



UNIVERSITÀ
DI TRENTO

Dipartimento di Ingegneria
e Scienza dell'Informazione

Ingegneria del Software

Grab a Book

Gruppo G23

Report Finale

Anno Accademico 2023/2024

Indice

| | |
|---|-----------|
| Scopo del documento | 3 |
| Approcci all'ingegneria del software | 4 |
| <i>BlueTensor</i> | 4 |
| <i>Simulazione metodo Kanban</i> | 4 |
| <i>IBM</i> | 5 |
| <i>META</i> | 5 |
| <i>U-Hopper</i> | 5 |
| <i>RedHat</i> | 6 |
| <i>Microsoft</i> | 6 |
| <i>Molinari</i> | 6 |
| <i>Marsiglia</i> | 7 |
| <i>APSS & Trentino.ai</i> | 7 |
| Organizzazione del lavoro | 8 |
| Ruoli e attività | 9 |
| Carico e distribuzione del lavoro | 10 |
| Criticità | 10 |
| Autovalutazione | 11 |

Scopo del documento

Il seguente documento presenta un report sull'attività svolta.

Verranno descritti i seminari seguiti con le loro specificità e come gli insegnamenti appresi sono stati applicati nel progetto.

Verrà in seguito descritta l'organizzazione del lavoro, la divisione dei ruoli all'interno del team, il carico di lavoro e le criticità riscontrate.

Infine verrà condivisa l'autovalutazione del progetto.

Approcci all'ingegneria del software

Verranno ora descritti brevemente i vari seminari che sono stati offerti durante il corso, mostrando caratteristiche, specificità, punti di forza e criticità degli approcci all'ingegneria del software presentati.

Discutendo infine l'approccio da noi scelto.

BlueTensor

BlueTensor è un'azienda trentina che si occupa di AI. Durante il seminario ci sono stati presentati i vari prodotti da loro sviluppati e il loro processo di sviluppo.

Uno dei punti cardine dell'azienda è la capacità di innovare il tessuto produttivo trentino e non solo grazie all'applicazione dell'AI.

Infatti ci sono stati mostrati alcuni casi studio di *Controllo qualità* e *Analisi predittiva* tramite uno dei loro prodotti chiamato: **EYERUS**. Questi casi mostrano l'importanza di usare strumenti di questo tipo per migliorare la produzione e la gestione di qualsiasi azienda. Per quanto riguarda nello specifico l'ingegneria del software il loro approccio prevede di utilizzare il processo agile soprattutto nella parte di sviluppo, training e test del codice. Mentre più preparatoria ovvero, l'analisi di fattibilità, il set-up & design e la preparazione dei dati, viene realizzato seguendo un approccio molto più simile al modello waterfall.

Questo approccio ci ha molto colpiti ed è quello che abbiamo cercato in buona parte di seguire anche noi, infatti i documenti D1, D2, D3 sono stati sviluppati in team mentre nella parte di sviluppo abbiamo utilizzato un processo agile.

Simulazione metodo Kanban

Durante questo seminario ci è stato presentato il metodo Kanban. La parte più interessante del seminario è stata la simulazione che prevedeva la simulazione del carico e dell'efficienza dell'azienda nel completare dei task, usando un metodo standard senza collaborazione tra i vari componenti del team, poi la possibilità di collaborare oltre a eseguire le proprie task e infine l'applicazione vera e propria del metodo Kanban.

La cosa sorprendente è stata l'efficacia di applicare un WIP (Work In Progress) limit per processo (design, implementation, done). Infatti applicando questo limite che è parte fondamentale del metodo, è stato possibile velocizzare il processo di sviluppo portando a termine task nel giro di 3 massimo 4 giorni di simulazione quando precedentemente le task venivano terminate anche fino a 10 giorni.

Scoprire questo approccio che prevede che il team possa non essere sempre al 100% del carico di lavoro in modo che si possa collaborare o lavorare su task più generali relativi all'azienda è stato veramente incredibile.

Ciò nonostante, dobbiamo ammettere che è stato praticamente impossibile applicare questo metodo nel nostro progetto, probabilmente a causa del tipo di lavoro e della dimensione molto ristretta del team.

IBM

In questo seminario ci è stato presentato il Cloud di IBM e tutte le sue pressoché infinite potenzialità. Parte del seminario è stata occupata dalla presentazione del cloud IBM, da cosa è costituito e a quali sfide tenta di rispondere.

