

Che cosa è l'ingegneria del software?

L'ingegneria del software (*software engineering*) è la branca dell'informatica che investiga i principi che guidano lo sviluppo dei sistemi **software grandi e complessi**. È una disciplina ancora in evoluzione.

Comprende argomenti, come la **gestione del personale**, che sono più legati alla gestione aziendale che all'informatica.

Ha l'obiettivo di studiare e sviluppare strumenti e metodologie per la pianificazione dello sviluppo, realizzazione e valutazione di applicazioni software complesse. Si occupa di teorie, metodi, strumenti per progettare, costruire e mantenere sistemi software complessi.

Il termine ISW viene introdotto in una conferenza NATO in Germania (Garmisch) del 1964 sulla crisi del software derivata dalla accresciuta potenza dei computer di terza generazione. I costi dell'hardware crollavano, quelli del software crescevano: si sapevano scrivere buoni programmi (strutturati, efficienti), non buoni sistemi.

Cerca di fornire le regole per il processo di produzione del software. Lo scopo dell'ingegneria del software è di pianificare, progettare, sviluppare il software tramite lavoro di gruppo. E' possibile che vengano rilasciate più versioni del prodotto software. Tale attività ha senso per progetti di grosse dimensioni e di notevole complessità ove si renda necessaria la pianificazione.

L'ingegneria del software si occupa principalmente di tre aspetti fondamentali: i principi, i metodi, le metodologie e gli strumenti.

Quali sono le differenze tra l'ingegneria del software e l'informatica?

L'informatica (**computer science**) si occupa delle teorie e dei metodi alla base dei sistemi software e si occupa di creare e ottimizzare algoritmi.

L'ingegneria del software (**software engineering**) si occupa di problemi pratici relativi alla produzione del software e si occupa della pianificazione e della progettazione con la finalità di ottenere un prodotto software.

Nell'ingegneria del software sono indispensabili conoscenze di informatica.

Spesso le teorie informatiche non possono essere applicate a problemi reali e complessi, e quindi interviene l'ingegneria del software.

Mentre la computer science si occupa di creare e ottimizzare algoritmi e si occupa degli aspetti teorici dell'informatica, il software engineering si occupa della pianificazione e della progettazione con la finalità di ottenere un prodotto software.

L'informatica si occupa delle teorie e dei metodi che stanno alla base dei sistemi software e di quelli informatici.

L'ingegneria del software si occupa dei problemi pratici relativi alla produzione del software.

Alcune conoscenze di informatica sono essenziali per gli ingegneri del software così come le nozioni di fisica sono essenziali per gli ingegneri elettrotecnici. In teoria, tutta l'ingegneria del software dovrebbe basarsi sui principi dell'informatica, in realtà gli ingegneri del software devono spesso trovare soluzioni ad hoc per lo sviluppo, e le eleganti teorie dell'informatica non sempre possono essere applicate ai problemi reali e complessi.

Cosa viene tollerato?

Un software o è **corretto** o è **sbagliato**. Un sistema di contabilità che è preciso con una tolleranza solo del due per cento non è accettabile.

Perché l'ingegneria del software non ha ancora una posizione ben definita?

L'impossibilità di valutare le funzionalità del software in modo quantitativo è uno dei motivi principali per i quali l'ingegneria del software non ha ancora trovato una sua posizione ben definita come l'ingegneria meccanica ed elettrica.

Che cosa è IEEE?

È l'acronimo di **Institute of Electrical and Electronic Engineers** (*Istituto degli ingegneri elettrici ed elettronici*), spesso pronunciato I triple E, è un'associazione internazionale di scienziati professionisti con l'obiettivo della promozione delle scienze tecnologiche.

Lo scopo principale dello IEEE è quello di cercare nuove applicazioni e teorie nella scienza elettrotecnica, elettronica, informatica, biomedica e delle telecomunicazioni; a questo scopo organizza conferenze e dibattiti tecnici in tutto il mondo, pubblica testi tecnici e sostiene programmi educativi. Si occupa inoltre di definire e pubblicare standard in tali campi.

