

NOME _____ COGNOME _____

Indicare gli scopi principali di ogni fase del processo dell'Ingegneria:

Il processo dell'ingegneria è diviso in 5 fasi.

Lo scopo della fase di specifica è individuare ed analizzare requisiti (funzionali e non) e stabilire se il progetto è fattibile e/o utile. A questo scopo, viene redatto un documento dei requisiti.

Lo scopo della fase di progettazione è definire come il prodotto verrà implementato: viene scelta l'architettura, le strategie di programmazione, la struttura del programma, ecc...

Lo scopo della fase di implementazione è implementare, a livello di codice, ciò che è stato proposto nelle fasi di specifica e progettazione.

Lo scopo della fase di collaudo è testare il prodotto implementato per eliminarne eventuali problemi prima di rilasciarlo al pubblico o al committente.

Lo scopo della fase di manutenzione è supportare il prodotto dopo il rilascio apportandone modifiche che correggano eventuali errori emersi oppure ne migliorino la stabilità e/o utilizzo.

Supponendo di dover sviluppare un sistema software per la vendita on line di prodotti, indicare quattro esempi di requisito funzionale e quattro esempi di requisito non funzionale:

RF: creazione nuovo utente, carrello dei prodotti da acquistare, visualizzazione di notifiche per un nuovo prodotto, ricerca prodotti

NF: interfaccia realizzata tramite Java Swing, database implementato con SQLite, password lunghe almeno 8 caratteri, utilizzo di Git per version control

Indicare cosa è prodotto in termini di documentazione, da ogni fase del processo di Specifica:

Durante la fase di studio di fattibilità viene prodotto il rapporto di fattibilità, che contiene la valutazione della possibilità di costruire il sistema ed eventuali modifiche agli obiettivi.

Durante la fase di deduzione dei requisiti viene redatto un documento contenente la definizione ad alto livello (in linguaggio naturale) dei requisiti.

Durante la fase di analisi dei requisiti viene redatto un documento contenente i requisiti (funzionali e non) riscritti in linguaggio tecnico e preciso; viene inoltre creato il diagramma degli stati.

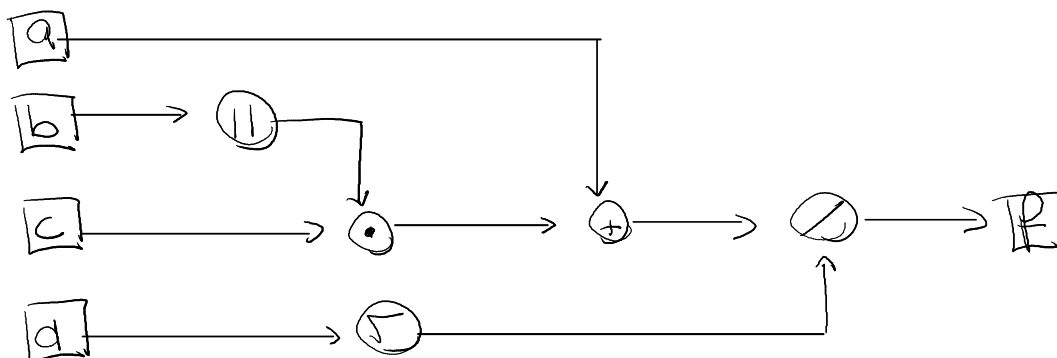
Durante la fase di validazione dei requisiti viene validato il documento dei requisiti.

