

Approcci all'Ingegneria del software

Per ciascuno dei seminari descrivere sinteticamente (1/4 pagina) caratteristiche, specificità, punti di forza e criticità degli approcci all'ingegneria del software presentati. Concludere descrivendo l'approccio che avete usato e illustrate similarità/differenze con quelli visti nei seminari motivandone la scelta.

15/9 Seminario BlueTensor - Andrea

29/9 Simulazione metodo Kanban - Lorenzo

20/10 Seminario IBM - Luca

30/10 Seminario META - Luca

09/11 Seminario U-Hopper - Andrea

16/11 Seminario RedHat

Questo seminario ha trattato il tema dell'open source, esplorandone la filosofia e delineando i vantaggi significativi che l'approccio open source offre.

Uno dei pilastri fondamentali è la condivisione della conoscenza, poiché permette ai neofiti di apprendere la programmazione attraverso l'analisi diretta di codice open source. Questo non solo accresce la trasparenza, ma anche la diffusione delle competenze in modo democratico. La revisione tra pari rappresenta un'opportunità senza precedenti, grazie agli innumerevoli "occhi" che possono scrutare il codice sorgente. Questa molteplicità di prospettive contribuisce a migliorare la qualità del software, con una varietà di esperti che apportano miglioramenti e correzioni.

La meritocrazia, intesa come il riconoscimento basato sulle competenze e sull'efficacia dell'implementazione, è un principio cardine nell'ambito open source. Le idee e il modo in cui vengono implementate sono gli unici criteri di valutazione, incarnando così una forma pura di meritocrazia.

La visibilità è un altro aspetto fondamentale, poiché nel mondo open source il tuo codice parla per te. Non c'è bisogno di lunghi colloqui di lavoro; il tuo contributo all'interno di progetti noti diventa la tua vetrina professionale.

Infine, la creazione di comunità è un elemento distintivo, poiché partecipare all'open source significa far parte di una rete globale in cui è possibile fornire e ricevere aiuto dai più esperti. Questa connessione offre un supporto inestimabile, contribuendo a costruire conoscenze e a risolvere sfide complesse attraverso la collaborazione.

Una componente fondamentale per quanto riguarda l'ingegneria del software applicata al modello open source è la documentazione del codice. I nuovi utenti che decidono di iniziare a contribuire allo sviluppo devono poter consultare una documentazione chiara e completa rispetto a tutte le componenti del software in questione. Inoltre, la documentazione aiuta a rendere il codice consistente e ne facilita la modifica e la correzione di problemi. È quindi importante all'interno di un progetto open source avere una o più persone incaricate di mantenere e aggiornare la documentazione.

20/11 Seminario Microsoft - Andrea

24/11 Seminario sistemi legacy

In questo seminario è emerso quanto ancora sia importante, oltre che necessario, interfacciarsi con il mondo legacy. Si tratta infatti di un insieme di tecnologie, sistemi o applicazioni sviluppate in passato che, nonostante siano considerate obsolete o superate, conservano una rilevanza fondamentale nell'ambito economico attuale.

Le sfide principali che emergono in questo contesto riguardano l'obsolescenza, la "user experience", la persistenza inadeguata, lo sviluppo professionale, i costi di gestione e manutenzione, oltre alla carenza di competenze specifiche legate al mondo legacy. Nonostante ciò, diverse ragioni contribuiscono a far mantenere al mondo legacy una centralità nell'economia odierna. Si evidenziano tra le tante cose un investimento iniziale considerevole per un sistema funzionante, solido e affidabile, oltre alla conoscenza accumulata nel tempo. Questi e altri fattori contribuiscono a quello che viene comunemente chiamato "tech debt", ovvero il costo futuro derivante dalle scelte di progettazione o sviluppo rapide, ma non ottimali, effettuate nel presente. Questo debito può accumularsi nel tempo se non affrontato, compromettendo la flessibilità, la manutenibilità e l'evoluzione del sistema.

Esistono diverse strategie per allineare i sistemi legacy ai requisiti dei moderni sistemi informativi, che vanno dalle soluzioni più radicali, come la dismissione, a opzioni meno invasive come la migrazione, l'interazione o l'inclusione.

Nel panorama del mondo legacy, notiamo come strumenti dell'ingegneria del software consolidati come l'UML siano adattati per rispondere a esigenze specifiche, evolvendosi in linguaggi di modellazione come il SysML. In alternativa, si fa uso di linguaggi indipendenti come ArchiMate per affrontare requisiti particolari.

È interessante inoltre notare come approcci di tipo "agile" incontrino difficoltà nell'adattarsi al contesto legacy. Sebbene tali metodologie dimostrino grande utilità ed efficienza in progetti di scala ridotta o media, nel mondo legacy ci si trova spesso di fronte a sfide legate a progetti di considerevole ampiezza. Sono quindi preferibili approcci di tipo "predittivo", caratterizzati da una pianificazione accurata e dalla definizione dettagliata dei requisiti del progetto sin dalle fasi iniziali, come nel caso del modello "waterfall".

30/11 Seminario: Marsiglia - Lorenzo

4/12 Seminario APSS & Trentino.ai - Lorenzo

L'argomento di questo seminario è stato la complessità di gestire il dipartimento di tecnologie all'interno di un'azienda sanitaria: esso è solo una parte di un sistema grande e complesso, ma è indispensabile per il funzionamento dell'intero ecosistema. Questo perché il software controlla, collega e registra tutti i processi dell'azienda.

Il Dipartimento tecnologie gioca un ruolo chiave nell'attuazione di iniziative tecnologiche allineate alla strategia aziendale. La sua missione è assicurare soluzioni digitali e tecnologie sanitarie per programmare e produrre servizi per cittadini, pazienti e operatori sanitari.

Attraverso l'introduzione, sviluppo e manutenzione di tecnologie dell'informazione e comunicazione, cerca di bilanciare i bisogni aziendali, l'innovazione e le risorse disponibili. Impegna attivamente nel massimizzare i benefici delle tecnologie, promuovendo lo sviluppo di capacità organizzative e competenze tecnologiche in collaborazione con altri dipartimenti e la Direzione aziendale.

Durante la progettazione di un software dedicato a un'azienda sanitaria è fondamentale la definizione e l'attuazione dei requisiti funzionali e non funzionali. L'apparato sanitario si basa sui principi di continuità e equità di servizio, di conseguenza gli applicativi sviluppati devono essere robusti, garantire un corretto funzionamento nel tempo ed essere reperibili da qualunque luogo.

importante: CONTINUITÀ del servizio, equità del servizio
rapporto tra vecchie (anche legacy) e nuove tecnologie

il software è un tutt'uno con i requisiti funzionali
componenti:
infrastrutture - software - dati - macchine (slide 10)

Organizzazione del lavoro

	D1	D2	D3	D4	D5	TOT
Lorenzo Ferranti						
Luca Agostinelli						
Andrea Denina						
TOT						

Ruoli e attività

Carico e distribuzione del lavoro

Criticità

Autovalutazione

	Voto
Lorenzo Ferranti	
Luca Agostinelli	
Andrea Denina	