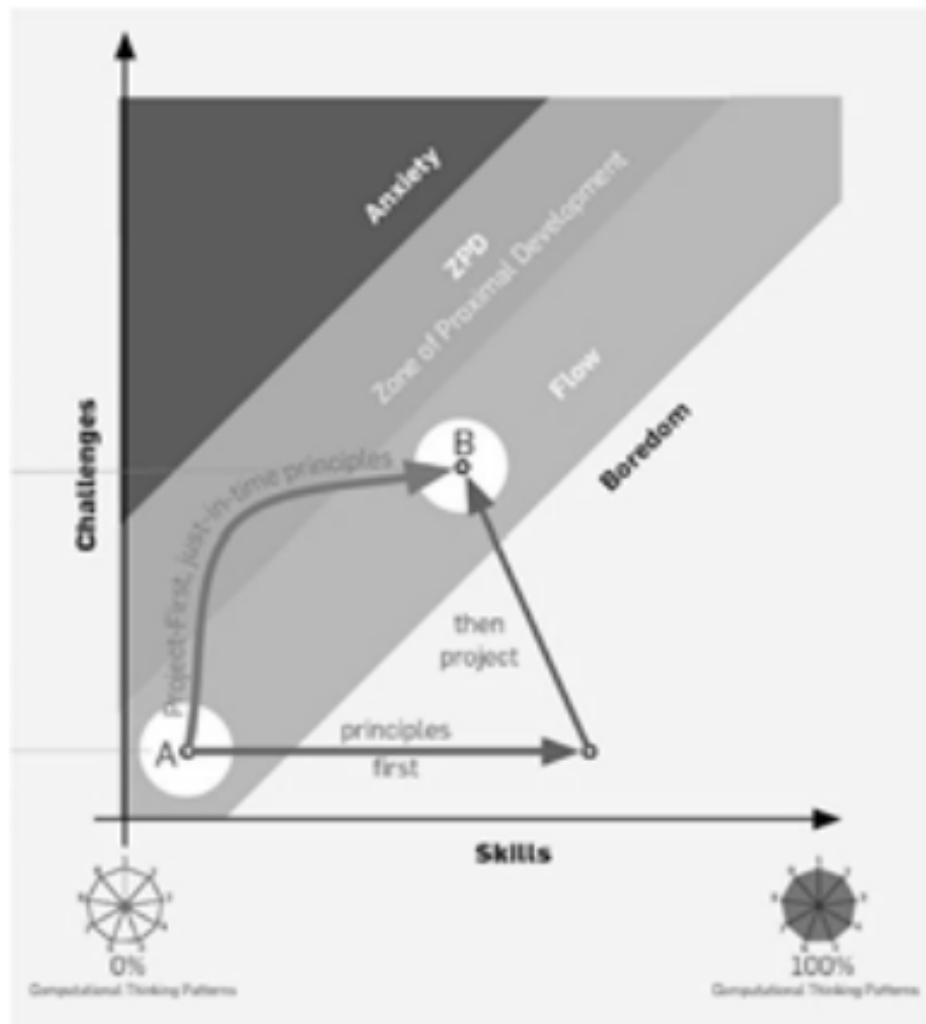


## Lezione 2

La tecnologia evolve ad una velocità incontrollabile, diventa obsoleta rapidamente. Bisogna capire la differenza tra le cose **essenziali** e le cose **accidentali**. Tra le cose essenziali c'è la disciplina, tra le cose accidentali c'è la tecnica e gli strumenti, E' importante concentrarsi sugli aspetti fondamentali e non accidentali. Occorre avere capacità di impegno concettuale, astrazione, di analisi e rigore (*best practice*). Vogliamo utilizzare la traccia di chi ha fatto le cose in passato e le ha **fatte bene**. Queste cose sono irraggiungibili senza impegno.

**Progetto didattico**, ci sono due approcci alternativi ed uno spazio intermedio. Bisogna fornire competenze per affrontare *challenges* (sfide) sempre più alte. Il metodo di insegnamento deve portare a trovare il punto di ottimo.



Nello spazio bianco c'è la **noia sicura**. Il principio opposto è la **sfida immediata**, in cui bisogna fare leva sui principi. C'è una cosa da fare e questa cosa è **ambiziosa**.

**Cosa non è un progetto**, mettere un accrocchio di cose che sembrano funzionali (progetti fatti finora), non è il "*basta che respiri*", è opporre il principio di **by correction** a quello di **by construction** (costruire sapendo che funzionerà). In questo modo si fa fatica subito ma meno in seguito. Niente correzioni in fase di sviluppo.

