UNIVERSITÀ DI TRENTO

Dipartimento di Ingegneria e Scienza dell'Informazione

Versione 1.0
Gruppo G11

Nome progetto :

BellHotel

Nome documento :

Report Finale

Indice

Scopo del documento

Nel seguente documento verranno riassunti vari aspetti dell'iter di progettazione della nostra WebApp BellHotel. Inizialmente verranno riportate caratteristiche e specificità degli approcci all'ingegneria del software dei seminari seguiti a lezione. Poi, ritornando al progetto, verranno analizzati l'organizzazione del lavoro, la suddivisione dei ruoli all'interno del gruppo, il carico assegnato ad ognuno dei membri e la gestione dei tempi. Si concluderà esponendo le criticità riscontrate e a seguire un'autovalutazione.

Approcci all’ingegneria del software

Seminario Blue Tensor
Blue tensor è un’azienda che si occupa dello sviluppo di soluzioni personalizzate costruite su un framework di AI. Spesso si occupa di ottimizzazione di produzioni industriali o di supporto decisionale in vari ambiti lavorativi. Il loro approccio all'ingegneria del software è fondamentalmente una strategia a cascata nelle fasi prototipali e di definizione del prodotto; che poi si evolve in un metodo agile nella fase di sviluppo. In questo modo si va incontro al rischio di arrivare in fase di sviluppo con requisiti o obiettivi poco chiari e/o di difficile realizzazione. Però essendo specializzati in IA e avendo richieste dai clienti spesso simili, potrebbero aver notato diverse congruenze nei documenti prototipali di diversi lavori. Dunque l’esperienza insieme al metodo a cascata potrebbe comunque risultare in un metodo rapido ed affidabile.

Seminario Metodo Kanban
Il Dr. York Rössler durante il seminario ci ha illustrato e fatto provare con mano i principi lavorativi del metodo Kanban; un metodo già in uso dagli anni 40 nell’industria classica, ma che ha iniziato ad essere applicato all’informatica solo dagli anni 2000. Il metodo Kanban è una filosofia di approccio al lavoro che si accosta spesso al metodo agile. Infatti si concentra sul saper contrastare imprevisti sul lavoro pur mantenendo alta l’efficienza lavorativa; convertendo casi di stallo del flusso lavorativo in occasioni per aiutare altri membri del team, per accrescere le proprie conoscenze o semplicemente per riposare. Il metodo Kanban però potrebbe risultare poco efficiente in ambienti dove sfruttare gli stalli per aiutare gli altri risulterebbe complicato con personale dai ruoli altamente specifici. Lo stesso vale gli stalli che si vogliono usare per istruire personale tramite corsi nel caso non sia possibile sostenere commercialmente questi ultimi.

Seminario IBM Cloud
IBM è una tra le aziende più importanti del settore informatico e durante il seminario ci viene illustrata la loro offerta di Cloud Computing. Grazie a quest’ultimo è infatti possibile utilizzare tramite un abbonamento uno tra i loro numerosi servizi proposti, che spaziano dai puri server fisici ai container, dalle virtual machine ai database. Dunque ad un progetto qualsiasi vengono poste davanti tantissime opportunità in grado di aggiornare o aumentare le capacità e gli scopi di un progetto. Inoltre IBM garantisce un elevato livello di sicurezza e manutenzione tramite sistemi di protezione e di backup dei dati, che in caso di problematiche inaspettate possono rivelarsi molto utili.

Seminario META
Meta è una delle più importanti imprese attive nel settore nella tecnologia e durante il seminario ci viene raccontato come l'ingegneria del software viene approcciata al suo interno. Il grande pregio di META si capisce subito essere la grande libertà; ogni lavoratore al suo interno può pressoché operare con il linguaggio da lui preferito purché sia significativo all'interno di un progetto. Questa filosofia si applica quindi anche ai metodi con cui vengono approcciati i progetti, che non seguono tutti un metodo specifico, ma quello più consono al progetto stesso o addirittura nessun metodo, seppur inconsciamente spesso si ritrovano congruenze con la metodologia agile.

