

Dire quale delle seguenti tecniche NON è di black box testing: non ricordo le opzioni ma era ovvia (complessità ciclomatica)

A cosa serve la complessità ciclomatica? (Anche questa a crocette e molto ovvia)

Domanda aperta: Spiegare cos'è e che ruolo ha il validation testing.

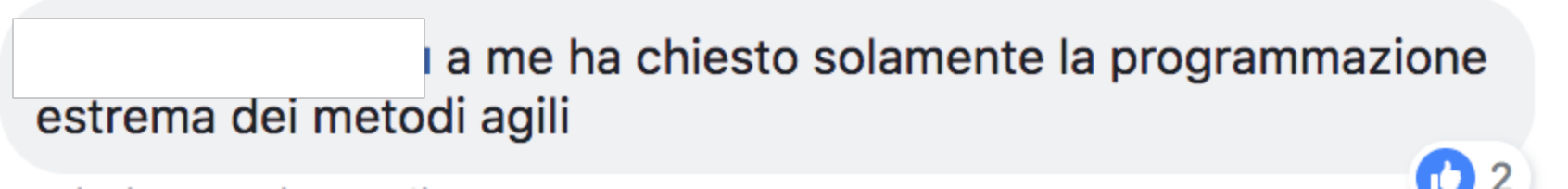
Esercizio: data la seguente funzione (codice per calcolare il voto di sas) utilizzare le classi di equivalenza per scrivere dei test per verificarne la correttezza.



A me Torta ha chiesto: panoramica generale sul

riuso del software, approcci che supportano il riuso del software,
framework applicativi nel dettaglio e linee di prodotti nel
dettaglio





Il cliente

mi ha chiesto solamente la programmazione
estrema dei metodi agili



domande anche per capire se hai capito. Tipo a me ha chiesto che cosa è la progettazione del software , come si sostiene , come si utilizza , o per esempio mi ha chiesto dove è possibile usare i prototipi , in che ambito , di fare esempi insomma.



Torta: la prototipazione, facendomi partire dal modello a spirale

Picardi: cos'è il White box test, e alcuni nomi/modi di test White box

Bono: uml, pattern e up

Torta/Picardi: testing



1

BONO:

Quando si fanno gli SDD (crocette)

Cos'è il modello di dominio (crocette)

Quali di questi non è un pattern GoF (crocette)

domanda sul riuso del codice (crocette)

Differenza tra modelli plan-driven e agile (aperta)

Disegnare il pattern Decorator e scrivere il codice fondamentale (aperta)

PICARDI:

Quale di queste non è black-box testing (crocette)

Difetto dell'approccio top-down (crocette)

Quale di queste è white-box testing (crocette)

Definire la complessità ciclomatica (cosa rappresenta e cosa serve nel testing) (aperta)

x = risposta corretta secondo Marco

1) Quando si parla di regression testing? (3pt)

a) quando si effettuano test frequenti e non esaustivi e relativi alle funzionalità principali per decidere se il prodotto è sufficientemente stabile per essere testato da un test specializzato

b) quando si scrivono i test prima di aver implementato il codice

x c) quando, a seguito di una correzione di un bug o dell'integrazione di un nuovo componente, si vanno a ripetere i test già eseguiti

per verificare che non siano state introdotti nuovi bug

d) quando si testano i componenti o le unità in ordine inverso rispetto a quando sono realizzati

2) Nel testing cosa si intende per stub (3pt)

a) una tecnica di black box testing nella quale si utilizza una versione fittizia dell'unità da testare per farle calcolare i risultati attesi e confrontarli con quelli effettivamente ottenuti

b) un approccio all'integration testing

c) il codice che invoca i metodi di un dato modulo dell'unità per testarlo

x d) una versione fittizia di un modulo/unità con la stessa interfaccia ma con comportamento predefinito, che consente il test in isolamento di altri moduli adesso dipendenti

3) quale de seguenti NON è una delle tante forme di riuso dell'ingegneria del software (3pt)

a) framework

b) design pattern

x c) pattern matching

d) librerie

4) quali dei seguenti ruoli non è previsto dal metodo SCRUM di gestione dei progetti Agili? (3pt)

- a) Scrum Master
- b) Project manager
- c) Developer
- x d) Tester

5) Tecniche di black box testing

6) In cosa consiste la complessità ciclomatica?