Cosa è un progetto:

- **Pianificazione**, pianifica chi sa cosa vuole fare, è l'essenza per controllarsi, per sapere se stiamo convergendo o divergendo. Ogni attività inizia con la **pianificazione**;
- **Analisi dei requisiti**, si analizza ciò di cui si ha bisogno. L'analisi ha un'importanza decisiva, bisogna capire il problema;
- **Progettazione**, si decide la forma della soluzione. Solo dopo si passa ad alla...
- **Realizzazione**, dove sta anche la programmazione, che deve aderire al 100% alla progettazione. Nella realizzazione attuo, ma non sono ancora sicuro che il risultato soddisfi il cliente. Per cui...
- **Verifica e validazione**;
- **Manutenzione**, per la maggior parte della sua vita il prodotto resta in manutenzione. Non esiste vita operativa senza manutenzione, non esiste sw perfetto quindi esso deve essere mantenuto.
- **Qualità**, uno dei principi su cui punteremo, perchè la qualità è possibile. Vorremmo quantificare la qualità in modo oggettivo.

**Libri**, “*Fare imparando*”. I libri aiutano poco e servono solo come appoggio o approfondimento. I concetti vanno vissuti nel progetto. **Libri teorici**, servono solo nel piano delle skills. **Esperienziali**, “*si fa così perchè l’ho fatto*”.

E’ importante costruire il proprio **glossario**. **SWEBOK** (swe body of knowledge), non è un libro per apprendere ma per risolvere dubbi; strutturato in 10 aree di conoscenza. L'essenza dell'esame sarà il progetto didattico da svolgere da metà Novembre fino alla fine dell'anno (primo appello a Marzo). Progetto di gruppo min 6 persone max 7. 100 ore di studio individuale.

**Attività di gruppo**, attività ripartite, non terrà conto delle abilità, nessuno è essenziale, è un danno all'efficienza ma è un beneficio didattico. Nessuno fa cose in autonomia, deve essere tutto deciso secondo un piano **regolato**. Nessuno fa in modo estroso ma disciplinato e sistematico.

**Analisi del problema da risolvere**, concentrarsi sulle cose essenziali e trovare una soluzione buona per costruzione.

**Approccio ingegneristico**, disciplinato, sistematico, quantificabile.

**Processi software**, attività coordinate, processi di ciclo di vita per far evolvere il sw da uno stato all'altro. Il sw è una macchina a stati:

- **Concezione**
- **Sviluppo**
- **Utilizzo**
- **Ritiro**

Nella fase di ritiro il sw cessa di esistere nel senso che non c'è più alcun tipo di supporto per quel prodotto. Le transizioni sono strettamente e formalmente regolate.

**Modelli di ciclo di vita**

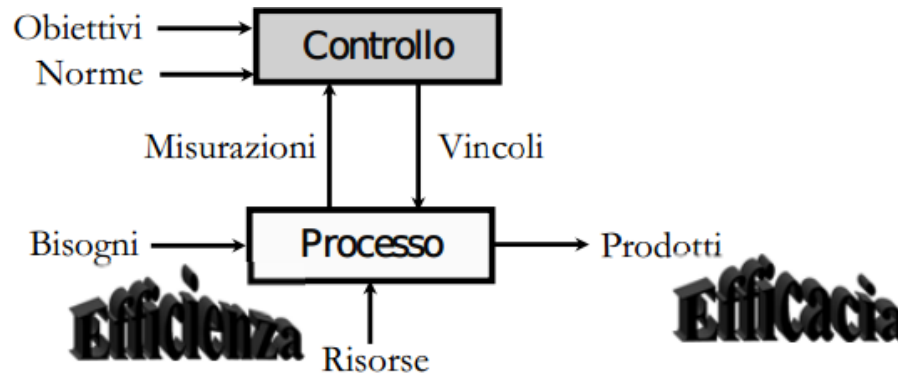
**Iterazione ed incremento**, iterazione significa che faccio una cosa più volte finchè non raggiungo un limite; incremento significa aggiungere, è additivo. Sono incrementale solo se aggiungo, è un obiettivo molto importante da raggiungere (e potenzialmente distruttivo, potrebbe far perdere tempo). Iterare significa riprovare, non ripetere. Non si può togliere quando si è incrementali.

**Prototipo**, serve per imparare, è tipicamente “usa e getta”. Viene diviso in due categorie: quelli rivolti al cliente, per fargli capire cosa avrà, e quelli rivolti verso di noi per aiutarci a trovare una soluzione. Quelli rivolti verso di noi sono un costo, mentre quelli rivolti verso il cliente sono un valore aggiunto. Il primo

impatto tra l'utente e il prodotto è l'**interfaccia**. I prototipi possono essere “usa e getta” ma costano tempo.

**Riuso**, il sw che già esiste è la maggior parte del nostro prodotto. L'informatico deve insegnare a *riusare* in modo sistematico e non opportunistico. Il riuso è cosa saggia se so approvvigionarmi da un fornitore intelligente.

La manutenzione richiede gestione della storia, **versionamento**. Bisogna avere una storia del proprio prodotto. Bisogna spiegare e documentare la scelta ed avere una tecnica che salvi la storia (**repository**).



L'efficienza si vede dove vedo il consumo di risorse. L'efficacia si misura guardando i prodotti e vedendo se sono buoni o cattivi rispetto alla produzione. Un processo è un insieme di attività coordinate e coese (tutti hanno bisogno di tutti).

Standard di riferimento: ISO/IEC 12207.