

Ingegneria del Software

Dipartimento di Ingegneria e Scienza dell'Informazione

Gruppo G-51: Giuliano Campagnolo, Elisa Beltrame, Mattia Marini



Mergify - merge your events

Deliverable 5

Report finale



Indice

1	Obiettivi del documento	3
2	Approcci all'ingegneria del Software	3
2.1	Seminario Blue Tensor	3
2.2	Simulazione Metodo Kanban	4
2.3	Seminario IBM	4
2.4	Seminario META	5
2.5	Seminario U-Hopper	5
3	Organizzazione del Lavoro	6
3.1	Organizzazione dei Gruppi di Lavoro	6
3.2	Organizzazione delle Attività	6
4	Ruoli e Attività	6
5	Carico e Distribuzione del Lavoro	7
6	Criticità	7
7	Autovalutazione	7

1 Obiettivi del documento

Questo documento presenta un'analisi approfondita dell'organizzazione del lavoro, compresi i ruoli assegnati, il tempo dedicato da ciascun membro, le criticità riscontrate e un'autovalutazione complessiva.

2 Approcci all'ingegneria del Software

2.1 Seminario Blue Tensor

Jonni Malacarne, CEO di Blue Tensor, ha illustrato i metodi di ingegneria del software applicati allo sviluppo di progetti di intelligenza artificiale (AI). Il team di sviluppo, composto da diverse figure tra cui Project Manager, Senior Software Architect, Ai Engineers e altri, utilizza un approccio basato sui Work Package (WP). Questi WP sono suddivisi in cinque operativi e uno trasversale:

- **Analisi e Fattibilità (WP1):** Il team collabora con il cliente per definire le funzionalità richieste e identificare le criticità del progetto, stabilendo requisiti funzionali e non funzionali.
- **Setup e Design (WP2):** Si progetta l'architettura del sistema, includendo l'infrastruttura hardware e software. Si utilizzano tecniche come lo Story Mapping per definire le funzionalità desiderate.
- **Data Preparation (WP3):** Si selezionano e preparano i dati necessari per l'addestramento dei modelli di intelligenza artificiale, creando dataset di training e di test.
- **Sviluppo - Training - Test - Deploy (WP4):** Il team sviluppa il software in modalità Agile, rilasciando piccole porzioni di software testabili. Si garantisce un feedback tempestivo per soddisfare le esigenze del cliente.
- **Final Release (WP5):** Si procede al rilascio finale del progetto, includendo la raccolta dei feedback e l'implementazione di eventuali miglioramenti identificati durante il processo.

Questo approccio strutturato e metodico consente al team di gestire il progetto in modo efficiente, garantendo il raggiungimento degli obiettivi e la soddisfazione del cliente.

2.2 Simulazione Metodo Kanban

Nel seminario tenuto dal Dr. York Rössel, si è andato a simulare un metodo di ingegneria del Software ideato da David J. Anderson: il metodo Kanban, che consiste in un approccio di gestione del lavoro basato sulla visualizzazione visiva dei processi per migliorare trasparenza, comunicazione ed efficacia. Le sue caratteristiche principali includono la creazione di una board divisa in colonne rappresentanti le varie fasi del progetto, come Design, Implementation e Done. Ogni parte del lavoro è simboleggiata da un post-it che, una volta completato, viene spostato alla colonna corrispondente, fino al completamento. Inoltre, per ogni colonna di lavoro, si stabilisce un limite massimo di post-it consentiti per mantenere un flusso costante e identificare inefficienze nel processo. Il metodo Kanban facilita il rilevamento tempestivo di ritardi e blocchi nel progetto, consentendo una visione complessiva del lavoro e un adattamento continuo alle esigenze e ai cambiamenti nel tempo. Questa flessibilità favorisce ulteriori miglioramenti e adattamenti, rendendo il processo più efficiente e adattabile.

