

PARTE I: Domande a Risposta Multipla (32 punti)

Selezionare la risposta corretta per ciascuna domanda. Ogni risposta corretta vale 1 punto.

1. Quale acronimo descrive le caratteristiche del processo di build automation?

- ☐ A. CARE: Continuous, Automated, Repeatable, Executable
- ☐ B. CATS: Complete, Automated, Testable, Schedulable
- ☐ C. CRISP: Completo, Ripetibile, Informativo, Schedulabile, Portabile
- ☐ D. CRAFT: Continuous, Repeatable, Automated, Fast, Testable

2. Nel contesto del software testing, quali tra i seguenti sono test non funzionali? (Selezionare due opzioni)

- ☐ A. Test di unità
- ☐ B. Test di sicurezza
- ☐ C. Test di integrazione
- ☐ D. Test di performance
- ☐ E. Test di accettazione

3. Quale delle seguenti affermazioni riguardo alla Continuous Delivery NON è corretta?

- ☐ A. Ogni cambiamento al sistema può potenzialmente essere rilasciato in produzione
- ☐ B. Il rilascio può essere fatto in ogni momento in ogni ambiente premendo un bottone
- ☐ C. Il processo di build per la produzione dei file binari deve essere eseguito più volte (una per ogni ambiente dove verrà rilasciato il prodotto)
- ☐ D. La Continuous Delivery è il passo successivo alla Continuous Integration

4. Quali sono le caratteristiche principali di Apache Maven? (Selezionare due opzioni)

- ☐ A. È un tool di analisi statica del codice specializzato per progetti Java
- ☐ B. Gestisce automaticamente le dipendenze del progetto
- ☐ C. Definisce Build Lifecycle che permettono di configurare ed eseguire il processo di build
- ☐ D. È un Distributed Version Control System
- ☐ E. È un componente di Jenkins per l'esecuzione di job

5. Quale tra le seguenti NON è una caratteristica di un processo di build?

- ☐ A. Completo: Indipendente da fonti non specificate nello script di build
- ☐ B. Ripetibile: Accede ai file contenuti nel sistema di gestione del codice sorgente
- ☐ C. Schedulabile: Può essere programmato ed eseguito automaticamente
- ☐ D. Non Portabile: Deve essere fortemente legato all'ambiente di esecuzione

6. A quali categorie appartengono i test di unità? (Selezionare tre opzioni)

- ☐ A. Test di verifica (non validazione)
- ☐ B. Test non funzionale
- ☐ C. Statico
- ☐ D. Dinamico
- ☐ E. White box
- ☐ F. Funzionale
- ☐ G. Black box

7. Quale tra le seguenti è la definizione migliore di Artifact Repository?

- ☐ A. È un sistema di controllo versione centralizzato
- ☐ B. È uno strumento software progettato per ottimizzare il download e l'archiviazione dei file binari utilizzati e prodotti nello sviluppo del software
- ☐ C. È un database che contiene la configurazione di tutti i componenti di un'applicazione
- ☐ D. È un componente di Docker per archiviare le immagini dei container

8. Nel framework Scrum, qual è il ruolo responsabile dell'adozione dei valori e delle pratiche Scrum?

- ☐ A. Product Owner
- ☐ B. Scrum Master
- ☐ C. Development Team
- ☐ D. Project Manager

9. In Git, un insieme di modifiche a uno o più file che sono state esplicitamente validate e salvate nel repository rappresentano un:

- ☐ A. Branch
- ☐ B. Merge

- ☐ C. Commit
- ☐ D. Pull Request

10. Quali tra le seguenti sono le funzioni principali di un Issue Tracking System? (Selezionare due opzioni)

- ☐ A. Condividere le informazioni con il team di sviluppo, il Project Manager e il cliente
- ☐ B. Avere un'istantanea delle attività da fare, completate e in corso d'opera
- ☐ C. Automatizzare il processo di build e test del software
- ☐ D. Memorizzare e versionare il codice sorgente
- ☐ E. Eseguire analisi statica del codice

11. Quale tra le seguenti affermazioni sui test di integrazione è vera?

- ☐ A. I test di integrazione sono più veloci da configurare ed eseguire rispetto ai test di unità
- ☐ B. I test di integrazione verificano il comportamento dell'intero sistema
- ☐ C. I test di integrazione verificano se sono rispettati i contratti di interfaccia tra più moduli o subsystem
- ☐ D. I test di integrazione verificano l'unità, il più piccolo sottosistema possibile

12. Quale strumento di analisi statica del codice permette di verificare se c'è un miglioramento o un deterioramento della qualità del progetto nel tempo?

