# Lezione 1

Il software, se utile, ha vita lunga. Il ruolo di un informatico è "al contorno della programmazione", in modo da non produrre codice usa e getta; deve **mantenere** il software e il codice. Tutte le attività hanno un costo e non possono essere buttate al vento.
**Engineering**, ingegnere e scienziato; l'ingegnere è colui che dai principi scientifici ne trae una finalità concreta (pratica). La scienza deve scoprire questi principi, prepararli. L'ingegnere, sapendo che la scienza esiste, la applica. Crea qualcosa che sia mantenibile nel tempo. Esempio di engineering: piramidi, acquedotti, ponti....
**Software**, è un terreno molto giovane, nato nella seconda guerra mondiale (40 – 45). Il *SWE* molto dopo. Il *SWE* non è un ramo di *computer science* (ramo della scienza che spiega perché i computer sono utili) ma è una disciplina ingegneristica che mette insieme elementi e conoscenze. In assenza di principi ingegneristici si "fanno dei pasticci". Molte conoscenze vengono da campi che non sono l'informatica:

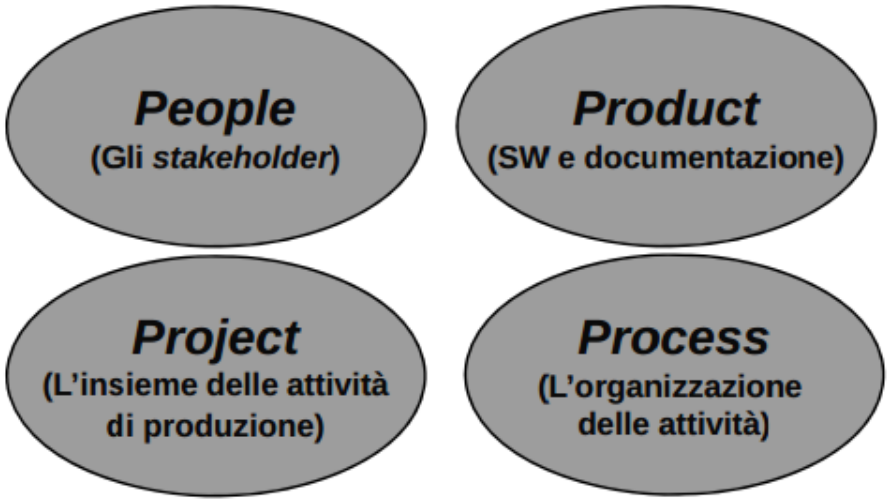
- **Informatica**,si vuole che un buon ingegnere del software conosca TUTTE le competenze informatiche;
- **Matematica**, scienza di base che aiuta a risolvere problemi;
- **Scienze gestionali ed economia**, è un attività di gruppo, correlazionale, capire come gestire risorse, tempo, denaro, cogestire;
- **Ingegneria**, è un pezzo di un sistema complesso che passa informazioni.
- **Psicologia**, il sw è rilasciato con interfacce molto orientate alle persone, bisogna intercettare le aspettative di chi usa il sw.

