

INGEGNERIA DEL SOFTWARE

A.A. 2009/10

Ing. Mastratisi - Dott.ssa Cozza

Lezione di introduzione alla Ingegneria del Software

1



Alcune fonti

- - [Sommerville/8e] Capitolo 1, Introduzione
 - [Pressman/6e] Capitolo 1, Il software e l'ingegneria del software
 - [Ghezzi/2e] Capitolo 1, Ingegneria del software: visione d'insieme



Obiettivi e argomenti

□ Obiettivi

- introdurre l'ingegneria del software
- motivare i temi fondamentali dell'ingegneria del software

□ Argomenti

- origini della ingegneria del software
- cosa è il software
- aree di conoscenza della ingegneria del software
- processi software
- metodi e strumenti della ingegneria del software
- sfide della ingegneria del software



Ingegneria del software: origini

□ Due termini conati verso la fine degli anni '60

- *crisi del software* – problemi incontrati nello sviluppo di sistemi software complessi
- *ingegneria del software* – soluzione alla crisi del software



Crisi del software e ingegneria del software

- Contesto degli anni '60
 - software
 - da programmi (sviluppati informalmente) – ad es., per risolvere sistemi di equazioni
 - a grandi sistemi commerciali – ad es., OS 360 per IBM 360
 - gli avanzamenti nelle tecniche di programmazione (ad es., programmazione strutturata) non aiutavano a costruire sistemi software (complessi) migliori
 - necessario un nuovo approccio, ingegneristico, con strumenti e tecniche opportuni



Crisi del software e ingegneria del software

- Problemi incontrati nello sviluppo di sistemi software complessi
 - progetti in ritardo rispetto ai termini prefissati
 - sforamento del budget
 - scarsa affidabilità
 - scarse prestazioni
 - manutenzione ed evoluzione difficile
 - alta percentuale di progetti software cancellati
 - ...
- Si ipotizzò che la crisi del software era temporanea
 - poteva essere risolta con strumenti e tecniche migliori – codificati nell'ambito di una nuova disciplina – l'*ingegneria del software*



Crisi del sw: temporanea o permanente?

- Dall'introduzione di Software Systems Architecture (2005)
 - i grandi sistemi software di oggi sono tra le strutture costruite dagli uomini più complesse
 - contengono milioni di linee di codice, migliaia di tabelle nelle basi di dati, e sono eseguiti da dozzine di calcolatori
 - ciò presenta ai team di sviluppo del software delle sfide formidabili
 - se queste sfide non vengono affrontate presto, i sistemi sono consegnati in ritardo, costano più del previsto, e con un livello di qualità inaccettabilmente povero



Crisi del sw: temporanea o permanente?

- Dunque, gli stessi problemi esistono tuttora – perché?
 - mentre migliora l'abilità generale nella produzione del software, aumenta anche la complessità dei sistemi software
 - nuove tecnologie e maggiori capacità di calcolo motivano nuove esigenze e nuovi obiettivi
 - le prestazioni dell'hardware crescono secondo la legge di Moore
 - in che modo crescono le aspettative sui sistemi che comprendono hardware e software?
 - in che modo cresce la capacità di produrre software?



Che cos'è il software?

- Il **software** è/comprende
 - istruzioni (programmi) che quando eseguite svolgono una funzione desiderata con prestazioni desiderate
 - strutture di dati mediante le quali i programmi possono manipolare le informazioni in modo adeguato, e
 - documenti che descrivono le operazioni e l'uso dei programmi
 - ma anche la documentazione associata – come requisiti, modelli di progetto, ulteriori scelte di progetto, suite di test, ...



