

Progetto Ingegneria del Software: Gestionale “Sweet & Sour”

Matteo Casarotto Sant’Ana, Alvin Preidt, Enrico Zasso

Progetto Ingegneria del Software: Gestionale “Sweet & Sour”	1
1. APPROCCI ALL’INGEGNERIA DEL SOFTWARE	2
1.1. Blue Tensor	2
1.2. Il metodo Kanban	2
1.3. IBM	3
1.4. META	3
1.5. U-Hopper	4
1.6. RedHat	4
1.7. Microsoft	4
1.8. Molinari	5
1.9. Marsiglia	5
1.10. APSS	6
2. ORGANIZZAZIONE DEL LAVORO	6
3. RUOLI E ATTIVITA’	7
4. CARICO E DISTRIBUZIONE DEL LAVORO	8
5. CRITICITA’	8
6. AUTOVALUTAZIONE	9

DELIVERABLE 5

1. APPROCCI ALL'INGEGNERIA DEL SOFTWARE

Nella seguente sezione forniamo una breve descrizione dei seminari tenuti in aula durante l'anno, talvolta esprimendo anche considerazioni personali.

1.1. Blue Tensor

BlueTensor Srl è una società specializzata nello sviluppo di progetti basati sull'intelligenza artificiale (AI) che offre soluzioni su misura per migliorare il processo decisionale, la produzione, l'esperienza utente e la gestione delle conoscenze. L'azienda è nota per la partecipazione a progetti multidisciplinari, infatti alcuni settori in cui hanno collaborato sono: controllo qualità, videosorveglianza, agricoltura e soccorso. Questo aspetto ci ha particolarmente colpito poiché ha evidenziato come l'ingegneria del software abbia applicazioni illimitate. Il CEO Jonni Malacarne ha però sollevato una criticità, cioè la specializzazione necessaria per portare a termine ogni progetto. In sintesi, il seminario rappresenta un'occasione per esplorare come l'ingegneria del software e l'intelligenza artificiale si intersecano con il mondo delle industrie moderne.

1.2. Il metodo Kanban

Creato negli anni 2000 da David J. Anderson, Kanban è un metodo incentrato sulla visualizzazione del lavoro, mantenendo costante il suo flusso. Questo consente ai team di gestire il lavoro in modo più efficiente. Le prime implementazioni sono state fatte in Corbis, Microsoft e Yahoo.

Kanban utilizza il Kanban Board e il Kanban System per rappresentare visivamente il flusso di lavoro e ciò aiuta i team a comprendere lo stato dei compiti in modo intuitivo. Kanban impone un limite di Work in progress (WIP) , cioè un limite al numero di attività svolgibili contemporaneamente, per evitare sovraccarichi; i compiti vengono presi nel flusso di lavoro solo quando c'è la capacità di gestirli. Abbiamo riscontrato però una criticità, cioè che se il metodo kanban non è applicato nel modo corretto a causa, per esempio, della mancanza di collaborazione tra i membri del team, il lavoro potrebbe essere inefficiente e si corre il rischio di sovraccaricare il team comportando un rallentamento generale.

1.3. IBM

Nel seminario IBM ci viene presentata la sua piattaforma IBM Cloud, la quale offre potenza computazionale e tecnologie software come servizi, ideali per modernizzare i sistemi software, facilitare la trasformazione digitale (Digital Twin) e creare nuove opportunità commerciali. Permette agli utenti di concentrarsi sullo sviluppo del codice, riducendo le preoccupazioni legate all'infrastruttura, e di fruire a tecnologie come AI, Quantum Computing, Blockchain, Serverless e Machine Learning che facilitano la modernizzazione dei sistemi software e la trasformazione digitale, aprendo nuove opportunità di business. Tra i principali vantaggi, IBM Cloud garantisce inoltre un ambiente sicuro per i sistemi software e promuove un uso ecologico e sostenibile dell'informatica ottimizzando i consumi energetici favorendo il riciclo del calore. Il seminario mira, inoltre, a riflettere su tematiche come la sicurezza informatica e all'ecosostenibilità nell'ambito dell'informatica industriale.