Gli obiettivi primari dell'istituto sono:

- migliorare la qualità della vita dell'uomo favorendo la conoscenza e l'applicazione delle nuove tecnologie
- nobilitare la professione tecnica e dei suoi membri

Cosa bisogna fare per rendere un prodotto il più sicuro possibile?

- prestare molta attenzione nella selezione dei dipendenti
- controllare le attività degli addetti
- controlli regolari dell'infrastruttura e degli asset critici
- impostare limitazioni ai diritti di accesso
- policy rigorose per gli account
- meccanismi di autenticazione forti

Che cosa è internet-of-things?

È una tecnologia in grado di sapere sempre di cosa abbiamo bisogno. Permette di regolare qualsiasi nostro oggetto dal condizionatore al microonde, anche a distanza. Per farlo dobbiamo semplicemente collegare le nostre cose al web. Questo permette agli hacker di avere libero accesso alle nostre case.

Che cosa è la sicurezza fisica? Che cosa è l'intelligence?

La sicurezza fisica è un insieme di misure che prevengono o dissuadono gli attaccanti dall'accedere a un locale, a una risorsa o a informazioni.

L'intelligence è la raccolta proattiva e successiva analisi di notizie e dati utili alla sicurezza fisica.

Che cosa è la sicurezza?

È la **conoscenza** che l'evoluzione di un sistema non produrrà stati indesiderati e che quello che faremo non provocherà dei danni.

Security corrisponde alla sicurezza intesa come protezione da atti intenzionali che potrebbero ledere cose o persone, **safety** riguarda la sicurezza delle persone, intesa come loro incolumità.

La sicurezza è fondata sulla **fiducia** ed è un aspetto essenziale nella valutazione della Sicurezza.

Che cosa implica la progettazione di un sistema di sicurezza?

È necessario disporre di: un'**identificazione**, un'**autenticazione** e un **autorizzazione**.

La quantità di sicurezza che si può ottenere dipende da ciò a cui si può rinunciare pur di ottenerla.

La valutazione del miglior compromesso fra rinuncia e sicurezza è soggettivo (dipende cioè dal contesto)

La valutazione del compromesso deve tener conto di:

- **pericoli**: i modi in cui un malintenzionato può nuocere a un sistema
- **rischi**: una combinazione fra la probabilità che un evento pericoloso effettivamente avvenga e la gravità delle conseguenze dell'attacco, se portato a termine

Ecco alcune domande che bisogna porsi:

- Quali sono le risorse da proteggere?
- Quali sono i rischi cui sono soggette le risorse?
- In quale misura la soluzione di sicurezza in esame mitiga i rischi?
- Quali sono i rischi associati alla soluzione di sicurezza in esame?
- Quali sono i costi e i compromessi legati alla soluzione di sicurezza in esame?

Come si evolve il software?

Le operazioni di modifica (o manutenzione) del software guidano la sua evoluzione:

- correzione degli errori
- il software viene adattato a un nuovo ambiente
- il cliente richiede nuove funzionalità
- l'applicazione deve essere riprogettata per fornire benefici legati a un contesto più moderno

A mano a mano che programmi di larga scala si espandono ben oltre i 30 o 50 milioni di righe di codice, i project manager di successo hanno imparato a dedicare più tempo a riusare il codice preesistente che a creare nuovo codice.

Le leggi dell'evoluzione del software sono:

- **la legge del cambiamento continuo**: i programmi software devono essere costantemente adattati o diverranno sempre meno soddisfacenti
- **la legge della complessità crescente**: ogni qual volta un programma software evolve, aumenta la sua complessità
- **la legge della conservazione della familiarità**: a mano a mano che un programma software si evolve, tutti gli attori associati (sviluppatori, personale di vendita, utenti) devono mantenere la padronanza del suo contenuto e del suo comportamento, per garantire un'evoluzione soddisfacente. Una crescita eccessiva riduce tale padronanza e induce un temporaneo rallentamento nel tasso di crescita o addirittura una decrescita. Pertanto la crescita incrementale media rimane al più costante durante l'evoluzione del sistema.
- **la legge della crescita continua**: il contenuto funzionale dei programmi software deve essere costantemente incrementato in modo da mantenere la soddisfazione degli utenti nel corso della vita del sistema.
- **la legge della riduzione della qualità**: gli utenti percepiscono la qualità dei programmi software come in costante declino, a meno che i sistemi non vengano gestiti rigorosamente e adattati ai cambiamenti dell'ambiente operativo.