Tipi di prodotti software

- Alcune grandi categorie di software
 - software di sistema – ad es., un SO
 - software applicativo
 - software scientifico o per l'ingegneria
 - software embedded
 - linee di prodotti software
 - applicazioni web
 - software per l'intelligenza artificiale



Tipi di prodotti software

- Un'altra classificazione – due tipi fondamentali di prodotti software
 - *prodotti generici* – sviluppati per il mercato generale, per essere venduto a tutti i clienti interessati
 - ad es., MS Excel o Mozilla Firefox
 - *prodotti personalizzati (a richiesta)* – sviluppati per un cliente particolare, sulla base delle sue specifiche
- Nuovo software può essere creato sviluppando nuovi programmi, configurando dei sistemi software generici oppure riusando software esistente



Tipi di sistemi che comprendono software

- Il software viene utilizzato in sistemi più ampi
 - *sistemi tecnico-informatici* – comprendono hardware e software – ma non procedure e processi
 - **sistemi socio-tecnici** – comprendono uno o più sistemi tecnici, processi operativi (che definiscono, tra l'altro, il loro uso) e persone – e sono governati da politiche e regole aziendali e/o nazionali
- Le proprietà complessive dei sistemi socio-tecnici (ad es., comportamento e raggiungimento di obiettivi aziendali) dipendono sia da proprietà dei singoli componenti, che dalle loro relazioni (ad es., il modo in cui le persone usano il software)
 - gli ingegneri del software devono avere alcune conoscenze dei sistemi socio-tecnici e dell'ingegneria dei sistemi



Quali le caratteristiche distintive del sw?

- Il software è diverso da altri prodotti del lavoro umano
 - il software non è vincolato da materiali, né governato da leggi fisiche o da processi manifatturieri
 - il software si sviluppa – non si fabbrica nel senso tradizionale
 - il software non “si consuma”
 - tuttavia, di “deteriora”
 - sebbene l’industria si rivolge sempre più verso l’assemblaggio di componenti, la maggior parte del software continua ad essere realizzato in modo specifico



Software e sviluppo di nuovi prodotti

- Spettro delle attività ingegneristiche
 - da una parte, *produzione di massa* (produzione prevedibile)
 - ad es., di telefoni cellulari
 - dopo averne sviluppati e costruiti alcuni modelli, la costruzione di nuovi modelli può avvenire in modo prevedibile
 - dall’altra, *sviluppo di nuovi prodotti* (progetti creativi)
 - soluzione di problemi che richiede un alto grado di originalità, creatività, cambiamento – poiché non ci sono casi precedenti identici o simili da cui derivare stime affidabili
- Lo sviluppo del software, nella maggior parte dei casi, è sviluppo di nuovo prodotto
 - non un problema di produzione di massa o prevedibile



Che cos'è l'ingegneria del software?

- L'**ingegneria del software** [Swebok, IEEE] è
 - l'applicazione di un approccio sistematico, disciplinato e quantificabile allo sviluppo, l'esercizio e la manutenzione del software
 - ovvero, l'applicazione dell'ingegneria al software
- Dunque
 - una disciplina ingegneristica
 - interessata allo sviluppo del software
 - interessata a tutti gli aspetti della produzione del software – dalla specifica alla manutenzione, dagli aspetti tecnici a quelli economici e di gestione del progetto e delle persone
 - interessata allo sviluppo efficiente di software di alta qualità
 - diversa da altre discipline ingegneristiche
 - a causa delle caratteristiche distintive del software



Aree di conoscenza dell'ingegneria del sw

- Lo SWEBOK identifica dieci aree di conoscenza relative all'ingegneria del software
 - requisiti del software
 - progettazione del software
 - costruzione del software
 - verifica del software
 - manutenzione del software
 - gestione delle configurazioni del software
 - gestione dell'ingegneria del software
 - processi dell'ingegneria del software
 - strumenti e metodi dell'ingegneria del software
 - qualità del software



Aree di conoscenza dell'ingegneria del sw

- Requisiti del software
 - un requisito è una proprietà che deve essere esibita nella risoluzione di un problema del mondo reale
- Progettazione del software
 - la progettazione (design) è il processo di definizione dell'architettura, dei componenti, delle interfacce e di altre caratteristiche di un sistema o componente
 - il progetto (design) è il risultato del processo di progettazione
- Costruzione del software
 - la costruzione del software si riferisce alla creazione dettagliata di software funzionante e significativo, attraverso una combinazione di codifica, verifica, test unitari, test di integrazione e debugging