Completare l'affermazione per ogni tipo di controllo:

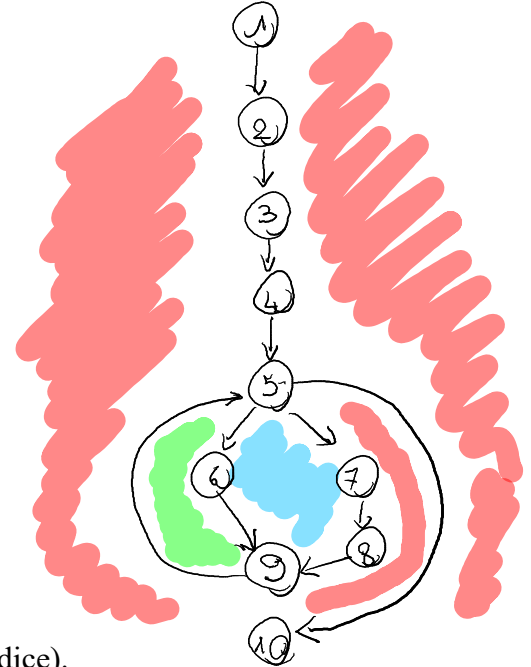
	controllo centralizzato	controllo basato sugli eventi
Un componente si attiva quando...	il controllore, che periodicamente controlla se <u>gli altri componenti hanno prodotto dati da elaborare</u> , <u>attiva un determinato componente</u>	<u>quando si verifica un evento esterno (segnale, comando, input ...)</u>

Rappresentare la seguente formula con un modello data-flow in cui a , b , c , d sono i dati di input iniziali, i nodi filter rappresentano gli operatori matematici, e f è l'output finale:

$$(a + |b| \cdot c) / \sqrt{d} = f$$



```
// Segue il codice della funzione calcola_potenza:
float calcola_potenza(float base, int esp) {
    int i=0;
    float potenza=1;
    while ( i < abs(esp) ) {
        if (esp > 0)
            potenza = potenza * base;
        else
            potenza = potenza / base;
        i++;
    }
    return potenza;
}
```



- Disegnare il flow-graph corrispondente al programma (a lato del codice).
- Indicare un metodo per calcolare la complessità ciclomatica ed applicarlo al flow-graph ottenuto.
- Cosa indica il valore ottenuto?
- Individuare i cammini indipendenti all'interno del flow-graph.
- Per ogni cammino ottenuto definire un test-case che determini tale cammino.

La complessità ciclomatica può essere calcolata in tre modi:

- $CC = \#archi - \#nodi + 2 = 3$
- $CC = \#regioni = 3$
- $CC = \#nodi_predicato + 1 = 3$

Cammini indipendenti:

- 1, 2, 3, 4, 5, 10
- 1, 2, 3, 4, 5, 6, 9, 5, ...
- 1, 2, 3, 4, 5, 7, 8, 9, 5 ...

Cammino 1: Input: (10,0) Output: 1

Cammino 2: Input: (2,2) Output: 4

Cammino 3: Input: (4,-1) Output: 0.25

Supponendo che l'esecuzione di un test-case abbia riscontrato la presenza di un difetto, cosa si intende per

Debugging: _____

si controlla l'esecuzione del programma, istruzione per istruzione, si controlla il valore delle variabili e si scopre la posizione dell'errore

Correzione: si modifica il codice per eliminare il difetto

Test di regressione: _____

si ripete l'ultimo test-case, per verificare che le modifiche al codice abbiano corretto l'errore, e si ripetono i test-case precedenti, per verificare che le modifiche non abbiano prodotto effetti collaterali

Considerando di nuovo il sistema software per la vendita on line di prodotti, fornire un esempio concreto di manutenzione correttiva, adattiva, migliorativa:

Correttiva: correzione di difetti non emersi i fase di collaudo

Adattiva: adattamento del sistema a cambiamenti di piattaforma

Migliorativa: aggiunta, cambiamento o miglioramento di requisiti funzionali e non, secondo le richieste del committente o tendenze di mercato

Correttiva: correzione di un bug che preveniva agli utenti di ordinare prodotti in esaurimento ma comunque disponibili

Adattiva: aggiornamento del DBMS ad una versione più recente

Migliorativa: modifica delle interfacce utente per renderle più chiare ed accessibili