2.3 Seminario IBM

Alessandro Gorga, rappresentante tecnico aziendale di IBM, illustra il potenziale del cloud come un vasto negozio di tecnologie software distribuite, fornendo potenza computazionale e servizi software accessibili a tutti. Il Cloud IBM affronta le sfide attuali fornendo un contesto sicuro per eseguire sistemi software, garantendo segretezza attraverso la crittografia avanzata e offrendo tecnologie open source all'avanguardia. Inoltre, favorisce l'utilizzo sostenibile dell'informatica industriale ottimizzando i consumi e riciclando il calore. Il Cloud IBM offre un ambiente ideale per lo sviluppo collaborativo del software, consentendo ai team di lavorare insieme in modo efficiente e flessibile. Attraverso una toolchain integrata nel catalogo del Cloud IBM, è possibile creare un ambiente di sviluppo completo che comprende strumenti per la gestione del codice, la compilazione, i test e il rilascio del software. La toolchain supporta l'integrazione continua (CI) e la distribuzione continua (CD), permettendo al team di eseguire automaticamente test e rilasci ogni volta che viene apportata una modifica al codice sorgente. Questo processo automatizzato garantisce la rapidità e l'affidabilità delle modifiche apportate al software, riducendo al minimo il rischio di errori e semplificando il processo di sviluppo. Inoltre, il Cloud IBM offre una vasta gamma di servizi e risorse che possono essere integrati nella toolchain per migliorare le capacità di sviluppo del software. Ad esempio, è possibile utilizzare servizi di analisi dei dati per ottenere insight utili dal proprio software, o servizi di intelligenza artificiale per aggiungere funzionalità avanzate.

2.4 Seminario META

Heorghi Raik condivide le metodologie di Software Engineering all'interno di Meta, un'azienda multinazionale. I Software Engineering in Meta sono responsabili della realizzazione dei progetti e hanno diverse responsabilità. All'inizio della loro carriera in Meta, si concentrano principalmente sulla programmazione, ma con l'esperienza acquisita, possono avanzare verso ruoli di Project Manager, una pratica non comune in tutte le aziende ma adottata in Meta. Le loro responsabilità tecniche includono la programmazione, la progettazione architettonica, il project management e altro ancora. Oltre alle responsabilità tecniche, i Software Engineering hanno anche compiti non tecnici, come la gestione delle risorse umane per il progetto, la promozione del team building e l'assunzione del ruolo di mentore o coach per altri membri del team. Rispetto alla metodologia di lavoro, Meta non impone un metodo specifico, lasciando a ciascun gruppo di lavoro la libertà di decidere come organizzare il proprio lavoro. Inoltre, non esiste un linguaggio di programmazione ufficiale all'interno dell'azienda; i dipendenti possono utilizzare il linguaggio che preferiscono. Tuttavia, è consigliabile utilizzare linguaggi noti per garantire la comprensibilità del codice. Tra i linguaggi più utilizzati vi sono Hack, JavaScript, Python e SQL. Questa flessibilità consente ai dipendenti di adattare il proprio lavoro alle esigenze specifiche del progetto e alle proprie competenze linguistiche.

2.5 Seminario U-Hopper

Daniele Miorandi, CEO di U-Hopper, guida un'azienda specializzata nell'analisi di big data e Intelligenza Artificiale. Utilizzano tecnologie open source e infrastrutture cloud per gestire grandi quantità di dati. I linguaggi principali sono Python, Scala e Go, con focus sui microservizi e sistemi asincroni per l'efficienza e la scalabilità. Il processo di sviluppo di U-Hopper prevede la suddivisione del codice in sprint iterativi, definendo milestone, issue e scadenze. Il processo di sviluppo inizia definendo una milestone e suddividendo il lavoro in una serie di issue, stabilendo scadenze per ogni fase. Questo framework guida lo sviluppo degli sprint, partendo dalla fase di codifica, seguita dalla definizione dei test (con una copertura minima del 70%) e la revisione del codice tramite Merge Requests per garantirne la solidità. I deployment avvengono attraverso diversi ambienti di test, con un Deployment staging che simula l'ambiente di produzione. Una volta superati i test, il codice viene distribuito in produzione, con una particolare attenzione al monitoraggio continuo e alla gestione delle anomalie attraverso il rollback automatico. Il monitoraggio e il controllo dell'operato sono fondamentali in U-Hopper, in quanto garantiscono il funzionamento ottimale del software e la rilevazione tempestiva di eventuali problemi. La metodologia adottata consente una gestione efficace e flessibile dei progetti, assicurando un alto standard di qualità e performance.