Aree di conoscenza dell'ingegneria del sw

- Verifica (testing) del software
 - il test del software consiste nella verifica dinamica del comportamento di un programma sulla base di un insieme finito di test case, scelto opportunamente da un dominio di esecuzione solitamente infinito
- Manutenzione del software
 - manutenzione del software in “operazione”, per gestire anomalie, cambiamenti nell'ambiente operativo e nuovi requisiti utente
- Gestione delle configurazioni del software
 - disciplina che riguarda l'identificazione delle configurazioni del software in momenti temporali diversi, al fine di controllare in modo sistematico cambiamenti della configurazione e di mantenere l'integrità e la tracciabilità della configurazione durante tutto il ciclo di vita del sistema



Aree di conoscenza dell'ingegneria del sw

- Gestione dell'ingegneria del software
 - riguarda la gestione e la misurazione dell'ingegneria del software
- Processi dell'ingegneria del software
 - riguarda definizione, implementazione, valutazione, misurazione, gestione, cambiamento e miglioramento del processo di ingegneria del software stesso
- Strumenti e metodi dell'ingegneria del software
 - riguarda metodi (informali, formali e basati su prototipazione) dell'ingegneria del software e strumenti di supporto alle diverse aree dell'ingegneria del software
- Qualità del software
 - riguarda considerazioni sulla qualità del software che trascendono i processi per la gestione del ciclo di vita del software



Ingegneria del software

- L'ingegneria del software comprende un corpo, vasto e in continua evoluzione, di tecniche e strumenti
 - non esiste un approccio singolo "ideale" all'ingegneria del software
 - le nozioni fondamentali ai vari approcci costituiscono l'essenza dell'ingegneria del software
- Gli ingegneri del software
 - adottano un approccio sistematico ed organizzato al loro lavoro
 - usano tecniche e strumenti appropriati
 - conoscono diverse tecniche e strumenti
 - sanno scegliere il metodo più adatto a determinate circostanze



Differenza tra ing. del sw e informatica?

- L'informatica si occupa di teorie e metodi che stanno alla base dei sistemi software e dei sistemi informatici
 - l'ingegneria del software si occupa dei problemi pratici relativi alla produzione del software
 - perciò anche di aspetti non strettamente informatici
- In teoria, l'ingegneria dovrebbe basarsi sui principi dell'informatica
 - tuttavia, le teorie (eleganti) dell'informatica non sempre si applicano ai problemi reali e complessi
 - spesso vanno sviluppate soluzioni ad hoc



Che cos'è l'ingegneria dei sistemi?

- L'**ingegneria dei sistemi** si occupa di tutti gli aspetti dello sviluppo e dell'evoluzione di sistemi complessi – in cui spesso il software gioca un ruolo chiave
 - ad es., un sistema di controllo del traffico aereo
 - si occupa anche dell'hardware e del suo eventuale sviluppo, della progettazione di politiche e processi, della messa in opera del sistema e, tra l'altro, dell'ingegneria del software
 - l'ingegneria dei sistemi è interessata alla definizione delle specifiche del sistema e della sua architettura generale, ed alle problematiche di integrazione delle diverse parti necessarie alla realizzazione del sistema
 - ma è meno interessata all'ingegneria dei singoli componenti
- L'ingegneria del software è spesso parte di un processo più ampio



Che cos'è un processo software?

- Un **processo software** è l'insieme delle attività e dei risultati che creano un prodotto software
 - ad es., XP, UP
- Un processo definisce *chi* fa *che cosa*, *quando* e *come* per raggiungere un certo obiettivo [Jacobson, Booch, Rumbaugh]



Che cos'è un processo software?