Indicare tre fattori che influenzano il costo di manutenzione e spiegarne il motivo:

Struttura del codice: codice ben strutturato e documentato rende più facile la manutenzione

Collaudo: una fase di collaudo approfondita riduce il numero di difetti scoperti in seguito alla consegna

Stabilità dello staff: i costi si riducono se lo staff che ha sviluppato il prodotto è lo stesso che lo mantiene

In caso di re-engineering di sistemi ereditati, fare un esempio di traduzione del codice _____

traduzione del codice da COBOL a Java

ristrutturazione del codice _____

eliminazione di parti di codice ridondanti

ristrutturazione dei dati _____

passaggio da da file in formato proprietario a DBMS

Fare un esempio di reverse engineering

dato il codice del sistema _____

generazione di documentazione

dato un archivio di dati _____

generazione di codice sorgente

Indicare gli scopi delle attività di pianificazione del progetto.

Scomposizione: _____

il processo viene diviso in task, vengono identificate le dipendenze tra di essi;
vengono definite milestone e deliverable

Tempistica (scheduling): _____

vengono determinate le durate dei task e la durata complessiva del progetto

Assegnazione delle risorse: _____

vengono assegnate risorse (personale, hw/sw e budget) ad ogni task

Data la classificazione dei rischi in base alla causa, fornire un esempio di

Rischio tecnologico _____

un database non riesce a processare tutte le richieste che gli vengono inviate

Rischio del personale _____

personale chiave del progetto non è disponibile

Rischio organizzativo _____

ristrutturazione causa un cambio dello staff manageriale

Rischio strumentale _____

il codice generato dai tool CASE non è efficiente/adeguato

Rischio dei requisiti _____

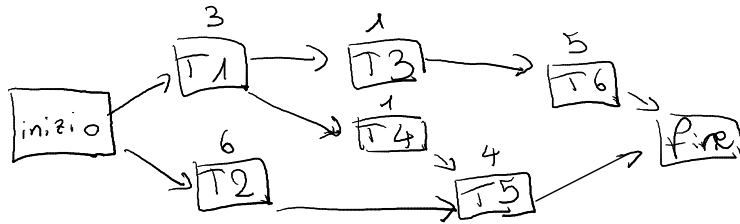
i requisiti non sono corretti e vanno modificati

Rischio di stima _____

il budget non è sufficiente per completare il progetto rispettando i requisiti

All'interno di un processo software sono stati individuati i task T1, T2, T3, T4, T5, T6. Tali task hanno le seguenti durate e dipendenze:

Task	Durata	Dipendenze
T1	3gg	-
T2	6gg	-
T3	1gg	T1
T4	1gg	T1
T5	4gg	T2, T4
T6	5gg	T3



In base alle dipendenze tra task, disegnare l'activity network (a lato della tabella).

In base all'activity network, indicare tutti i task che da ultimare per consentire l'inizio di T5: 1, 2, 4

In base alla durata dei task, individuare il cammino critico nell'activity network: 2, 5

In base al cammino critico, determinare la durata del progetto: 10 giorni

Qual è il ritardo massimo consentito a T1 in modo da non alterare la durata del progetto, assumendo che gli altri task rispettino i tempi previsti? 2 giorni

Spiegare il motivo:

dato che il task T2 impegna 6 giorni per essere svolto, il tempo massimo per lo svolgimento dei task T1 e T4 non deve superare i 6 giorni; dato che T4 rispetta le tempistiche, T1 può impiegare al massimo 5 giorni

Per ogni caratteristica, indicare i modelli di processo che li riguarda:

caratteristica	Modello a cascata	Sviluppo basato sul riuso
Le fasi del processo non sono cicliche	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
I requisiti possono essere adattati ai componenti già esistenti	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Il sistema è consegnato solo alla fine del processo	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Si assume che tutti i componenti siano sviluppati durante il processo	<input checked="" type="checkbox"/>	<input type="checkbox"/>