

PARTE A: DOMANDE A RISPOSTA MULTIPLA

Issue Tracking System

- 1. Quale delle seguenti NON è una funzionalità principale di un Issue Tracking System?** a) Condividere informazioni con il team di sviluppo b) Gestire la priorità delle attività c) Compilare automaticamente il codice sorgente d) Avere un'istantanea delle attività da fare, in corso e completate
- 2. Un Work Item in un ITS può essere caratterizzato da tutti i seguenti campi ECCETTO:** a) Assegnatario e Segnalante b) Priorità e Stato c) Codice SHA-1 del commit d) Tipo e Descrizione
- 3. Il Workflow di un Work Item rappresenta:** a) La sequenza di operazioni per compilare il codice b) L'insieme di stati e transizioni che attraversa durante il ciclo di vita c) Il tempo impiegato per completare l'attività d) La lista dei file modificati

Version Control System

- 4. Quale delle seguenti è una caratteristica dei DVCS rispetto ai CVCS?** a) Permettono il lock dei file b) Sono più semplici da apprendere c) Hanno un'architettura distribuita per replica d) Utilizzano solo il Centralized Workflow
- 5. In Git, un COMMIT rappresenta:** a) Un insieme di diff validate e salvate nel repository b) Solo le modifiche all'ultimo file editato c) Un backup completo del progetto d) La sincronizzazione con il repository remoto
- 6. Il comando `git pull` esegue:** a) Solo git fetch b) Solo git merge c) git fetch seguito da git merge d) git push seguito da git fetch

Framework Scrum

- 7. Quale NON è un ruolo nel Framework Scrum?** a) Product Owner b) Scrum Master c) Development Team d) Project Manager
- 8. La durata tipica di uno Sprint in Scrum è:** a) 1 settimana b) 2-4 settimane c) 1-2 mesi d) 6 mesi
- 9. Il Daily Scrum serve principalmente per:** a) Risolvere problemi tecnici complessi b) Sincronizzarsi e pianificare la giornata per raggiungere lo Sprint Goal c) Fare reporting al Product Owner d) Rivedere il codice sviluppato

Build Automation e Maven

10. L'acronimo CRISP descrive le caratteristiche di un processo di build. Cosa significa? a) Completo, Ripetibile, Informativo, Schedulabile, Portabile b) Continuo, Rapido, Integrabile, Sicuro, Performante c) Centralizzato, Robusto, Immediato, Stabile, Preciso d) Configurabile, Riutilizzabile, Incrementale, Scalabile, Personalizzabile

11. In Maven, il POM (Project Object Model): a) È un file binario generato automaticamente b) Contiene solo le dipendenze del progetto c) È un file XML che descrive il progetto e le sue configurazioni d) Viene utilizzato solo per progetti Java Enterprise

12. Il comando `mvn install` esegue tutte le fasi ECCETTO: a) compile b) test c) package d) deploy

Testing

13. I test di unità appartengono alle seguenti categorie: a) Dinamico, Funzionale, White box, Verifica b) Statico, Non funzionale, Black box, Validazione c) Dinamico, Non funzionale, White box, Validazione d) Statico, Funzionale, Black box, Verifica

14. Quale delle seguenti NON è una caratteristica desiderabile dei test di unità (A TRIP)? a) Automatic b) Thorough c) Repeatable d) Collaborative

15. L'analisi statica del codice è caratterizzata da: a) Esecuzione del programma con dati di test b) Analisi del codice senza esecuzione c) Test dell'interfaccia utente d) Verifica delle performance in produzione

Continuous Integration e Continuous Delivery

16. La Continuous Integration consiste principalmente in: a) Rilascio automatico in produzione ad ogni commit b) Allineamento frequente degli ambienti di lavoro verso il mainline c) Esecuzione di test solo una volta al giorno d) Integrazione manuale settimanale del codice

17. Quale delle seguenti affermazioni sulla Continuous Delivery NON è vera? a) Il rilascio può essere fatto premendo un bottone b) Ogni cambiamento può potenzialmente essere rilasciato c) Il processo di build deve essere eseguito più volte per ogni ambiente d) È il passo successivo alla Continuous Integration

18. In una Deployment Pipeline, un "gate" rappresenta: a) Un punto di verifica tra due stage b) Un ambiente di produzione c) Un tipo di test automatico d) Un artefatto software

Configuration Management e Container

19. Il principio di idempotenza nel Configuration Management significa: a) Operazioni diverse danno sempre lo stesso risultato b) Applicando la stessa operazione più volte il risultato è sempre lo stesso c) Ogni operazione deve essere unica d) Le operazioni devono essere eseguite in sequenza

20. Quale delle seguenti è una caratteristica principale dei container rispetto alle macchine virtuali? a) Richiedono un sistema operativo guest completo b) Consumano più risorse c) Condividono il kernel dell'host d) Sono più lenti nell'avvio

PARTE B: DOMANDE APERTE TEORICHE

1. Issue Tracking System

Definire cos'è un Issue Tracking System e descrivere almeno 4 benefici del suo utilizzo in un progetto software. Spiegare inoltre la relazione tra Tipi di Issue, Workflow e Collegamenti.