- Attività fondamentali comuni a tutti i processi software
 - *specifica* – definizione di che cosa deve fare il software e dei vincoli per il suo sviluppo
 - *sviluppo* – progettazione e programmazione
 - *validazione* – verifica che il software prodotto sia conforme alle richieste del suo committente
 - *evoluzione* – modifica del prodotto per adeguarlo ai requisiti dell'utente e del mercato che cambiano
- Processi diversi organizzano queste attività in modi diversi
- Perché esistono diversi processi software?
 - perché contesti diversi (tipo di sistema, tipo di committente, caratteristiche degli sviluppatori e del team di sviluppo, tempo, budget, ...) richiedono tipi differenti di processi di sviluppo



Che cos'è un modello di processo software?

- Un **modello di processo software** è una descrizione semplificata di un processo (o di un suo aspetto) – osservato da un determinato punto di vista
- Alcuni punti di vista possibili
 - **modello a flusso di lavoro (workflow)** – il processo è una sequenza di attività (azioni), con relativi input, output e dipendenze
 - **flusso di dati o modello di attività** – il processo è un insieme di attività (trasformazioni), che modificano delle informazioni
 - ad es., come trasformare le specifiche in progetto
 - **modello ruolo/azione** – rappresenta i ruoli delle persone coinvolte e le attività di cui sono responsabili



Modelli generici di processi software

- Molti modelli di processo sono basati sui seguenti modelli generici (paradigmi) per lo sviluppo del software
 - **approccio a cascata** – le attività sono fasi di lavorazione separate – un'attività inizia quando le precedenti sono concluse e “firmate per accettazione”
 - **sviluppo iterativo** – il processo è organizzato in iterazioni, in cui vengono eseguite varie attività
 - **ingegneria del software basata su componenti** – tecnica che presuppone l'esistenza di alcune parti del sistema, e che concentra lo sviluppo sull'integrazione di queste parti, anziché sullo sviluppo dal nulla



Che cos'è un metodo dell'ingegneria del sw?

- Un **metodo dell'ingegneria del software** è un approccio strutturato relativo ad uno o più aspetti (tecnici e focalizzati) dello sviluppo del software, che ha l'obiettivo di facilitare la produzione di software di alta qualità
 - i metodi costituiscono il sapere tecnico relativo allo sviluppo del software
 - ad es., il metodo OO di Larman, progettazione di basi di dati con ER
 - metodi diversi hanno aree di applicabilità diverse
 - non esiste un metodo ideale
- I metodi dell'ingegneria del software comprendono
 - modelli (linguaggi) – con la loro sintassi e semantica
 - raccomandazioni e linee guida – a vari livelli di dettaglio



Quali gli strumenti per l'ingegneria del sw?

- **CASE** significa *Computer-Aided Software Engineering*
 - gli strumenti CASE sono sistemi software che forniscono un supporto automatico o semiautomatico alle attività dei processi software
 - di solito supportano uno o più metodi
- Varie funzionalità (integrabili o meno)
 - requisiti e loro gestione
 - creazione di modelli
 - programmazione e testing
 - generatori di codice
 - guida del processo
 - ...



Processi software, metodi e strumenti

- L'ingegneria del software è una tecnologia stratificata



- un processo software è un framework
 - che definisce un contesto in cui applicare i metodi, per creare prodotti ed elaborati intermedi
 - con il supporto di opportuni strumenti
- con un impegno alla qualità



Quali sono i costi dell'ingegneria del sw?

- I costi dell'ingegneria del software dipendono da molti fattori
 - ad es., dal tipo di sistema da sviluppare, dalla sua complessità e dalle qualità richieste
- La ripartizione dei costi dipende anche dal tipo di processo usato
- Alcune osservazioni
 - relativamente alla fase di sviluppo del software, i costi legati alla verifica possono andare dal 25% al 50% ed oltre – soprattutto nel caso di sistemi critici e/o sviluppo basato su componenti
 - per sistemi di lunga vita, i costi di manutenzione ed evoluzione successivi al rilascio possono essere il 300% ed oltre dei costi di sviluppo iniziale



Quali le caratteristiche di un buon sw?

