

## Premesse al Corso(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 manutenibile nel tempo.*

**Software**, è un terreno molto giovane, nato nella seconda guerra mondiale (40 – 45). Il *SWE* molto dopo.

**SWE:** **Def 1** l'insieme di regole e procedure di loro attuazione che vanno conosciute all'origine e applicate.

**Def 2** L'approccio **sistematico, disciplinato e quantificabile** allo sviluppo, uso, manutenzione e ritiro del software.

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.

- **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:** **Def** Stati che il prodotto assume dal concepimento al ritiro. Il sw spende la maggior parte del suo tempo nello stato di manutenzione (correttiva, adattativa, evolutiva) in cui si possono correggere degli errori e cambiare lo stato. Noi vorremmo una manutenzione che sia il meno invasiva possibile.

**Efficienza** **Def** Inversamente proporzionale alla quantità di risorse impiegate nell'esecuzione delle attività richieste.

**Efficacia** **Def** Misura della conformità rispetto alle norme vigenti. Si è efficaci se si raggiunge con rapidità l'obiettivo; non misura le risorse.

Il goal è massimizzare sia l'efficacia che l'efficienza, per farlo **Best Practice**

**Def** Prassi che per esperienza o per studio mostra di garantire i migliori risultati.

**SWE:** 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 (best); niente improvvisazione o creatività.
- **Disciplinato:** la SWE è una disciplina collaborativa. Seguire in modo rigoroso la disciplina, produce affidabilità.
- **Quantificabile:** le azioni che compiamo seguendo i due principi precedenti devono essere quantificabili in Efficacia ed Efficienza a priori (proattivamente).



**SWE** != **PROGRAMMING**, la programmazione è solo un elemento, il meno importante e non può essere creativa.