2. Version Control System

Confrontare i Version Control System Centralizzati (CVCS) e Distribuiti (DVCS), evidenziando vantaggi e svantaggi di ciascuna tipologia. Descrivere almeno 3 workflow pattern utilizzabili con i DVCS.

3. Framework Scrum

Descrivere i tre pilastri su cui si basa il Framework Scrum e spiegare il ruolo di ciascuno dei tre ruoli principali. Illustrare brevemente gli eventi principali di uno Sprint.

4. Build Automation

Spiegare cosa si intende per Build Automation e descrivere le caratteristiche CRISP che dovrebbe avere un processo di build. Fornire esempi concreti per ciascuna caratteristica.

5. Testing Software

Illustrare i 7 principi fondamentali del testing software, fornendo per ciascuno una spiegazione dettagliata e un esempio pratico.

6. Test Driven Development

Descrivere il ciclo di sviluppo TDD (Test Driven Development), spiegando le tre fasi principali e i benefici di questo approccio.

7. Continuous Integration

Definire la Continuous Integration ed elencare almeno 5 best practices per implementarla correttamente. Spiegare perché è importante che il processo di build

sia veloce.

8. Continuous Delivery

Spiegare la differenza tra Continuous Delivery e Continuous Deployment. Descrivere il principio "Deploy the Same Way to Every Environment" e motivarne l'importanza.

9. Configuration Management

Definire il Configuration Management e descrivere i due obiettivi principali: riproducibilità e tracciabilità. Fornire esempi di strumenti moderni di CM.

10. Container e Orchestrazione

Spiegare cosa sono i container e come differiscono dalla virtualizzazione tradizionale. Descrivere brevemente perché è necessaria l'orchestrazione dei container.

PARTE C: DOMANDE PRATICHE

1. Git - Comandi Base

Descrivere la sequenza di comandi Git necessari per:

- Clonare un repository remoto
- Creare un nuovo branch per una feature
- Effettuare modifiche, commit e push
- Gestire un conflitto durante il merge

2. GitHub Issue Integration

Spiegare come è possibile chiudere automaticamente le issue di GitHub tramite commit. Fornire esempi concreti di messaggi di commit che attivano questa funzionalità.

3. Maven - Configurazione Progetto

Descrivere la struttura di un file pom.xml base, spiegando:

- Coordinate GAV (GroupId, ArtifactId, Version)
- Sezione dependencies
- Sezione build con plugin principali
- Come aggiungere una nuova dipendenza

4. JUnit - Test di Unità

Scrivere un esempio di classe di test JUnit che include:

- Metodi con annotazioni `@Before`, `@Test`, `@After`
- Utilizzo di assertion (`assertEquals`, `assertTrue`, etc.)
- Gestione di eccezioni attese
- Test parametrizzati

5. GitHub Actions - CI Pipeline

Descrivere come configurare una GitHub Action per:

- Trigger automatico su push/pull request
- Compilazione con Maven
- Esecuzione test
- Pubblicazione artefatti

6. Jenkins Pipeline

Illustrare la struttura di base di una Jenkins Pipeline dichiarativa che includa:

- Stage di build
- Stage di test
- Stage di deploy
- Gestione degli artefatti

7. Docker - Containerizzazione

Spiegare come scrivere un Dockerfile per containerizzare un'applicazione Java Maven, includendo:

- Immagine base appropriata
- Copia del codice sorgente
- Esecuzione della build
- Esposizione delle porte
- Comando di avvio

8. SonarQube - Analisi Statica

Descrivere come integrare SonarQube in una pipeline CI/CD per:

- Analisi automatica del codice
- Configurazione di Quality Gates
- Gestione delle metriche di qualità
- Reportistica sui risultati

SOLUZIONI PARTE A

1. c) 2. c) 3. b) 4. c) 5. a) 6. c) 7. d) 8. b) 9. b) 10. a)
11. c) 12. d) 13. a) 14. d) 15. b) 16. b) 17. c) 18. a) 19. b) 20. c)

Questa raccolta copre tutti i principali argomenti del corso e riflette il tipo di domande tipicamente poste negli esami reali, bilanciando teoria e pratica secondo lo stile richiesto dal corso.