- Un prodotto software ha – oltre alle *funzionalità* – diverse caratteristiche associate che ne riflettono la qualità
 - *affidabilità* – il software deve effettivamente fornire le funzionalità richieste
 - *disponibilità* – il software deve essere funzionante in modo continuato nel tempo
 - *efficienza* – il software deve fornire le prestazioni desiderate
 - *sicurezza* – il software deve fornire protezione alle sue funzionalità ed ai suoi dati
 - *mantenibilità* – il software deve poter evolvere per soddisfare i requisiti che cambiano dei suoi clienti
 - *verificabilità* – deve poter essere possibile verificare le funzionalità e le altre qualità del software
 - *usabilità* – il software deve essere facilmente utilizzabile dagli utenti per il quale è stato sviluppato
 - ...



Quali le sfide chiave per l'ing. del sw?

- L'ingegneria del software del XXI secolo deve affrontare diverse sfide chiave
 - *eterogeneità* – far interagire sistemi distribuiti che sono eterogenei – per piattaforma, rappresentazione dei dati, ambiente di esecuzione, ... – consentendo un accesso ubiquitario e pervasivo
 - *consegna e flessibilità* – sviluppare tecniche per rispondere con velocità alla consegna del software ed alle richieste di cambiamento
 - *fiducia* – è essenziale potersi fidare del software
 - ...
 - *agilità di business* – poter creare o modificare dinamicamente processi di business – e le applicazioni software che li supportano – in modo da reagire in modo competitivo nel proprio settore di business



Che cosa sono le architetture software?

- L'*architettura di un sistema software-intensive* è la struttura o le strutture del sistema, che comprendono elementi software, le proprietà visibili esternamente di questi elementi, e le relazioni tra di essi
- Perché è importante l'architettura software di un sistema?
 - l'architettura software riconosce l'importanza di tutte le parti interessate allo sviluppo del sistema e dei loro interessi
 - tiene conto delle qualità del sistema (requisiti non funzionali)
 - prestazioni, sicurezza, verificabilità, evolvibilità, ... sono spesso qualità cruciali per la soddisfazione del cliente
 - si occupa di criteri di decomposizione per gestire la complessità del sistema – sulla base di soluzioni pre-esistenti, principi ed euristiche



Quali sono i miti del software?

- Miti del software
 - opinioni diffuse – ma in realtà atteggiamenti fuorvianti
 - ovvero – perché l'ingegneria del software non è inutile?



Miti del software

□ Miti del management

- abbiamo già interi volumi di standard e procedure da seguire nello sviluppo del software – non c'è forse tutto l'indispensabile?
- se siamo in ritardo, possiamo sempre recuperare aumentando il numero di programmatori
- se decido di far realizzare un progetto software a una terza parte, posso restare tranquillo perché tutto il lavoro verrà svolto esternamente



Miti del software

□ Miti della clientela

- un'affermazione generica degli scopi è sufficiente per iniziare a scrivere i programmi – i dettagli si possono trattare in seguito
- i requisiti di un progetto mutano di continuo, ma i mutamenti si gestiscono agevolmente grazie alla flessibilità del software



Miti del software

□ Miti del programmatore

- una volta scritto e fatto funzionare il programma, il nostro lavoro è finito
- fino a quando il programma non può essere eseguito, non c'è modo di valutarne la qualità
- il solo prodotto di un progetto concluso è il programma funzionante
- l'ingegneria del software ci farà scrivere un'inutile e voluminosa documentazione che inevitabilmente rallenterà le cose



Risposta ai miti del software

□ L'ingegneria del software

- non si occupa di creare di documenti
- piuttosto, si occupa di creare qualità
 - una migliore qualità porta a minor lavoro
 - un minor lavoro porta a tempi di consegna più rapidi