

Appunti Ingegneria del software

(A.A 2018/2019)

Grigoras Valentin

Matricola: 1099561

Contenuti

- 1 **Indicare le differenze (natura, finalità, collocazione) che intercorrono tra le attività di verifica e quelle di validazione** 3
- 2 **Fissando l'attenzione sulla definizione di "processo" associata allo standard ISO/IEC 12207, indicare come (secondo quali regole), quando (in quali fasi di progetto) e perché (attraverso quali attività) la vostra esperienza di progetto didattico ha visto attuato tale concetto.** 3
- 3 **Fornire una definizione del formalismo noto come "diagramma di Gantt", discuterne concisamente le finalità e modalità d'uso, l'efficacia e i punti deboli eventualmente rilevati nell'esperienza del progetto didattico** 3
- 4 **Dare una definizione ben fondata del concetto di "architettura software". In relazione a tale concetto, dare una definizione ai termini "framework" e "design pattern" spiegando come questi si integrino fra loro e all'interno di una architettura.** 4

- 1 Indicare le differenze (natura, finalita, collocazione) che intercorrono tra le attività di verifica e quelle di validazione**
- 2 Fissando l'attenzione sulla definizione di "processo" associata allo standard ISO/IEC 12207, indicare come (secondo quali regole), quando (in quali fasi di progetto) e perché (attraverso quali attività) la vostra esperienza di progetto didattico ha visto attuato tale concetto.**

Un processo è un insieme di attività **correlate** (ovvero contiene solo attività che hanno a che fare con il progetto) e **coese** (un insieme di cose è coeso se tutto ciò che c'è serve, ci deve essere e se non ci fosse mancherebbe, quindi non c'è nulla di superfluo) con l'obiettivo di rispondere ai bisogni in ingresso restituendo risposte (prodotto delle attività del processo) in uscita agendo secondo regole date consumando risorse nel farlo. Significa che un processo non è mai solo perché sopra di lui c'è un *controllo* che sa come le cose stanno andando perché emette vincoli sul modo in cui il processo lavora misurando l'efficienza e l'efficacia.

L'efficienza si misura con l'efficienza produttiva. Guardo quindi il rapporto tra quantità di prodotto realizzato e risorse utilizzate.

L'efficacia invece si misura in base a quanti obiettivi interni (del fornitore) o esterni (gradimento del cliente) raggiungo.

L'insieme di efficienza ed efficacia si chiama economicità, ovvero raggiungo gli obiettivi se sono efficace consumando poche risorse.

L'ISO/IEC 12207 contiene una descrizione approfondita dei processi del ciclo di vita del software, ed è infatti il modello più noto e riferito, anche se ne esistono altri. Questo modello è ad alto livello, ed identifica i processi dello sviluppo software, ed ha una struttura modulare che permette, nel processo di specializzazione, di identificare le entità responsabili dei processi ed i prodotti dei processi. Secondo questo modello si hanno processi (processes), che sono divisi in attività (activities) che, a loro volta, sono divisi in compiti (tasks). Così si ha una struttura modulare (perché i processi si interfacciano), ma con una forte coesione (perché i compiti sono chiusi). I processi descritti in ISO 12207 hanno lo scopo di eliminare tutti gli sprechi di tempo e risorse, eliminando le particolari attività per un progetto specifico. Nel corso del progetto didattico, si è innanzitutto cercato di determinare i processi necessari allo sviluppo del prodotto attenendosi fedelmente allo standard ISO/IEC 12207. Si è poi passati alla definizione generica delle attività costituenti ogni processo per poi contestualizzarla e dettagliarla sempre più con il progredire del progetto. Si è dunque assicurati che all'istanziatura di ogni processo questo fosse normato e ben definito rispetto alle attività che lo compongono e alle sue pre e post condizioni. Da un punto di vista pratico, questo approccio ha permesso di lavorare secondo una migliore suddivisione in task, migliorando progressivamente la qualità dei processi una volta concluse le relative attività e verificati i relativi prodotti.

- 3 Fornire una definizione del formalismo noto come "diagramma di Gantt", discuterne concisamente le finalità e modalità d'uso, l'efficacia e i punti deboli eventualmente rilevati nell'esperienza del progetto didattico**

Il diagramma di Gantt, usato principalmente nelle attività di project management, è uno strumento che serve a pianificare un insieme di attività in un certo periodo di tempo. È costituito da 2 assi. Sull'asse orizzontale si indica il tempo totale del progetto, suddiviso in fasi incrementali (ad esempio, giorni, settimane, mesi), mentre sul asse verticale ci sono le attività da svolgere avente un tempo d'inizio e un tempo di fine, ma non la quantità di lavoro in termini di ore.

Le barre orizzontali di lunghezza variabile rappresentano le sequenze, la durata e l'arco temporale di ogni singola attività del progetto. Le attività da svolgere possono essere sovrapposte durante il medesimo arco temporale ad indicare la possibilità dello svolgimento in parallelo di alcune delle attività, oppure dipendenti se un'attività finisce prima che inizi la successiva attività.

Una linea verticale è utilizzata per indicare la data di riferimento.

Il diagramma di gantt permette quindi di visualizzare chiaramente il flusso di lavoro mostrando la data di inizio e di fine di una determinata attività, consentendo un uso intelligente ed efficace delle risorse.

4 Dare una definizione ben fondata del concetto di "architettura software". In relazione a tale concetto, dare una definizione ai termini "framework" e "design pattern" spiegando come questi si integrino fra loro E all'interno di una architettura.

Un'architettura software è un insieme di elementi architettureali, utilizzati secondo una particolare forma(intesa come organizzazione e strutturazione) insieme a una giustificazione logica che coglie la motivazione per la scelta degli elementi e della forma. Per **forma** si intende la divisione di tale sistema in componenti, nella disposizione di essi e nei modi in cui tali componenti comunicano tra loro. **La giustificazione logica** ha lo scopo di rendere esplicite le motivazioni per la scelta degli elementi e della forma – in particolare, con riferimento al modo in cui questa scelta consente di soddisfare i requisiti/interessi del sistema. **Lo scopo dell'architettura software** è di facilitare lo sviluppo, la distribuzione, il funzionamento e la manutenzione del sistema software in esso contenuto, quindi di supportare il ciclo di vita del sistema.

Un pattern software è una soluzione provata e ampiamente applicabile a un particolare problema di progettazione che è descritta in una forma standard, in modo che possa essere facilmente condivisa e riusata.

Un **design pattern** è la descrizione di oggetti e classi che comunicano tra di loro, personalizzati per risolvere un problema generale di progettazione in un contesto particolare. I design pattern sono spesso utili nel descrivere le connessioni tra elementi architettureali indicando un approccio uniforme nella loro realizzazione. **I benefici dei design pattern** soprattutto dal punto di vista dell'architettura del software sono la riduzione del rischio, sulla base di soluzioni provate e ben comprese e un maggior incremento della produttività, della standardizzazione e della qualità.

Con la parola **framework** intendiamo una micro architettura che mette a disposizione tipi estendibili nell'ambito di uno specifico dominio. Nell'architettura software un framework è una parte di software riutilizzabile ed estendibile.