- ☐ A. SonarQube
- ☐ B. FindBugs/SpotBugs
- ☐ C. Checkstyle
- ☐ D. PMD

13. Quale delle seguenti è la differenza principale tra CVCS (Centralized Version Control System) e DVCS (Distributed Version Control System)?

- ☐ A. I CVCS non permettono di creare branch, mentre i DVCS sì
- ☐ B. I CVCS hanno un singolo punto di rottura (server centrale), mentre nei DVCS ogni sviluppatore ha una copia completa del repository
- ☐ C. I CVCS sono open source, mentre i DVCS sono sempre a pagamento
- ☐ D. I CVCS permettono la collaborazione tra sviluppatori, i DVCS no

14. Quali tra le seguenti sono configurazioni che possono essere specificate nel file pom.xml di Maven? (Selezionare

due opzioni)

- ☐ A. Le dipendenze del progetto
- ☐ B. I risultati dell'esecuzione dei test di unità
- ☐ C. L'identificativo del progetto (Gruppo, Artefatto e Versione)
- ☐ D. I file da ignorare nel sistema di versioning
- ☐ E. Le issue e i bug registrati nel progetto

15. Quale principio NON appartiene ai 7 principi del software testing?

- ☐ A. I test esaustivi non esistono
- ☐ B. Il testing mostra la presenza di difetti, non la loro assenza
- ☐ C. Un buon testing garantisce l'assenza di bug nel software
- ☐ D. Il paradosso del pesticida: gli stessi test ripetuti trovano sempre meno difetti
- ☐ E. Testare prima permette di risparmiare sui costi del progetto

16. Quali tra i seguenti test possono essere eseguiti nel processo di integrazione continua? (Selezionare due opzioni)

- ☐ A. Test di unità
- ☐ B. Test di sistema in ambiente di produzione
- ☐ C. Test di performance completi
- ☐ D. Test di integrazione tra subsystem interni
- ☐ E. Analisi statica del codice

17. Quale delle seguenti affermazioni riguardanti la Continuous Integration è corretta?

- ☐ A. Se il processo di build fallisce, si può commentare temporaneamente il test che ha causato il fallimento
- ☐ B. Il processo di build deve includere il rilascio automatico in produzione
- ☐ C. I membri del team integrano il loro lavoro frequentemente, almeno una volta al giorno
- ☐ D. La Continuous Integration richiede sempre l'utilizzo di Jenkins

18. In Scrum, quale evento viene utilizzato per sincronizzare le attività e creare un piano per le prossime 24 ore?

- ☐ A. Sprint Planning
- ☐ B. Daily Scrum
- ☐ C. Sprint Review
- ☐ D. Sprint Retrospective

19. Quale tra le seguenti è la migliore definizione per Workflow di un work item in un Issue Tracking System?

- ☐ A. È l'insieme delle persone coinvolte nella gestione di un work item
- ☐ B. È l'insieme di stati e transizioni che un work item attraversa durante il suo ciclo di vita
- ☐ C. È l'insieme delle stime di tempo necessarie per completare un work item
- ☐ D. È l'insieme dei commenti associati a un work item

20. Quale delle seguenti è una caratteristica propria di Git rispetto a SVN?

- ☐ A. Git memorizza snapshot completi del filesystem, mentre SVN memorizza le differenze tra versioni
- ☐ B. Git può gestire solo progetti software, mentre SVN può gestire qualsiasi tipo di file
- ☐ C. Git richiede una connessione costante al server, SVN funziona anche offline
- ☐ D. Git non supporta branching e merging, SVN sì

21. Nel Test Driven Development (TDD), qual è l'ordine corretto delle tre fasi?

- ☐ A. Write Production Code, Write Test, Refactor
- ☐ B. Refactor, Write Test, Write Production Code
- ☐ C. Write Test, Write Production Code, Refactor
- ☐ D. Write Test, Refactor, Write Production Code

22. Quale principio della Continuous Delivery stabilisce che bisogna costruire l'artefatto una sola volta e riutilizzarlo in tutte le fasi?