**Ciclo di vita di un software**, per regola il sw nasce e ha una vita che termina con il **ritiro**. Dal momento in cui nasce esso passa in diversi **stati**, che lo fanno transire. Il sw spende la maggior parte del suo tempo in uno stato che si chiama manutenzione, in cui si possono correggere degli errori e cambiare lo stato. Noi vorremmo una manutenzione che sia il meno invasiva possibile.
**Efficienza**, quante risorse ho impiegato per fare ciò che ho richiesto; misura il consumo e cresce al diminuire del consumo. La massima efficienza è il consumo zero, quindi da sola non basta. Le risorse che si consumano sono persone, tempo, denaro, materiale.
**Efficacia**, è una misura della conformità, il raggiungere l'obiettivo atteso. Si è efficaci se si raggiunge con rapidità l'obiettivo; non misura le risorse.
Bisogna cercare l'ottimo tra efficienza ed efficacia, trovare il massimo equilibrio, massimizzare gli obiettivi che sono tra l'efficienza e l'efficacia. Questi due termini sono in contrasto tra loro.
**Progetto**, *assignment* (incarico), sono progetti tipo P2, basi, Tw... qui si parla di un incarico contrattualizzato fra parti e non più negoziabile, mentre tutto il resto è negoziabile. Verrà enfatizzato molto l'aspetto contrattuale. *Assignment + commitment* (impegno inderogabile); il lavori vengono dati da un assignement e trasformati in commitment.
**Engagement**, essere impegnati formalmente, impegno dal quale non si può fallire. A volte i progetti falliscono clamorosamente, per un sacco di motivi che il swe dovrebbe cercare alla radice. Il progetto potrebbe essere obsoleto dalla nascita, incapacità di chi ha firmato l'impegno, oppure esaurimento di tempo e/o finanziamenti.
In questo caso vogliamo imparare come si fa a *non fallire*, a soddisfare gli obiettivi entro i tempi e i costi noti a priori. Quante ore produttive mi servono e in quante *ore di calendario* posso soddisfare determinati obiettivi (a prescindere dalle persone). Per fare ciò devo applicare principi ingegneristici. Per imbarcarsi in un progetto devo sapere che *ce la posso fare*.
**Best practice**, il miglior modo di fare le cose, la selezione di ciò che è meglio fare un una certa professione.
Il termine SWE nasce nel 1968, 26 anni dopo la nascita del primo software. Dopo 45 anni la situazione è ancora terribile, la differenza è che i costi sono minori ma i prodotti buoni sono pochi.

**Stakeholder**, è una persona che conta, un portatore di interesse, che ha influenza sul prodotto.

L'approccio **sistematico**, **disciplinato** e **quantificabile** allo sviluppo, uso, manutenzione e ritiro del software

**Sistematico**, agisco secondo un sistema, affronto il medesimo problema sempre nello stesso modo (nel modo giusto); niente improvvisazione o creatività.
**Disciplinato**, ciascuno fa il suo e nessuno fallisce, perchè il swe è un'attività collaborativa. Seguire in modo rigoroso la disciplina è nostro dovere, produce affidabilità.
**Quantificabile**, esprimere una quantità; se sono sistematico e disciplinato saprò dire a priori quanto tempo mi ci vorrà per fare una cosa. Si può essere quantificabili solo se si è sistematici e disciplinati. Noi cerchiamo un approccio con queste caratteristiche.




- **People**, insieme delle persone che commissionano e ricevono;
- **Product**, il sw è parte di un sistema complesso;
- **Process**, l'insieme di attività che svolgiamo ad un particolare fine;
- **Project**, attività specifiche svolte a fronte di un assignment che diventa un commitment.

Tre strati:

- **Customer**, il cliente, quello che ha un bisogno espresso o inespresso; due poli, gli stakeholder che costruiscono le *opportunità*;
- **Solution**, dove esiste il customer c'è qualcuno che trova una soluzione, fatta di **requisiti** (di cosa c'è bisogno);
- **Endeavor**, l'impegno per realizzare il sw, il team interagisce con gli stakeholder; il swe è un'attività strettamente collaborativa, ho bisogno di un metodo di lavoro per il team. *Way of working.*

**SWE != PROGRAMMING**, la programmazione è solo un elemento, e anche il meno importante. Il programmatore deve fare solo quello che viene chiesto (da un membro del team stesso). Il programmatore deve obbedire, non può essere creativo.
Principi etici:

- Considerare la qualità come primo obiettivo;

- Produrre sw di qualità è possibile;
- Aiutare il cliente a comprendere i suoi veri bisogni;
- Adottare i processi più adatti al progetto;
- Ridurre la distanza intellettuale tra il sw e il problema da risolvere;
- Essere proattivi nel cercare e rimuovere gli errori;
- Motivare, formare, far crescere le persone.