

Lezione 3

Il rischio è tutto ciò che mina le nostre certezze, ciò che non riusciamo a governare. Bisogna convivere con il rischio e mitigarlo, cercare di prevenire il danno.

Risk mitigation, ma prima bisogna capire quali sono i rischi. Un progetto deve guardare queste cose. Bisogna affrontarli in modo che siano addomesticati.

Un progetto non inizia se non con un'attività importante che faccia emergere i **requisiti**; la maggior parte dei quali sono inespressi. Gli stakeholder non hanno la capacità o la pazienza di scrivere tutto ciò di cui hanno bisogno. Un fornitore di un prodotto deve **intuire** queste necessità. Questo si fa con analisi ed emersioni predefinite. Bisogna vedere i requisiti non scritti. Questa attività è un valore aggiunto dominante.

Verifica e validazione, insieme li chiamiamo qualifica ma sono due cose distinte ed occupano un sacco di tempo. Consapevolezza di consegnare al *customer* un prodotto valido. Non dobbiamo far fare a lui il *debug*. E' irragionevole sviluppare sw da zero (**from scratch**). Non è più l'epoca di far le cose così, bisogna puntare sul riuso non sull'"*ex novo*". Il *coding* è un'attività frazionaria con un notevole rischio, quindi meno tempo utilizzo per questo meglio è.

Un processo è un insieme di attività complesse. Le attività sono suddivisibili in parti più piccole, chiamate **task**. Devo **vivere sulla pianificazione**, suddividere il tempo. Quanto più piccolo è il compito tanto più piccolo è il rischio. Interesse decisivo è spezzare le attività ad una grana fina per ridurre i tempi, costi e rischi. E' importante capire come *frantumare le attività*. Bisogna poi lavorare **in parallelo**. Dare compiti grandi è fonte di rischi e va evitato. Bisogna concentrarsi su obiettivi raggiungibili facilmente.

C'è un' enfasi dominante sulle **prove** (*tests*). La programmazione è congiunta al *testing*. La misura di produttività in un processo sw è *linee codice/ora*. Il numero massimo di linee produttive è 25 al giorno. Questo sul sw ex-novo. Non è scrivere codice che è importante ma scrivere codice che funzioni alla consegna. Oltre che scrivere codice bisogna anche verificarlo prima di immetterlo nella *repository*.

Si può scendere sotto i tasks, dal basso nascono strumenti e tecniche che dobbiamo acquisire prima di metterci sui tasks. Un compito lo si assume avendo acquisito consapevolezza degli strumenti e le tecniche per portarlo a termine. Bisogna scegliere uno strumento che sia **collaborativo**.

Un sw non è fine a se stesso ma è parte di un sistema. Il sistema lo compongono tutti coloro che sono chiamati a usarlo. Bisogna capire bene il sistema. La parte dei requisiti emergerà dal **dominio**, dall'ambiente. Dicesi organizzazione un aggregato di persone che agiscono secondo regole, sono sistematici, disciplinati e quantificabili. Senza organizzazione c'è il **chaos**.

Un'impresa certamente funziona a **pipeline**. L'organizzazione è divisa in *settori*, ciascuno con un proprio compito, che sono finalizzati a un **flusso**. Ogni settore risponde alle esigenze di attività trasversali. Le regole sono i processi. Prendere gli standard rilevanti ed utilizzare un'istanza. Poi fissare ai processi *specifiche di progetto*.

La conoscenza su come il sw funzione deve essere scritta. I processi vanno documentati. **Norme di progetto**, che dicono come un ruolo va svolto.

Influenza la grandezza del progetto, che dice quanto tempo mi ci vorrà. In questo modo potrò fare un conto in base ai tempi e saprò di quante persone avrò bisogno. Dipende anche dalla **complessità** del progetto e dai **rischi**, come la competizione, il rischio finanziario... Devo capire la distanza tra le competenze disponibili e quelle richieste. Devo colmare il gap, studiare in modo organizzato.

I processi, che sono l'organizzazione, hanno bisogno di manutenzione, non possono durare per sempre. Voglio migliorare rispetto all'efficienza e all'efficacia. Ciclo di ritorno dall'esperienza, imparare dai propri errori.

I processi produttivi devono avere un **ciclo interno** atto a migliorarli costantemente. Il ciclo interno di miglioramento è indicato con l'acronimo PCDA (o principio di Deming). Questo ciclo è fatto di 4 attività a ciclo:

- **Plan**, non è il piano delle attività di processo, ma il piano di miglioramento; pianifico ciò che produce efficienza ed efficacia;

- **Do**, stretta attuazione di ciò che ho pianificato. Devo sapere rispetto a cosa migliorare;
- **Check**, guardo l'esito migliorativo delle modifiche fatte rispetto al piano;
- **Act**, ottenuto il risultato porto questo esito a migliorare il processo. Porto miglioramenti o, se fallisco, ragiono su come portare miglioramenti.

Il PCDA serve a migliorare l'efficienza e l'efficacia attraverso degli obiettivi di miglioramento. Prima di fare il plan del PCDA devo fare **analisi**.

Un ciclo di vita sono gli stati di maturità che ha un sw dalla sua nascita al suo ritiro. Bisogna fornire agli stakeholder evidenza di fattibilità.