Quali sono le realtà dello sviluppo di un software?

- Gli standard possono esistere, ma sono applicati? I programmatori li conoscono? Quegli standard riflettono i metodi moderni di sviluppo del software? Sono completi? Sono adattabili? Sono stati ottimizzati in modo da migliorare i tempi di consegna pur mantenendo l'attenzione sulla qualità? In molti casi, la risposta a tutte queste domande è negativa.
- Lo sviluppo di software non è un processo meccanico come la produzione manifatturiera. Aggiungere persone a un progetto software in ritardo lo rallenta ulteriormente.
- Se un'organizzazione non sa come gestire e controllare internamente i progetti software, inevitabilmente incontrerà problemi se deciderà di appaltare all'esterno lo sviluppo.
- Non sempre è possibile un'ampia e stabile affermazione generica degli scopi, un'affermazione ambigua di tali scopi è una sicura fonte di disastri. È possibile definire requisiti non ambigui (derivati iterativamente) solo tramite una comunicazione efficace e continua fra il cliente e lo sviluppatore.
- L'effetto delle modifiche varia secondo il CVS (ciclo di vita del software) e la fase in cui vengono introdotte. Se i cambiamenti e i requisiti vengono richiesti in tempo (prima che inizi la fase di progettazione o

programmazione), l'impatto sui costi è relativamente limitato. Mano a mano che passa il tempo, l'impatto sui costi cresce rapidamente. Quando le risorse sono state allocate ed è stata definita una struttura progettuale, ogni cambiamento può richiedere ulteriori risorse e importanti modifiche alla struttura.

- "Prima cominci a stendere il codice, più tempo impiegherai a terminare il programma". Alcuni dati del settore indicano che tra il 60 e l'80% del lavoro speso su un programma avviene dopo la consegna della prima versione al cliente.
- Uno dei metodi più efficaci di esame della qualità del software, la revisione tecnica formale, si applica già dall'inizio di un progetto. Le revisioni sono un "filtro della qualità", rivelatosi più efficace del testing a posteriori nel rilevare alcuni tipi di errori del software.
- Un programma funzionante è solo una parte di un Prodotto Software che comprende più elementi. La documentazione è il fondamento di uno sviluppo riuscito e, cosa più importante, è la guida dell'attività di manutenzione.
- L'ingegneria del software non prevede la creazione di documenti. Si occupa di creare qualità. Una migliore qualità porta a minor lavoro. E un minor lavoro porta a tempi di consegna più rapidi.

Comportamenti e metodi abituali alimentano prassi manageriali e tecniche inadeguate anche quando la realtà suggerisce vie più efficaci. Il riconoscimento della realtà del software rappresenta il primo passo verso la formulazione di soluzioni pratiche ai problemi di sviluppo. Molti ingegneri del software hanno adottato un approccio "agile" che prevede modifiche incrementali con un conseguente controllo dell'impatto e dei costi.

Che cosa è un workflow?

È un'attività di lavoro che richiede l'esecuzione coordinata di più compiti da parte di più agenti. Promuove la gestione di gruppi di lavoro collaborativi secondo un modello processuale. (Un processo consiste in una o più attività, ognuna delle quali rappresenta un lavoro da svolgere per giungere a un obiettivo comune).

Gli agenti possono essere umani o artificiali. Le attività possono essere svolte dai partecipanti o da applicazioni informatiche.

I compiti possono essere manuali o automatizzati. Interpretano una serie di regole procedurali, integrano diverse funzioni per la condivisione dell'informazione e gestiscono la comunicazione e il passaggio di compiti da un collaboratore all'altro.

Che cosa sono i modelli?

I modelli sono le basi della nostra comprensione del mondo fisico e della nostra interazione con esso.

Un modello simbolico viene creato utilizzando un linguaggio che definisce il significato dei diversi simboli e le loro possibili relazioni.

