Ingegneria del Software
A. A. 2016-2017

Esame di teoria
20 gennaio 2017

NOME_____ COGNOME_____

Fornire una definizione di Ingegneria del software: _____

Con Ingegneria del software s'intende l'applicazione del processo dell'Ingegneria alla produzione di sistemi software.
L'Ingegneria del software dovrebbe limitare la presenza di difetti alla consegna del sistema software e fare in modo che il sistema soddisfi il committente.

Indicare gli scopi principali di ogni fase del processo software:

Il processo software è diviso in 5 fasi: specifica, progettazione, implementazione, collaudo e manutenzione.

Durante la fase di specifica viene compiuto lo studio di fattibilità, vengono dedotti ed analizzati i requisiti funzionali e non funzionali ed infine vengono validati. È un processo ciclico.

Durante la fase di progettazione determinata la struttura del sistema, il suo deployment ed il suo funzionamento.

Durante la fase di implementazione viene implementato il sistema, ossia vengono programmate le varie componenti specificate nel documento dei requisiti.

Durante la fase di collaudo vengono eseguiti l'ispezione ed il testing del sistema per assicurare il corretto funzionamento di tutte le componenti ed identificare e correggere eventuali errori.

Durante la fase di manutenzione il sistema, ormai rilasciato, viene modificato per adattarlo alle esigenze dei committenti o per correggere errori non individuati durante la fase di testing.

Supponendo di dover sviluppare un sistema software per la vendita on-line di prodotti, fornire tre esempi di requisito funzionale, un esempio di requisito non funzionale di prodotto, un esempio di requisito non funzionale organizzativo, un esempio di requisito non funzionale esterno:

Funzionali: lista prodotti, login utente, effettua ordine

NF prodotto: interfaccia utente intuitiva

NF organizzativo: utilizzo di un databse noSQL

NF esterno: rispetto della privacy degli utenti

Indicare gli scopi principali di ogni fase del processo di Specifica:

Il processo di specifica ha 4 fasi: studio di fattibilità, deduzione dei requisiti, analisi dei requisiti e validazione dei requisiti.

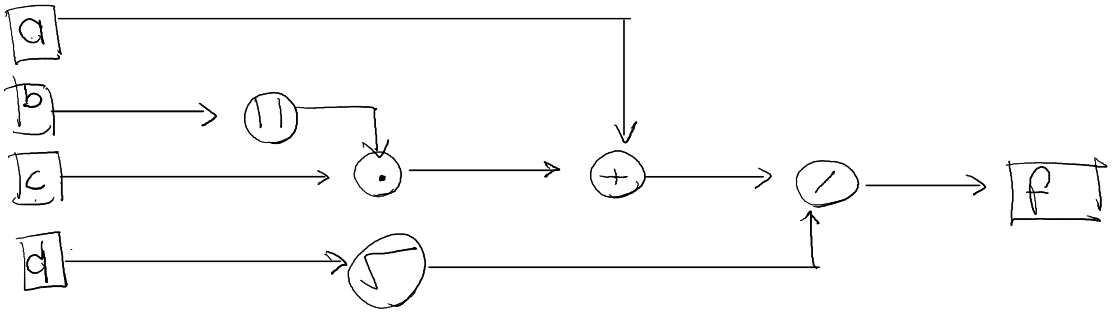
Durante lo studio di fattibilità viene valutata la possibilità di sviluppare il sistema e dei suoi vantaggi per il committente,a questo scopo viene preparato un rapporto di fattibilità.

Durante la fase di deduzione dei requisiti si compie un'indagine per dedurre i requisiti. Le informazioni vengono raccolte dallo studio del dominio applicativo e dialogo con stakeholder.

Durante la fase di analisi dei requisiti questi vengono classificati ed organizzati, stabilendo priorità e modellandoli con l'utilizzo di strumenti CASE.

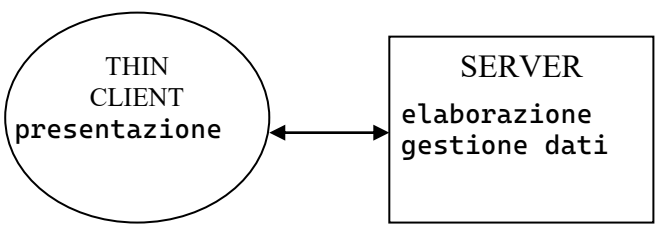
Durante la fase di validazione dei requisiti si verifica che il documento dei requisiti non contenga errori di specifica che potrebbero impattare fasi successive del prodotto software.

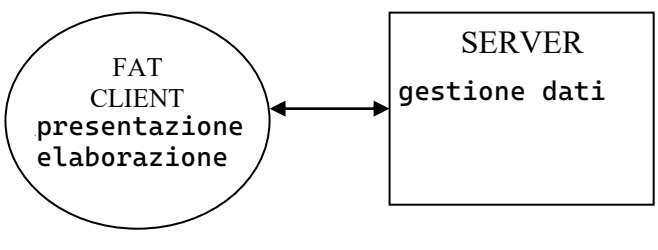
Rappresentare la seguente formula in un modello data-flow in cui a*, b, c* sono i dati di input iniziali, i nodi filter rappresentano gli operatori matematici, e *f* è l'ouput finale: $(a+|b|\cdot c)/\sqrt{d}=f$



Si consideri l'architettura Client-Server stratificata su 3 livelli e distribuita su 2 macchine.