La parte più interessante è stata la seconda in cui ci è stato mostrato seppur in maniera semplificata come si potrebbero utilizzare i servizi offerti per creare le nostre idee di sviluppo. In particolare la facilità nel deployment e nella scrittura di piccoli programmi senza la necessità di scrivere righe di codice ma direttamente tramite blocchi dimostra la potenzialità di raggiungere un gran numero di clienti.

Il fatto che la garanzia di protezione da attacchi esterni e la capacità di distribuire il contenuto rendendolo velocemente accessibile in qualunque parte del mondo, rende il cloud IBM una soluzione molto interessante.

Per quanto riguarda il nostro caso però, crediamo che quello che ci è stato mostrato, seppur interessante, fosse poco applicabile nel nostro progetto. Ciò nonostante è stato interessante conoscere questa possibilità da poter applicare in progetti futuri.

META

In questo seminario abbiamo avuto l'opportunità di conoscere cosa vuol dire lavorare per Meta, in particolare è stato interessante conoscere il loro approccio all'ingegneria del software. Come ci si poteva immaginare, infatti, applicano praticamente un modello agile, ma cosa molto interessante, non applicano nessuna costrizione sugli strumenti e i linguaggi da utilizzare. Addirittura nel colloquio è il recruiter che si adatta al linguaggio che il candidato vuole usare per risolvere i problemi proposti.

D'altra parte alcuni processi come l'utilizzo di fogli di calcolo Google per organizzare il lavoro ci ha abbastanza sorpreso ma è comprensibile visto la libertà che Meta lascia ai suoi dipendenti.

È stato comunque interessante vedere quanto sia considerata importante la libertà di ogni sviluppatore di lavorare come preferisce in modo da essere il più produttivo possibile.

U-Hopper

U-Hopper è un'azienda trentina che si occupa di *"help companies create value out of data"*. Riguardo alla loro storia ci ha molto colpito come una piccola azienda sia stata in grado di lavorare per grandi colossi come Walmart.

Per quanto riguarda invece il loro approccio allo sviluppo e all'ingegneria del software utilizzano metodi agile con sprint per lo sviluppo e per il deployment.

Approccio fondamentale nel loro caso dove è importante mostrare al cliente dei risultati e anche essere in grado di adattarsi all'arrivo di nuove tecnologie.

Come per BlueTensor anche in questo caso ci sono stati mostrati dei progetti sviluppati da loro che ci hanno permesso di comprendere meglio quanto i dati siano una risorsa fondamentale e presente ovunque.

RedHat

Il seminario tenuto da Mario Fusco, che lavora a RedHat, è stato più incentrato sull'open source che su RedHat in sé come azienda.

Una delle cose più interessanti è stata scoprire l'importanza dell'open source non solo come gratificazione personale ma proprio come strumento per migliorare il proprio profilo nei confronti delle aziende.

Infatti Fusco ha raccontato che quando è stato assunto in RedHat non ha dovuto fare alcun colloquio, perché il suo lavoro nello sviluppo di *lambda*j prima e di *Hammurabi* dopo era più che sufficiente per dimostrare le sue qualità come software developer.

Questa nuova consapevolezza dell'importanza del mondo open source ci ha motivati come team a cercare in futuro un progetto in cui poter partecipare anche noi.

Microsoft

Il seminario tenuto da un dipendente di Microsoft era incentrato sul testing, che è spesso una parte molto sottovalutata quando si sviluppa codice.

Il seminario era diviso in due parti. La prima è stata più teorica, in cui Diego Colombo ci ha spiegato perché fare testing, che tipi di test esistono e in che fasi dello sviluppo va testato cosa.

La parte più interessante è stata però la seconda, in cui è stata presentata una piccola demo dei vari software che Diego Colombo utilizza normalmente durante lo sviluppo.

È stato un seminario molto importante per il nostro team perché ci ha permesso di capire l'importanza del fare testing in maniera corretta e durante lo sviluppo, infatti ha poco senso costruire i test per farli funzionare con il nostro codice o modificare il codice per adattarlo al test. Invece è fondamentale che il processo sia fatto insieme e che si utilizzino correttamente i diversi tool disponibili.

Molinari

In questo seminario Andrea Molinari ci ha presentato la sua esperienza con i sistemi legacy, mostrandoci come essi sono ancora molto presenti e in parte fondamentali per molti settori. Durante il seminario abbiamo avuto modo di vedere l'importanza di questi sistemi sia per questioni di sicurezza, per esempio le banche che non possono permettersi failure inaspettate o errori nel codice base che possono portare gravi disservizi ai propri clienti, si affidano ancora molto a sistemi legacy basati su Cobol.