Seminario U-Hopper
U-Hopper è un'azienda che si occupa di valorizzare i dati prodotti da un cliente tramite l'analisi di quest'ultimi, spesso utilizzando l'intelligenza artificiale. In U-Hopper si utilizza la metodologia agile, che viene agevolata dall'utilizzo di GitLab, piattaforma che grazie a funzionalità come il branching permette di mettere mano a piccole o grandi porzioni di codice senza intaccarne le componenti consolidate, testate e funzionanti. Ciò permette di fare diversi cambiamenti in totale sicurezza, comodità e velocità; caratteristiche fondanti del metodo agile. Per supportare l'analisi di grandi quantità di dati fanno affidamento a diversi storage, linguaggi e strumenti; tutti utilizzati in maniera molto specifica, permettendo così una grande efficienza, ma anche una più alta complessità gestionale.

Seminario RedHat
RedHat è una società multinazionale che è conosciuta principalmente per il suo supporto allo sviluppo di software open source. L'open source permette a qualunque programmatore di poter contribuire liberamente ad un progetto, che a sua volta può essere utilizzato da chiunque poiché di principio distribuito in maniera gratuita. Uno sviluppo open source giova quindi dell'alta visibilità che ottiene da questa libertà, insieme solitamente ad una qualità più alta del codice, dovuta appunto al fatto che molte persone contribuiscono al progetto. Purtroppo però commercialmente questa libertà è molto rischiosa, essendo soggetta a diversi tipi di truffa.

Seminario Microsoft
Diego Colombo, dipendente di Microsoft, ci ha spiegato durante il seminario l'importanza del testing del codice. Principalmente si è parlato dell'approccio da lui usato in ingegneria del software, ovvero i metodi TDD e BDD. Si tratta del Test-Driven Development, che si concentra sul progettare dei test automatici ancor prima del software stesso; e del Behavior-Driven Development che può essere inteso come un ibrido del ciclo TDD, dove i test vengono affiancati allo sviluppo invece tramite tools esterni invece di prepararli in anticipo. Questo approccio però, come riscontrato nelle domande poste a Diego Colombo in classe, può essere difficile da approcciare da programmatori inesperti o che non sono in grado di visualizzare ancor bene il progetto e quindi di conseguenza i test a cui sottometterlo. Però questi metodi permettono di sondare immediatamente il codice sviluppato, evitando il rischio di dover rimaneggiare il codice a sviluppo completo in caso di test fallimentari.

Seminario Sistemi Legacy Molinari
Andrea Molinari durante il seminario ci parla dei Sistemi Legacy, ovvero di obsolescenze che però continuano ad essere utilizzate. Queste obsolescenze possono spaziare dall'ambito hardware a quello software come da quello dei dati alle persone. Si è parlato di come modernizzare questi sistemi sia molto utile, per permettersi nuove opportunità di sviluppo o semplicemente di ottimizzazione del lavoro. Purtroppo molto spesso vi è resistenza nei confronti di queste migrazioni essendo molto costose, sia in ambito di risorse tecnologiche richieste che in ambito di personale specializzato. Queste problematiche sono spesso identificate col nome "tech debt". Si è parlato principalmente dei sistemi main-frame che subiscono maggiormente questi problemi anche per via della scarsa applicazione in questi ambienti del metodo agile .

Il nostro approccio
Il nostro gruppo si è rivolto al progetto con la metodologia agile. Questa si è rivelata molto efficace nei Deliverable 1, 2 e 3. Infatti questo ci ha consentito di sviluppare dei documenti di requisiti, di specifica ed architetturali molto saldi; permettendoci di procedere con scopi chiari e precisi durante il Deliverable 4. Purtroppo la scarsa esperienza ci ha limitato nell'usa di tecnologie, metodi e tool esterni mostrati durante i seminari; che riteniamo poter essere d'impatto solo per persone con competenze informatiche più profonde. Una approccio consigliato dai seminari che è tornato utile durante lo sviluppo del progetto è stato l'uso di GitHub, citato in particolar modo dal seminario di U-Hopper.