Che cosa sono i dati?

I dati consistono di fatti e sono l'unità base dell'informazione. Diventano informazione se interpretati in un contesto, utilizzando la conoscenza sul contesto stesso.

Che cosa è la conoscenza?

La conoscenza è un insieme di modelli costruiti per comprendere il mondo.

Che cosa è un sistema?

È una collezione significativa di componenti interrelati che lavorano assieme per realizzare un determinato obiettivo, un modello che connette concettualmente elementi più semplici in un insieme funzionale. Così come un modello, ha una struttura definibile, un comportamento e quindi una funzione. Si basa su un modello, comprende dati e informazione. Viene sviluppato per gestire un insieme di attività; l'informazione viene utilizzata per misurare lo stato delle attività e per prendere decisioni sulle attività (azioni gestionali).

Un sistema è un insieme di parti correlate fra loro che operano in maniera congiunta per svolgere un determinato compito, è qualcosa di più della semplice somma delle parti, è una collezione significativa di componenti interrelati che lavorano assieme per realizzare un determinato obiettivo.

Che cosa è un sistema informativo?

È un insieme ordinato di elementi che raccolgono, elaborano, scambiano e archiviano dati allo scopo di produrre e distribuire, nel momento e nel luogo opportuni, le informazioni alle persone che ne hanno bisogno per lo svolgimento delle proprie attività in azienda.

Il SI incorpora informazioni provenienti da diversi dipartimenti, all'interno dell'organizzazione, ma il SI non è un sistema dipartimentale (Magazzini, ordini, contabilità, Controllo di Gestione) che è limitato nel loro scopo.

I sistemi dipartimentali sono progettati per trattare solamente il reparto e raramente contengono dati di entità provenienti da altri reparti. I sistemi dipartimentali forniscono un sottodominio del SI per le esigenze informative gestionali dell'azienda e del cliente. Un sistema informatico aziendale non è un SI.

La finalità di un sistema informativo è la gestione di informazioni utili alla misura ed alla valutazione dei processi gestionali al fine di ottimizzare le risorse impiegate nel conseguimento degli obiettivi istituzionali e ottimizzare le modalità di comunicazione.

Un **sistema informativo formale** contiene un modello approvato per l'interpretazione dei dati e i dati all'interno del sistema sono strutturati in base a questo modello.

Un **sistema informativo informale** è neutrale nei confronti dell'interpretazione dei dati e fornisce una struttura minima a tutti i dati del sistema.

Entrambi i tipi di sistemi informativi possono essere realizzati mediante supporto elettronico. Il livello di formalità dipende da considerazioni di tipo costo/beneficio.

Che cos'è un algoritmo?

Un Algoritmo è un insieme ordinato di passi eseguibili e non ambigui che giunge certamente a terminazione.

Caratteristiche:

- **Non ambiguità** (le istruzioni devono essere univocamente interpretabili dall'esecutore)
- **Eseguibilità** (l'esecutore deve essere in grado, con le risorse a disposizione, di eseguire ogni istruzione)
- **Finitezza** (l'esecuzione deve terminare in un tempo finito)
- **Correttezza**
- **Efficienza**

Un algoritmo può essere rappresentato in diversi modi e deve essere **misurabile** (Complessità degli algoritmi).

Che cosa è un Data Flow Diagram?

Il Data Flow Diagram (abbreviato in DFD) è un tipo di diagramma definito per aiutare nella definizione delle specifiche. È una notazione grafica molto usata per i sistemi informativi e per la descrizione del flusso di dati in quanto permette di descrivere un sistema per livelli di astrazione decrescenti con una notazione di specifica molto "intuitiva".

Attraverso i Data Flow Diagram si definiscono soprattutto come fluiscono (e vengono elaborate) le informazioni all'interno del sistema, quindi l'oggetto principale è il flusso delle informazioni o, per meglio dire, dei dati. Motivo per il quale diventa fondamentale capire dove sono immagazzinati i dati, da che fonte provengono, su quale fonte arrivano, quali componenti del sistema li elaborano.

Aiutano a identificare le procedure e a comunicare con il cliente in fase di analisi.