1.4. META

Nel seminario di Meta, condotto da un ex studente dell'università di Trento, non ci si è concentrati su un metodo di sviluppo software specifico, ma ci si è focalizzati sulle responsabilità del software engineering, affrontando ambiti sia tecnici che non. Gli aspetti tecnici comprendono codifica, progettazione architeturale, gestione di progetti e mansioni correlate, mentre le responsabilità non tecniche includono il reclutamento (tramite interviste), la formazione del team e il mentoring. Per quanto riguarda le metodologie di sviluppo, non è preso in considerazione un approccio specifico, ma ciascun team adotta le pratiche ritenute più efficaci. I team, quindi, si adattano in base alle esigenze specifiche adottando un approccio di tipo AGILE. Ciò che ci ha particolarmente colpiti è il fatto che si dia molta più importanza alle competenze generali e alle capacità logiche piuttosto che alla conoscenza specifica di molti linguaggi di programmazione. Infatti l'ingegnere del software non deve possedere competenze unicamente nella programmazione ma deve considerare anche il project management e deve sapersi adattare alle esigenze dei singoli progetti.

1.5. U-Hopper

U-Hopper è un'azienda deep-tech che aiuta le imprese a creare valore dei dati attraverso soluzioni personalizzate di data analysis e intelligenza artificiale, evidenziando gli strumenti tecnologici e i framework utilizzati, sia proprietari che open source come Spark, MongoDB, PostgreSQL per l'analisi e archiviazione di dati e MLFlow, Tensor Flow e PyTorch nel campo dell'IA. Per quanto riguarda lo sviluppo software, U-Hopper adotta un approccio metodologico che include la definizione di best practices da loro chiamate 'sprint definition' e 'sprint development' che portano ad un approccio agile e dinamico. Vengono effettuate spesso revisioni tramite merge requests per mantenere il codice ordinato e pronto per il rilascio. Utilizzano notifiche su Slack per gli aggiornamenti sullo stato dei progetti e considerano la documentazione essenziale.

1.6. RedHat

Questo seminario è incentrato sul mondo dell'open source. Quest'ultimo offre numerosi vantaggi: condivisione della conoscenza, peer review, meritocrazia, visibilità e una vasta comunità di supporto. Le licenze open source si dividono in due categorie principali: copyleft (come GPL) che richiedono la redistribuzione con la stessa licenza, e non-copyleft (come Apache e MIT) che non impongono tale obbligo. L'open source offre numerosi vantaggi alle aziende; i clienti scelgono questo tipo di software per i costi inferiori e la migliore qualità generale. Partecipare a un progetto open source può avvenire a vari livelli, dal segnalare bug (principianti) fino a diventare membri chiave del team. Questo seminario ci ha fatto inoltre capire l'importanza delle relazioni pubbliche e dei contributi esterni e al saperli includere nel proprio progetto per mantenerlo sano nel tempo.

1.7. Microsoft

Il seminario Microsoft ci ha fatto porre maggiormente l'attenzione su una singola parte dell'ingegneria del software, cioè il testing. Esso è una verifica, automatizzata o manuale, dei requisiti funzionali e non funzionali e può essere fatta su più livelli e ciò porta ad avere diverse tipologie di testing (es. acceptance testing). Vengono esplorate le diverse motivazioni che portano all'esecuzione dei test, come la scoperta di bug, la prevenzione della regressione comportamentale e prestazionale e la verifica delle specifiche contrattuali. Il seminario, inoltre, ci presenta alcuni framework e strumenti di testing.

