

Durante il seminario, sono state presentate diverse alternative per la pianificazione dell'architettura del software, iniziando dal cosiddetto "monolito", composto da una gerarchia di layers (GUI, Business e Data Layer in ordine), fino all'architettura a microservizi nella quale il funzionamento del software si basa su servizi piccoli e indipendenti tra di loro.

Approccio utilizzato da noi in relazione con quello esposto nel seminario:

All’interno del nostro gruppo, abbiamo optato per una frequenza di delivery più bassa. Tale scelta, tuttavia, è stata finalizzata a garantire un controllo più efficace sul corretto funzionamento delle singole componenti e del sistema nel suo complesso.

j. Seminario: APSS & Trentino.ai

Durante il seminario tenuto da Alessandro Bazziga, direttore del Dipartimento Tecnologie dell'Azienda Provinciale per i Servizi Sanitari (APSS), e Alessandro Zorer, cofondatore di Trentino.ai, è stata presentata l'architettura del sistema impiegato dall'APSS nel settore pubblico.

I relatori hanno sottolineato l'importanza di un approccio maggiormente prevedibile, specialmente dal punto di vista finanziario, in contesti simili. Inoltre, hanno illustrato il workflow organizzativo all'interno di un'azienda di grandi dimensioni, suddividendolo nei seguenti processi:

- processo di demand;
- assessment legale-etico;
- processo tecnologico (integration, transformation, creation…).

Il gateway è emerso come elemento centrale del workflow, con la funzione di unico punto di accesso e di restituzione dei risultati ottenuti.

Nella parte finale del seminario, Zorer ha illustrato le potenzialità di integrare sistemi di intelligenza artificiale nel sistema esistente dell’APSS. Tali integrazioni promettono un miglioramento significativo nella qualità delle cure e nell’efficienza operativa. Le soluzioni basate su intelligenza artificiale possono affrontare la carenza di personale medico specializzato, consentendo un monitoraggio più tempestivo dei pazienti e persino consulenze mediche a distanza.

Approccio utilizzato da noi in relazione con quello esposto nel seminario:

Il workflow organizzativo presentato in questo seminario si è rivelato utile durante la revisione dei vari deliverable. In particolare, nell’analisi di alcune funzionalità all’interno di tali documenti, come il D1 e il D2, è emersa la necessità di apportare alcune modifiche al fine di garantire la conformità agli standard legali, soprattutto dal punto di vista della sicurezza.

2. Organizzazione del lavoro

Il processo lavorativo è stato principalmente strutturato in base alle competenze specifiche di ciascun membro. Alcuni hanno dimostrato talento naturale nello sviluppo del codice, mentre altri si sono distinti nella parte logica o concettuale del progetto.

I mezzi impiegati sono stati diversi:

- **Google Docs** - per consentire la collaborazione simultanea su testi anche quando la presenza fisica non era possibile, agevolando così il lavoro di gruppo;
- **Microsoft Word** è stato impiegato per la stesura definitiva dei documenti;
- **Lucidchart** - è stato sfruttato per creare diagrammi;
- **FlowMapp** - è stato utilizzato per la realizzazione degli User Flow Diagram;
- **Canva** - è stato impiegato per la creazione dei mockup;
- **Discord** - ha favorito le interazioni tra i membri, soprattutto quando non era possibile riunirsi fisicamente nello stesso luogo;
- **VSCode** - è stato il nostro strumento principale per la scrittura del codice effettivo del sito web;
- **GitHub** - è stato adoperato per archiviare le diverse versioni del codice e dei documenti, facilitando la collaborazione tra i membri del gruppo.

In alcune parti del progetto, la collaborazione e la sinergia tra i membri sono state più fluide, consentendo una maggiore coesione nel lavoro su documenti e nello sviluppo della Web App.
In periodi più complessi, abbiamo affrontato le sfide attraverso l'implementazione di linee guida rigide, le quali sono state seguite dai membri nel corso della realizzazione delle rispettive attività. Questo approccio ci ha permesso di mantenere l'allineamento sulle scadenze anche quando il lavoro individuale risultava essere il metodo più efficace per la creazione dei documenti.

3. Ruoli e attività

Componente del team	Ruolo	Principali attività
Emma Leonarduzzi	Project leader	Il ruolo principale è stato quello di gestione del progetto. Questo implica la redazione sia delle bozze che delle versioni definitive dei documenti, gestendo la consegna e verificando la correttezza e coerenza degli stessi. Ha contribuito a tutti i deliverable (D1, D2, D3, D4 e D5).
Alessandro Busola	Developer	Il ruolo principale è stato quello dello sviluppo della WebApp e delle corrispondenti API. Inoltre, ha dimostrato la sua utilità nella parte logica del lavoro. Ha contribuito a tutti i deliverable, in particolare D1, D2, D4.
Lorenzo Zarantonello	Co-redattore e developer	Il ruolo principale è stato quello di stesura delle bozze e di sviluppo e codifica di alcune sezioni del progetto. Ha contribuito al D1, parte del D2 e al D5.

4. Carico e distribuzione del lavoro

Nel corso dello sviluppo del progetto, il team ha cercato di suddividere il lavoro in modo collaborativo e mirato, assegnando le responsabilità di sviluppo alle persone più competenti in quel settore e affidando il resto della creazione agli altri membri.