a) indicare nel caso di "thin client" e "fat client" quali livelli risiedono sulla macchina client e quali sulla macchina server:





b) Descrivere lo scopo di ogni livello:

presentazione: raccolta dati dall'utente, visualizzazione output all'utente

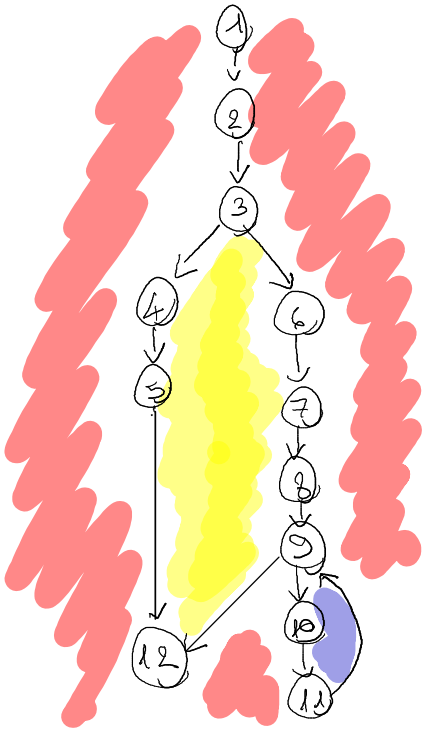
elaborazione: gestione input, produzione dati in output

gestione dati: aggiunta, modifica, rimozione e ricerca dati

Indicare per ogni aspetto se può essere collaudato tramite ispezione e/o testing. Motivare la risposta:

aspetto	ispezione	testing
posizione dei bachi nel codice la posizione può essere determinata tramite testing mirato oppure accurata ispezione del codice, anche sfruttando software apposito	X	X
tempo di risposta del sistema il tempo di risposta non può essere dedotto solamente dall'ispezione, in quanto dipende da molti fattori non solo legati al codice	○	X
strutturazione del codice la struttura del codice è evidente tramite ispezione del codice stesso	X	○

```
// funzione che calcola il fattoriale (n!) di un numero
int fattoriale(int n) {
    int i, fatt;
    if (n<0) {
        printf("ERRORE\n");
        fatt = -1; }
    else {
        fatt=1;
        i=n;
        while (i>1) {
            fatt=fatt*i;
            i--; }
    }
    return fatt;
}
```



a) Costruire il flow-graph corrispondente al programma.
b) Indicare un metodo per calcolare la complessità ciclomatica ed applicarlo al flow-graph ottenuto.
c) Individuare i cammini indipendenti all'interno del flow-graph.
d) Per ogni cammino ottenuto definire un test-case che determini tale cammino.

La complessità ciclomatica può esere calcolata in tre modi:
1. CC = #archi + #nodi + 2 = 3
2. CC = #regioni = 3
3. CC = #nodi_predicato + 1 = 3

Cammini indipendenti:
1. 1,2,3,4,5,12
2. 1,2,3,6,7,8,9,12
3. 1,2,3,6,7,8,9,10,11,9, ...

Cammino 1: Input: -1 Output: -1
Cammino 2: Input: 0 Output: 1
Cammino 3: Input: 5 Output: 120

Fornire la definizione e un esempio di manutenzione correttiva, adattiva, migliorativa, con riferimento al sistema software per la vendita on-line di prodotti considerato in precedenza:

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

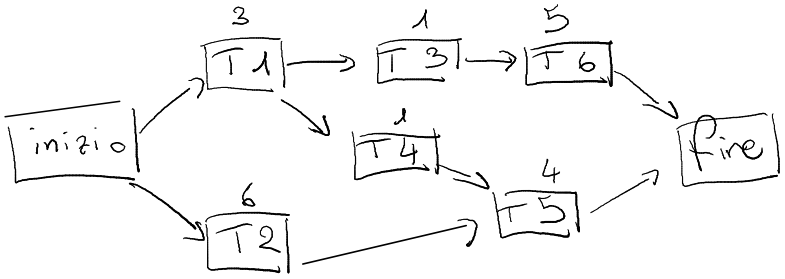
Descrivere il possibile effetto di ogni rischio. Quindi classificare il rischio in base a tale effetto:

Rischio	Rischio di progetto	Rischio di prodotto	Rischio di business
Eccessivo ritardo nel reperimento di hardware o software può causare un ritardo nella consegna del prodotto	X	○	○
Cambiamento radicale di un gruppo di requisiti può causare il ritardo nella consegna del prodotto e/o impattarne negativamente la qualità	X	X	○
Uno sviluppatore lascia l'azienda se lo sviluppatore era una figura chiave nel progetto, si possono verificare ritardi nella consegna del prodotto	X	○	○
Un prodotto concorrente viene messo sul mercato la competizione è detrimentale a livello economico	○	○	X

All'interno di un processo software sono stati individuati i task T1, T2, T3, T4, T5, T6. Essi 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, costruire l'activity network.



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? 5 giorni
Spiegare il motivo:

dato che il task T2 impega 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

Dato il modello di processo detto eXtreme Programming, indicare due modi per ottenere la rapidità di consegna e due modi per ottenere la qualità del prodotto, spiegandone gli effetti.

Rapidità di consegna: specifica e progettazione ad alto livello, i dettagli vengono definiti durante lo sviluppo

Qualità del prodotto: la programmazione avviene a coppie (aumenta la probabilità di notare un errore, più sviluppatori conoscono il codice) e test-driven development (si scrivono prima i test e poi il codice, finchè il test non ha successo)