Inoltre i sistemi legacy vengono usati quando grandi aziende fanno investimenti a lungo termine che si aspettano possano durare per molti anni.

Per questo motivo lo sviluppo di questi grandi sistemi non può seguire un metodo agile standard, ma dato i lunghi tempi di sviluppo è necessario usare una metodologia agile creata appositamente per sistemi legacy.

Questo seminario è stato per noi un modo di scoprire un mondo diverso da quello che siamo abituati a vedere e che ci ha dato anche degli spunti per possibili prospettive di lavoro futuro.

Marsiglia

In questo seminario Gerardo Marsiglia ci ha raccontato cosa vuol dire essere un enterprise architect e l'importanza del passare ad un metodo di sviluppo DevOps per stare al passo con i tempi.

Infatti ci ha spiegato come l'evoluzione della tecnologia ma anche il passaggio ad infrastrutture cloud in cui è necessario rispettare degli standard, ha portato moltissime aziende ad adottare il metodo DevOps.

Il metodo prevede di avere una stretta collaborazione tra la parte di development del prodotto alla parte operativa, ovvero tutta la parte di deployment e monitoraggio.

In questo modo si possono adottare soluzioni di Continuous Delivery in cui il codice terminato passa attraverso una serie di test automatizzati per poi essere messo sempre in automatico in produzione.

Tutte queste tecniche possono poi essere utilizzate anche nella necessaria modernizzazione dell'applicazione, parte dei compiti di un application architect, oltre che in qualunque progetto, compresi i nostri, che necessitano di essere scalabili e standardizzati per poter funzionare su piattaforme di cloud diverse.

APSS & Trentino.ai

In quest'ultimo seminario abbiamo avuto modo di conoscere come lavora la APSS e quali tecnologie adopera per gestire l'ingente mole di lavoro.

È stato molto interessante scoprire come lavora un reparto IT in una pubblica amministrazione, soprattutto la grande libertà dal punto di vista di ricerca e sviluppo che viene loro concessa,

Inoltre abbiamo potuto vedere come all'interno di un grande sistema informativo come quello della sanità pubblica trentina si mescolino e vengano utilizzate le diverse tecnologie che abbiamo avuto modo di conoscere attraverso i precedenti seminari.

Infatti stanno migrando la maggior parte dei propri servizi su cloud, si trovano ancora a lavorare con sistemi legacy che vanno mantenuti e aggiornati e utilizzano strumenti di AI per migliorare le prestazioni sanitarie rivolte ai pazienti come per esempio per i soggetti diabetici che rischiano di sviluppare la retinopatia diabetica e che tramite strumenti di analisi delle immagini della retina possono guidare il medico ad una diagnosi rapida e precisa.

Possiamo quindi affermare che questo seminario conclusivo ci ha confermato ancora una volta come tutte le tecnologie e i processi presentati nel corso delle lezioni di ingegneria del software siano parte fondamentale per il sano funzionamento di qualunque azienda: dalla piccola start-up alla grande pubblica amministrazione.

Organizzazione del lavoro

Abbiamo deciso di dividere il progetto principale in due parti: progettazione e sviluppo. Per la prima parte che comprende i Deliverables 1, 2 e 3, abbiamo lavorato per la maggior parte del tempo insieme trovandoci in università, sia a lezione che nei momenti liberi, in modo da poter scambiarci facilmente idee e opinioni sul lavoro che stavamo svolgendo.

In questo modo abbiamo potuto in poco tempo accordarci su un progetto comune da portare a termine e lavorare tutti nella stessa direzione.

Per ognuno dei Deliverable da consegnare creavamo una prima struttura del documento basandoci sulla struttura presentata a lezione e poi lavoravamo ai vari paragrafi dividendoceli tra noi in modo da poter procedere in maniera efficiente.

In questo modo abbiamo cercato di suddividere il lavoro in maniera uniforme sfruttando volta per volta le qualità in ambiti diversi dei componenti del gruppo per ottenere il risultato migliore possibile.

Per la parte di sviluppo e per la stesura dell'ultimo Deliverable abbiamo deciso di suddividere il lavoro in maniera più precisa.

In maniera tale che un componente del gruppo continuasse la stesura dei documenti mentre le altre due potessero portare avanti lo sviluppo del codice.