3 Organizzazione del Lavoro

3.1 Organizzazione dei Gruppi di Lavoro

Il lavoro è stato organizzato in gruppi, ciascuno con compiti specifici e obiettivi definiti. Ogni membro del gruppo è stato assegnato a compiti specifici, con obiettivi chiari e scadenze precise. La suddivisione dei compiti è stata effettuata tenendo conto delle competenze e delle preferenze di ciascun membro del gruppo.

3.2 Organizzazione delle Attività

L'organizzazione delle attività è avvenuta attraverso una pianificazione dettagliata delle varie fasi del progetto. Sono stati utilizzati strumenti di gestione del progetto per monitorare lo stato di avanzamento e per garantire che tutte le scadenze fossero rispettate. Ogni attività è stata assegnata a un membro del gruppo, con obiettivi chiari e scadenze precise.

4 Ruoli e Attività

Componente del team	Ruolo	Principali attività
Giuliano Campagnolo	Progettista tecnico, relatore e tester	Si è occupato principalmente alla stesura dei diagrammi e al testing. Ha contribuito alla stesura e alla revisione dei Deliverables, in particolare D3 e D4.
Mattia Marini	Designer, sviluppatore front-end, back-end e documentazione	Designer del gruppo, principale progettista del prototipo e successivamente programmatore di front, back-end e strumenti di utility, documentazione e testing. Ha contribuito nella stesura dei deliverables e nella loro impaginazione.
Elisa Beltrame	Product Manager	Si è occupata della scrittura dei documenti e della loro revisione, in particolare del D1, D2 e D5. Inoltre ha collaborato nella progettazione del prototipo.

Tabella 1: Ruoli e principali attività del team

5 Carico e Distribuzione del Lavoro

L'analisi dei file di log ha evidenziato la distribuzione del carico di lavoro, espresso in ore/persona, per ciascun membro del gruppo. Tale distribuzione è stata progettata con l'obiettivo di assicurare una suddivisione equa del lavoro, basata principalmente sulle competenze individuali di ciascun membro. Un riepilogo delle ore è riportato nella Tabella 2.

Componente del gruppo	D1	D2	D3	D4	D5	TOT
Giuliano Campagnolo	15h	10h	6h	84h	6h	121h
Mattia Marini	24h	15h	4h	131h	3h	177h
Elisa Beltrame	23h	18h	6h	56h	20h	123h
TOT	52h	37h	16h	234h	32h	371h

Tabella 2: Ore di lavoro per componente del gruppo

6 Criticità

All'inizio del progetto, abbiamo subito affrontato una sfida: l'organizzazione degli incontri per lavorare collettivamente. Ciò ha causato ritardi nel progresso complessivo, in particolare nella stesura del D1 come riportato nella Tabella 2. In seguito, abbiamo adottato un approccio organizzativo diverso, preferendo suddividere il lavoro in piccole parti e lavorare autonomamente per il raggiungimento degli obiettivi del progetto. Questo cambiamento ha portato a una significativa riduzione delle ore impiegate per il D2 e il D3, accelerando così il ritmo di lavoro e migliorando la qualità del prodotto finale.

Per quanto riguarda il D4, ci sono state difficoltà nella scrittura del codice, vista l'assenza di conoscenze pregresse da parte dello sviluppatore, le quali hanno portato ad un ritardo secondo quanto previsto, complice anche l'inizio della sessione invernale. Per evitare ulteriori ritardi, abbiamo quindi deciso di riprendere prima i seminari del D5.

7 Autovalutazione

Durante l'intero progetto, abbiamo mantenuto un costante impegno lavorativo. Nonostante i problemi riscontrati e il differente numero di ore impiegato, ogni membro del gruppo ha contribuito a sostenere gli altri. L'organizzazione, la suddivisione e il metodo di lavoro scelto hanno permesso al nostro gruppo di superare queste difficoltà con successo, provando l'indiscutibile valore aggiunto di un lavoro in team in una simulazione di contesto lavorativo. Sulla base delle considerazioni espresse, la valutazione finale che noi riteniamo opportuna è riportata nella Tabella 3.

Componente del gruppo	Voto
Giuliano Campagnolo	30
Mattia Marini	30
Elisa Beltrame	30

Tabella 3: Valutazione finale del gruppo