1.8. Molinari

Il seminario del professore Andrea Molinari si è focalizzato sui sistemi legacy, cioè i sistemi informativi formati interamente o parzialmente da componenti obsolete. Questi sistemi persistono ancora poiché continuano a rispondere alle esigenze originali, la loro conoscenza è approfondita e la loro affidabilità riduce i rischi operativi, garantendo, inoltre, continuità aziendale. Un ingegnere del software deve tenere conto dell'esistenza di questi sistemi poiché quando va a sviluppare un sistema, questo deve essere in grado di interagire con un sistema legacy. Ovviamente, l'utilizzo di questi sistemi ha delle criticità, infatti un hardware o un software obsoleto possono causare problemi di incompatibilità, la gestione e la manutenzione di tali sistemi richiedono risorse e costi molto alti e il loro utilizzo limitano l'innovazione.

1.9. Marsiglia

La presentazione sull'Application LifeCycle Management (ALM) copre vari aspetti del ciclo di vita del software, inclusi gestione della configurazione, gestione dei progetti e gestione del cambiamento. Viene analizzata l'evoluzione dei processi di sviluppo software, dai metodi tradizionali a cascata (waterfall) alle pratiche moderne come DevOps sottolineando l'importanza dell'automazione e della collaborazione. Inoltre, vengono approfonditi i concetti fondamentali legati all'architettura software, mostrando come le architetture monolitiche si siano evolute verso approcci più moderni come i microservizi e le applicazioni cloud-native. In particolare ci si concentra sulla scalabilità, la resilienza e l'uso dei container. Nella conclusione della presentazione, si enfatizza quanto sia importante modernizzare le applicazioni tradizionali per sfruttare al massimo le opportunità offerte dalle nuove architetture, tecnologie e piattaforme.

1.10. APSS

L'Azienda Provinciale per i Servizi Sanitari (APSS) coordina le attività sanitarie e sociosanitarie in Trentino, garantendo continuità e umanizzazione dei percorsi di cura e un'offerta equa dei servizi sanitari. Il Dipartimento Tecnologie dell'APSS è cruciale in questo processo, sviluppando e mantenendo tecnologie di informazione e comunicazione, collaborando con la Direzione aziendale e altri dipartimenti. Ciò viene realizzato semplificando e decentralizzando i processi decisionali tecnico-amministrativi, implementando tecnologie innovative e digitalizzando i sistemi gestionali. Le aree di interesse del Dipartimento Tecnologie includono l'ingegneria clinica, che gestisce le attrezzature sanitarie, e i sistemi informativi, che coprono infrastrutture, reti, digitalizzazione e analisi dei dati. È centrale l'integrazione dell'IA per migliorare la qualità delle cure, l'efficienza operativa, la diagnosi, la ricerca medica e la telemedicina. Tuttavia, l'introduzione dell'Intelligenza Artificiale richiede una valutazione etica e la gestione dei dati in conformità alle normative sulla privacy. Il Fascicolo Sanitario Elettronico (FSE) assume un ruolo centrale, poiché consente ai cittadini di accedere alla propria storia clinica e di condividerla con i professionisti sanitari, facilitando l'analisi e la gestione dei dati clinici. Il Patient Summary, parte del FSE, supporta i professionisti nella diagnosi e cura personalizzata dei pazienti. L'approccio del Dipartimento enfatizza l'interconnessione dei dati sanitari, l'accessibilità delle informazioni e il rispetto rigoroso della privacy e della sicurezza dei dati.

2. ORGANIZZAZIONE DEL LAVORO

Il lavoro è stato gestito in questo modo:

- La realizzazione di ogni Deliverable è stata preceduta da un incontro iniziale utile a definire gli obiettivi e a suddividere i compiti. Successivamente venivano fissati nuovi incontri per aggiornare tutti i membri del gruppo dei progressi effettuati o di eventuali criticità rilevate. Gli incontri si sono svolti prevalentemente tramite videochiamata.
- I diversi compiti sono stati assegnati in base alle competenze e alle preferenze di ogni componente del gruppo.