7) fasi V-shaped

NO 8) La differenza tra riuso COTS e sistemi applicativi integrati (Non dovremmo aver fatto questa roba quest'anno)

Pre-Test per la teoria di Sviluppo Applicazioni SW
Appello del 25/6/2017 - Data pre-test: 25/6/2017

NOME:

MATRICOLA:

Il punteggio del pre-test arriva fino a 32. Nella media finale un voto superiore a 30 viene conteggiato come un 30 e lode.

Nota per gli studenti degli A.A. precedenti: chi aveva in precedenza sostenuto la parte di teoria con la prof. Bono deve rispondere solo alle domande delle sezioni B1 e B2. Chi aveva in precedenza sostenuto la parte di teoria con i proff. Picardi e Torta deve rispondere solo alle domande della sezione A.

Sezione A: UML, UP e Design pattern.

Domande a risposta chiusa

A-1: Quando si fa il System Sequence Diagram (SSD)? [2 punti]

- ☐ Quando si comincia a scrivere il codice.
- ☐ All'inizio dell'analisi dei requisiti.
- ☒ Dopo i casi d'uso dettagliati.
- ☐ Non esiste questo diagramma in UP.

A-2: Cos'è il modello di dominio? [2 punti]

- ☐ È una rappresentazione implementativa del sistema.
- ☐ È un altro nome per indicare il modello di progetto.
- ☒ È una rappresentazione concettuale del sistema.
- ☐ Serve a rappresentare i casi d'uso in modo grafico.

A-3: Quante iterazioni ha la fase di ideazione? [2 punti]

- ☐ Quante ne servono.
- ☐ Normalmente una.
- ☐ Dipende dalla percentuale dei casi d'uso che si intende implementare.
- ☒ Non ha iterazioni.

A-4: Dire quale NON è un pattern GoF. [2 punti]

☐ Abstract Factory

☒ Proxy

☐ Factory

☐ Composite

Domande a risposta aperta

A-A: Disegnare il pattern **Decorator** e scrivere il codice delle sue classi caratterizzanti, inclusi i costruttori. (NOTA: basta scrivere del codice schematico, che contenga però gli elementi essenziali.) [6 punti]

Sezione B2: Testing**Domande a risposta chiusa**

B2-1: Qual è il principale svantaggio della **strategia top-down** per l'integration testing, che la strategia bottom-up invece non presenta? [2 punti]

- ☒ Poiché combina subito insieme tutte le unità, rende difficile capire la causa dei problemi rilevati.
- ☐ Non può essere applicata in caso di progettazione orientata agli oggetti.
- ☒ Richiede di scrivere molti stub
- ☐ Testa per ultime le componenti più complesse e quindi più soggette ad errori, ritardandone la scoperta.

B2-2: Quando si parla di **regression testing** o **test di regressione**? [2 punti]

- ☐ Quando si effettuano dei test frequenti e non esaustivi relativi alle funzionalità principali, per decidere se il prodotto è sufficientemente stabile per essere testato da un team specializzato.
- ☐ Quando si testano i componenti o le unità in ordine inverso rispetto a quando sono stati realizzati.
- ☐ Quando si scrivono i test prima di avere implementato il codice.
- ☒ Quando, a seguito della correzione di un bug o dell'integrazione di un nuovo componente, si vanno a ripetere i test già effettuati per verificare che le modifiche apportate non abbiano introdotto dei problemi nuovi o non rilevati.

B2-3: Quale delle seguenti **non** è una tecnica di **black-box testing**? [2 punti]

- ☐ Testing casuale
- ☐ Calcolo dei cammini indipendenti
- ☐ Boundary value analysis o analisi dei valori limite
- ☐ Array ortogonali

B2-4: Quale delle seguenti affermazioni sul validation testing è falsa? [2 punti]

- ☐ Il validation testing verifica che il software realizzato corrisponda alle specifiche iniziali/requisiti
- ☐ Alcune fasi del validation testing vengono eseguite direttamente dall'utente finale
- ☒ Documenti quali i Casi d'Uso dettagliati o i Contratti delle Operazioni costituiscono un input rilevante per il validation testing
- ☒ Il validation testing può essere effettuato solo quando il software è stato integralmente sviluppato

Domanda a risposta aperta

B2-A: Esporre sinteticamente come si calcola la **complessità ciclomatica** di una porzione di codice, cosa esprime, e in che modo questo è rilevante per il testing. [4 punti]