Gli strumenti utilizzati sono stati un gruppo **Telegram** per la comunicazione asincrona tra i componenti del gruppo, **Google Drive** per la stesura dei Deliverables, **draw.io** per la realizzazione dei vari diagrammi, **GitHub** per l'hosting e l'organizzazione dei Deliverables e del codice, **Vercel** per il deployment.

Ruoli e attività

Riportiamo ora la tabella che riassume i ruoli e le attività svolte da ogni componente del gruppo.

| Componente del team | Ruolo | Principali attività |
|---------------------|---|---|
| Alessandro Cornella | Project leader, Backend, Frontend, Documentazione (Swagger) | Si è occupato di coordinare e organizzare il lavoro del gruppo. Inoltre si è occupato principalmente di scrivere i documenti D4 e D5, della scrittura di parte del backend e della maggior parte del frontend. Si è occupato anche della scrittura della documentazione swagger e del deployment su Vercel. Ha anche partecipato alla realizzazione dei primi 3 deliverables e ai diagrammi del D4. |
| Alessio Paolazzi | Database Manager, Backend, Testing | Il ruolo principale è stata la gestione del database, l'implementazione del backend soprattutto sulla parte di autenticazione. Si è inoltre occupato del testing delle varie api, implementando almeno 2 o 3 test per API sviluppata dal gruppo. Ha partecipato alla realizzazione dei primi tre documenti, e ai vari schemi necessari per il D4. |
| Morgana Pasquini | Backend, Frontend, Testing | Principalmente si è occupata di: Creare delle API riguardanti gli annunci (creazione e visualizzazione), Scrivere i test relativi alle API create, Realizzare una parte del frontend (header, pagina profilo, visualizzazione di un annuncio), Realizzare dei mock up del frontend per il documento D1. Ha partecipato alla realizzazione dei documenti D1, D2, D3 e dei diagrammi del D4. |

Carico e distribuzione del lavoro

Riportiamo ora il carico di lavoro espresso in ore/persona e diviso per i diversi Deliverables.

| Componente del team | D1 | D2 | D3 | D4 | D5 | TOTALE |
|--------------------------------------|------|------|-----|-------|-----|--------|
| Alessandro Cornella | 11,5 | 16,5 | 5 | 45 | 4,5 | 82,5 |
| Alessio Paolazzi | 11 | 18 | 6,5 | 34 | 1 | 70,5 |
| Morgana Pasquini | 14 | 18,5 | 6,5 | 32,5 | 2 | 73,5 |
| di cui ore comuni | 9 | 14 | 5 | 8 | 1 | 37 |
| TOTALE (somma ore totali singole) | 36,5 | 53 | 18 | 111,5 | 7,5 | 226,5 |

Criticità

Durante la prima fase, come descritto precedentemente, abbiamo lavorato molto tempo insieme, questo è stato in parte positivo ma ha allungato di molto i tempi di lavoro portandoci spesso ad essere un po' in ritardo sulle consegne previste.

Inoltre la difficoltà dei Deliverables è aumentata con il passare del tempo, cosa che ha causato un ulteriore rallentamento, insieme alla sovrapposizione delle prove intermedie degli altri corsi da noi seguiti.

Ciò nonostante, dopo l'inizio dello sviluppo del codice siamo riusciti a riorganizzarci, lavorando in base alle nostre disponibilità e dividendo meglio il lavoro in modo tale da poter consegnare nei tempi previsti.

Durante lo sviluppo del codice abbiamo avuto problemi nella rilevazione della copertura dei test: venivano tutti superati, ma la copertura rilevata è sempre stata dello 0%. Anche dopo aver consultato siti come stackoverflow.com e aver configurato correttamente jest, non siamo riusciti a venire a capo di questo problema.

Autovalutazione

Durante questi mesi abbiamo tutti lavorato con impegno e costanza al progetto nonostante le varie difficoltà, ci siamo sempre aiutati tra di noi condividendo le nostre conoscenze.

E' stato per tutti e tre il primo progetto così strutturato che abbiamo mai fatto e crediamo nel complesso di aver portato a termine un buon lavoro sia nella parte dei documenti che in quella del codice.

A seguito di queste considerazioni la nostra autovalutazione è:

| Componente del team | Valutazione |
|---------------------|-------------|
| Alessandro Cornella | 29 |
| Alessio Paolazzi | 28 |
| Morgana Pasquini | 28 |