Nel primo stadio del processo, i membri del gruppo hanno partecipato a una sessione di brainstorming per capire l’idea di base, delineare in modo generale le caratteristiche e definire le dinamiche principali della WebApp e del progetto stesso.

Nella fase D1, Emma Leonarduzzi, Lorenzo Zarantonello e Alessandro Busola hanno collaborato nella stesura della bozza del documento D1. Successivamente, Emma ha lavorato autonomamente per riscrivere la bozza, producendo la versione finale del D1 e realizzando tutte le schermate di design del front-end. Nel frattempo, Alessandro ha sviluppato lo schema del design del back-end.

Nella fase D2, Emma, Lorenzo e Alessandro hanno collaborato alla stesura della bozza dei capitoli “Requisiti funzionali” e “Requisiti non funzionali”. In questa fase, Alessandro si è focalizzato sulla creazione degli schemi degli Use Case, mentre gli altri membri si sono concentrati maggiormente sulla componente testuale. Successivamente, Emma ha completato la stesura degli altri capitoli del D2, realizzando anche il diagramma di contesto e il diagramma dei componenti.

Nella fase D3, Emma ha lavorato in autonomia, scrivendo sia la bozza che la versione finale del documento e realizzando il diagramma delle classi e il codice in OCL.

Nella fase D4, Alessandro ha sviluppato quasi interamente da solo la WebApp, le relative API e alla stesura di un’estensiva bozza del documento, mentre Emma ha redatto il documento finale riguardante la documentazione.

Nella fase D5, Lorenzo si è concentrato sulla parte relativa ai seminari, mentre Emma ha scritto la bozza sui punti rimanenti, che è stata successivamente concordata da tutti i membri prima della consegna.

	D1	D2	D3	D4	D5	TOT
Emma Leonarduzzi	15	36	18	26	13	108
Alessandro Busola	9	24	7	71	1	112
Lorenzo Zarantonello	5	6	2	0	18	31
TOT	29	66	27	96	32	251

5. Criticità

Le criticità riscontrate durante il corso del progetto si concentrano prevalentemente sulla limitata partecipazione di un membro del team.

In particolare, Lorenzo Zarantonello ha affrontato delle difficoltà personali nel periodo relativo alla fase D2, compromettendo la sua capacità di contribuire al progetto. Tuttavia, non c’è stata alcuna iniziativa da parte sua nel riprendere il lavoro.

Per questo motivo, Emma Leonarduzzi ha cercato di capire se le circostanze personali di Lorenzo giustificassero la sua limitata partecipazione, sollecitandolo poi a riprendere le attività.

A causa di questa situazione, caratterizzata da una collaborazione limitata, è emersa l'urgenza di riformulare l'approccio di lavoro per adattarsi alla presenza di soli due membri attivi.

In prossimità della data di consegna, Lorenzo ha manifestato interesse a contribuire e, di conseguenza, si è trovato un ruolo nella fase D5.

Un ulteriore elemento critico riguarda lo sviluppo effettivo della WebApp. Infatti, ad Alessandro Busola è stato assegnato il ruolo di sviluppatore, in quanto è il membro più competente, avendo una conoscenza pregressa, seppur minima, in HTML, CSS e JavaScript.

Tuttavia, tale competenza non si è rivelata sufficiente, causando problematiche e ritardi rispetto alla tabella di marcia iniziale.

Inizialmente, nella suddivisione dei compiti, anche Lorenzo era coinvolto nella realizzazione della WebApp. Tuttavia, non essendo stato disponibile per molto tempo, Alessandro si è ritrovato a gestire un carico di lavoro molto più elevato.

6. Autovalutazione

Nel complesso, il nostro gruppo è riuscito a portare a termine l’intero lavoro nonostante le criticità incontrate lungo il percorso, come evidenziato precedentemente.

L’approccio di mantenere ruoli fissi e specifici, nonostante le competenze versatili di ciascun membro, è stato cruciale nel gestire la prolungata assenza di un componente del team. Questa struttura ci ha permesso di adattarci alle circostanze e mantenere un progresso costante nel progetto.

Alessandro Busola si è dedicato con determinazione allo sviluppo della WebApp, partecipando comunque alla redazione dei documenti come impiego secondario. La sua dedizione e la sua visuale improntata alla programmazione sono state una risorsa preziosa per il progresso complessivo del progetto.

Emma Leonarduzzi ha dimostrato una versatilità notevole, concentrandosi principalmente sullo sviluppo concettuale e sulla stesura dei testi nei vari documenti, rendendosi anche disponibile per eventuali chiarimenti riguardanti il design e i vari requisiti da implementare nello sviluppo della WebApp. Il suo coinvolgimento in ogni aspetto del progetto non solo ha contribuito a garantire una coesione e completezza globali, ma ha anche fornito supporto in situazioni in cui si sono verificati problemi. Emma ha rispettato le sue responsabilità acquisite tramite il ruolo di Team Leader cercando di gestire al meglio la situazione riguardante le criticità di cui sopra.

Nonostante le difficoltà legate alla sua assenza e al mancato rispetto delle scadenze, Lorenzo Zarantonello ha dimostrato precisione e competenza nelle parti del progetto a cui si è dedicato. La sua presenza, sebbene limitata, ha contribuito in modo positivo a determinati aspetti.

	VOTO
Emma Leonarduzzi	30
Alessandro Busola	
Lorenzo Zarantonello	