Economia di scala: l'economia di scala si riferisce ai vantaggi economici ottenuti quando si produce o si gestisce una quantità maggiore di un prodotto o un servizio 🡪 più produco e più abbatto i costi.

Cos’è un software

Esistono due categorie principali di servizi:

* Programma: l’autore è anche l’utente, non è documentato, quasi mai è testato e non c’è progetto.
* Prodotto software: usato da persone diverse da chi lo ha sviluppato. È software industriale il cui costo è circa 10 volte il costo del corrispondente programma. Bisogna avere un approccio formale allo sviluppo. Per questo il prodotto industriale costa di più.
  + prodotti generici: sistemi prodotti da una organizzazione e venduti a un mercato di massa.
  + prodotti specifici: sistemi commissionati a un committente e sviluppati specificamente per questo da qualche contraente.

Mi costa molto di più sviluppare un prodotto specifico poiché ci sono dei costi da tenere in considerazione. Quando sviluppo un prodotto per mercato di massa devo sviluppare un piano per prevedere quando verrà raggiunto il break event (pareggio delle spese).

Si definisce la differenza tra software e prodotto software:

* il software: insieme di programmi, procedure e, possibilmente, documentazione e dati correlati relativi al funzionamento di un sistema informatico.
* Il prodotto software è la parte del software che viene rilasciata.

Natura del software

Un software differisce dal prodotto industriale classico poiché è intangibile (non è una cosa che fisicamente si può toccare), malleabile, manutenibile (non solo fisicamente).

Problemi della produzione software:

* Costi. Il software ha costi elevati. Sono i costi delle risorse usate per ore di lavoro (man power), hardware, software e risorse di supporto. Il testing impiega fino a 50% del tempo di sviluppo e la manutenzione costa più dello sviluppo.
* Ritardi: molti progetti sono “runaway” (in ritardo o fuori dal budget e lo schedule stimato).
* Abbandoni: molti progetti vengono abbandonati a causa di ritardi e costi lievitati o stimati non correttamente.
* Affidabilità: il software è spesso inaffidabile. Molti malfunzionamenti sono rilevati durante l’operatività del sistema.

Ingegneria del software

L’ingegneria del software riguarda la costruzione di software di grandi dimensioni, notevole complessità e sviluppati tramite lavoro di gruppo. Tipicamente progetti software di questi tipo hanno versioni multiple, lunga durata e frequenti cambiamenti (eliminazione difetti, adattamento a nuovi ambienti, miglioramenti). La maggior parte del software è collocata all’interno di un “sistema” misto tra software e hardware: l’obbiettivo finale di chi produce è quello di creare un sistema che soddisfa globalmente i requisiti dell’utente.

* Necessario la conoscenza del dominio applicativo: il software è utile quando riesce a condensare nei suoi algoritmi la conoscenza del dominio applicativo.

È ingegneria del software:

* l’applicazione di una strategia sistematica, disciplinata e misurabile allo sviluppo, esercizio e manutenzione del software.
* La disciplina tecnologica e manageriale che riguarda la produzione sistematica e la manutenzione dei prodotti software che vengono sviluppati e modificati entro i tempi e costi preventivati.

L’ingegneria del software si occupa dei metodi e delle metodologie, dei processi e degli strumenti per la gestione professionale:

* Metodo (o tecnica): procedimento generale per risolvere classi di problemi specificati di una volta in volta. Sono presenti delle linee guida o regole che governano le attività.
* Metodologia: insieme di principi, metodi, degli elementi di cui una o più discipline si servono per garantire la correttezza e l’efficacia del proprio procedere.
* Strumento (tool): un artefatto, un sistema per fare qualcosa in modo migliore.
* Procedura: combinazione di strumenti e metodi che assieme permettono di produrre un certo prodotto.
* Paradigma: un particolare approccio o filosofia per fare qualcosa.

Processo

Un processo è un particolare metodo per fare qualcosa costituto da una sequenza di passi che coinvolgono attività, vincoli e risorse.

* Processo software: un metodo per sviluppare del software 🡪 insieme organizzato di attività che sovraintendono alla costruzione del prodotto da parte del team di sviluppo utilizzando metodi, tecniche, metodologie e strumenti. Suddiviso in varie fasi secondo il **ciclo di vita del software** e descritto da un **modello** informale, semi-formale o formale.

Classi di progetti

Il modello delle "Classi di Progetto" di Barry Boehm è un concetto utilizzato nella gestione dei progetti software per classificare i progetti in base alla loro dimensione, complessità e rischio. Questa classificazione aiuta i responsabili di progetto a identificare le sfide specifiche associate a ciascun tipo di progetto e ad adottare le strategie di gestione più adeguate. Necessità del project management (project planning, project monitoring e controllo e metriche di prodotto e processo per misurare e controllare).

CASE (Computer-Aided Software Engineering)

Si definiscono CASE i sistemi software che intendono fornire un supporto automatico per le attività di un processo software.

* Upper-CASE: strumenti che supportano le attività delle fasi di analisi e specifica dei requisiti e progettazione di un processo software. Includono editor grafici per sviluppare modelli di sistema e dizionari dei dati.
* Lower-CASE: strumenti che supportano le attività delle fasi finali del processo, come programming, testing e debugging. Includono generatori di graphical UI per costruire interfacce utente, debuggers e traduttori automatici per generare nuovi versioni di un programma.