Organizzazione del lavoro

Abbiamo approcciato inizialmente il Deliverable 1 insieme, senza stabilire dei ruoli, cercando di definire chiaramente le fondamenta del progetto. In seguito, con il Deliverable 2, abbiamo iniziato a dividerci dei compiti, cercando di ottimizzare il processo di specifica del progetto. Durante il D2, osservando qualità e difetti di ognuno, siamo riusciti a definire dei ruoli più specifici, da assegnare a ciascuno, mantenendoli anche per il Derivable 3. Dopo un lungo periodo di pausa, dovuto a impegni accademici e personali, si è deciso di dividere il Deliverable 4 in Back-End e Front-End tra due membri, mentre il terzo membro avrebbe lavorato al Deliverable 5. La comunicazione tra di noi ha avuto luogo attraverso incontri diretti, in special modo durante il periodo delle lezioni. Avendo scelto di non consegnare il progetto durante la sessione invernale la comunicazione si è poi spostata principalmente online. Per comunicare rapidamente abbiamo usato WhatsApp, mentre per dialogare a distanza per lunghe sessioni si è usata la piattaforma Discord.
Per la compilazione dei Deliverable e la realizzazione dei design del Front-End nel D1 abbiamo usato rispettivamente Google Docs e Figma, mentre per realizzare i vari diagrammi abbiamo utilizzato Draw.io e Lucidchart. Questi servizi hanno agevolato la cooperazione grazie ad una feature che permetteva a tutti e tre di poter lavorare contemporaneamente su un singolo documento.

Ruoli e attività

Componente del team	**Ruolo**	**Principali attività**
Leonardo Lorenzini	Capo del progetto, Revisore, Progettista	-Organizzazione deliverable 1 e 2. -Revisione deliverable 1,2,3,5 -Stesura “Analisi dei componenti” nel D2 -Stesura del D5
Lorenzo Cortese	Progettista, Programmatore, Revisore	-Revisione di ogni deliverable -Stesura requisiti non funzionali nel D1 e nel D2 -Sviluppo diagramma delle classi nel D3 -Sviluppo, debugging e testing backend -Revisione e debugging frontend
Luca Moretto	Progettista, Programmatore, Revisore	-Revisione di ogni deliverable - Stesura dei requisiti funzionali/non funzionali nel D1 e requisiti funzionali nel D2 -Sviluppo frontend -Sviluppo userflow, modello delle risorse e diagrammi delle API nel D4

Carico e distribuzione del lavoro

Di seguito vi è una tabella che descrive quante ore sono state svolte sui deliverable da ogni membro del gruppo.

	D1	D2	D3	D4	D5	Totale
Leonardo Lorenzini	30	50	3	1	27	106
Lorenzo Cortese	20	29	15	50	2	116
Luca Moretto	25	36	3	46	1	111
Totale	75	115	21	97	30	338

Criticità

Le criticità del progetto sono principalmente da ricollegare ad una organizzazione non molto efficace. Leonardo Lorenzini, leader del gruppo, ha avuto qualche difficoltà a svolgere il suo ruolo durante la totalità del progetto, in particolare nei deliverable 3 e 4. Per risolvere questo problema Leonardo è stato affiancato da Lorenzo Cortese, il quale ha aiutato a far progredire il progetto dove il leader aveva difficoltà. Tra il periodo invernale e quello estivo sono insorte difficoltà che hanno ostacolato uno sviluppo costante del progetto, a causa di impegni accademici e personali legati ai vari membri. Questo ha portato a concludere le fasi finali del progetto verso la fine del periodo estivo. In particolare Leonardo Lorenzini, cosciente delle difficoltà non previste incontrate durante l'esercizio del suo ruolo, si assume la responsabilità dei ritardi derivanti dalle sue azioni. Lorenzo Cortese e Luca Moretto hanno cercato di far fronte alle difficoltà insorte e ai ritardi non previsti apportando supporto al progetto, oltre che durante le lezioni, durante la sessione estiva di esami. Da notare che i ritardi complessivi non sono da attribuire unicamente alle difficoltà incontrate dal leader durante lo sviluppo del progetto, ma anche dalla non conoscenza dei linguaggi di programmazione utilizzati, che hanno richiesto tempo per poter essere compresi ad un livello tale da poter essere utilizzati durante la progettazione dell'architettura e dello sviluppo del software.

Autovalutazione

In conclusione, nonostante alcuni rallentamenti nello sviluppo, siamo soddisfatti di aver portato a termine il progetto, in quanto ciò dimostra come siamo riusciti a superare ogni difficoltà che abbiamo incontrato. Quindi infine, sulla base delle considerazioni precedenti la nostra autovalutazione è:

	Voto
Leonardo Lorenzini	26
Lorenzo Cortese	28
Luca Moretto	28