- Per la stesura dei documenti è stato utilizzato Google Document, in modo tale da vedere in tempo reale i progressi e rendere più facile la condivisione del materiale e delle risorse tra i membri del gruppo.
- Gli schemi sono stati realizzati tramite LucidChart e Draw.io, i quali offrono gratuitamente una vasta gamma di diagrammi e schemi UML.
- Per quanto riguarda lo sviluppo della app abbiamo utilizzato React per il front-end, Java Spring per il back-end e MYSQL per il database.
- E' stata poi effettuata una revisione totale dei documenti in modo congiunto.

3. RUOLI E ATTIVITA'

COMPONENTE	RUOLO	PRINCIPALI ATTIVITA'
<i>Matteo Sant'Ana Casarotto</i>	<i>Progettista / Capo Sviluppatore</i>	<i>Ha fornito un contributo significativo come progettista, ideando la maggior parte delle funzioni dell'applicazione. Si è occupato, inoltre, della realizzazione della maggior parte sia del front-end che del back-end</i>
<i>Alvin Preidt</i>	<i>Analista / Sviluppatore</i>	<i>Si è occupato della realizzazione di diversi diagrammi e della scrittura dei documenti. Ha fornito anche supporto alla realizzazione del back-end.</i>
<i>Enrico Zasso</i>	<i>Analista / Sviluppatore</i>	<i>Si è occupato della realizzazione di diversi diagrammi, della scrittura dei documenti e della loro revisione generale. Ha avuto inoltre il compito di verificare che i documenti fossero sempre coerenti tra loro. Ha fornito anche supporto alla realizzazione del front-end.</i>

4. CARICO E DISTRIBUZIONE DEL LAVORO

La tabella sottostante mostra come il carico di lavoro sia stato suddiviso in modo equo tra i tre componenti del gruppo e di come tutti i componenti abbiano contribuito allo stesso modo alla realizzazione del progetto. La differenza più sostanziale la si può notare nel D4 poichè Matteo è stato il membro del gruppo che si è occupato maggiormente dello sviluppo dell'applicazione e nel D5 dove Enrico è stato il componente del team che si è occupato, quasi totalmente, della scrittura del documento.

	<i>D1</i>	<i>D2</i>	<i>D3</i>	<i>D4</i>	<i>D5</i>	<i>TOT</i>
Matteo Sant'ana Casarotto	22	16	17	49	2	106
Alvin Preidt	19	24	25	25	3	96
Enrico Zasso	20	26	22	21	10	99
TOT	61	66	64	95	15	301

5. CRITICITA'

Una delle criticità maggiori che abbiamo riscontrato è stata quella di definire le date e gli orari degli incontri. Questo è stato causato dall'elevato numero di ore di lezioni universitarie dei membri del gruppo durante la settimana e dagli impegni lavorativi di Matteo e Alvin. Per risolvere il problema abbiamo cercato di rendere il più utile possibile ogni incontro effettuato, definendo i compiti nel modo più preciso possibile; alcuni incontri sono stati fissati da solo due dei componenti del gruppo, aggiornando successivamente il terzo sui progressi effettuati.

Inoltre, non siamo riusciti a sviluppare il testing del nostro sistema per dei problemi di configurazione di Jest.

6. AUTOVALUTAZIONE

Nonostante le criticità sopra citate, crediamo di aver svolto un ottimo lavoro. Ogni componente del gruppo ha partecipato con lo stesso impegno allo sviluppo del progetto, contribuendo con le proprie capacità e competenze. Ci consideriamo soddisfatti del lavoro svolto e, sulla base delle nostre considerazioni e sulla qualità effettiva del prodotto finale, questa è la nostra autovalutazione:

<i>Membro</i>	<i>Voto</i>
Matteo Sant'ana Casarotto	28
Alvin Preidt	28
Enrico Zasso	28