- ☐ A. Deploy the Same Way to Every Environment
- ☐ B. Smoke-Test Your Deployments
- ☐ C. Only Build Your Binaries Once
- ☐ D. If Any Part of the Pipeline Fails, Stop the Line

23. Quali sono le caratteristiche principali di Docker rispetto alle macchine virtuali? (Selezionare due opzioni)

- ☐ A. Docker richiede meno risorse poiché i container condividono il kernel del sistema operativo host
- ☐ B. Docker è più sicuro perché isola completamente i container
- ☐ C. I container si avviano in secondi, mentre le VM richiedono minuti
- ☐ D. I container sono sempre più stabili delle VM

- ☐ E. Docker funziona solo su Linux, le VM su qualsiasi sistema operativo

24. Quale delle seguenti NON è una caratteristica di Ansible?

- ☐ A. Non richiede l'installazione di agent sui nodi gestiti
- ☐ B. È multiplatforma
- ☐ C. Richiede l'installazione di un database centrale per funzionare
- ☐ D. È idempotente
- ☐ E. Ha una curva di apprendimento bassa

25. Quali sono i ruoli ufficiali definiti nel framework Scrum? (Selezionare tre opzioni)

- ☐ A. Product Owner
- ☐ B. Scrum Master
- ☐ C. Development Team
- ☐ D. Project Manager
- ☐ E. Tester
- ☐ F. Technical Lead

26. Quale dei seguenti è un vantaggio dell'utilizzo di un Artifact Repository?

- ☐ A. Elimina la necessità di test di unità
- ☐ B. Sostituisce completamente il Version Control System
- ☐ C. Aumenta la velocità del processo di build grazie alla cache locale delle dipendenze
- ☐ D. Semplifica la modifica del codice sorgente

27. Nel contesto di Configuration Management, cosa significa "idempotenza"?

- ☐ A. La capacità di un sistema di scalare automaticamente
- ☐ B. La possibilità di ripristinare un sistema a uno stato precedente
- ☐ C. L'impossibilità di modificare una configurazione una volta applicata
- ☐ D. L'applicazione della stessa operazione più volte produce sempre lo stesso risultato

28. Quali sono i componenti principali dell'architettura di Kubernetes? (Selezionare due opzioni)

- ☐ A. Pod
- ☐ B. Network
- ☐ C. Registry

- ☐ D. Node
- ☐ E. Container

29. Quale affermazione sulle property A-TRIP dei test di unità è corretta?

- ☐ A. A-TRIP significa Accurate, Testable, Repeatable, Independent, Portable
- ☐ B. La "T" in A-TRIP sta per "Thorough" (esaustivi)
- ☐ C. La "P" in A-TRIP sta per "Performance"
- ☐ D. A-TRIP è un modello per l'analisi statica del codice

30. Quale delle seguenti è la definizione corretta di Continuous Deployment?

- ☐ A. È sinonimo di Continuous Delivery
- ☐ B. È un'estensione della Continuous Delivery in cui tutte le modifiche che superano i test automatizzati vengono rilasciate automaticamente in produzione
- ☐ C. È un prerequisito per la Continuous Integration
- ☐ D. È una pratica che automatizza solo la fase di testing

PARTE II: Domande Teoriche (16 punti)

Rispondere in modo esaustivo alle seguenti domande. Ogni risposta vale 4 punti.

1. Descrivere il principio "Deploy the Same Way to Every Environment" della Continuous Delivery. Spiegare perché è importante e come può essere implementato.

2. Descrivere le fasi principali del processo di Continuous Integration. Quali sono i prerequisiti necessari e quali best practice dovrebbero essere seguite per implementarlo correttamente?

3. Spiegare il concetto di "Feedback Loop" nel contesto delle pipeline di Continuous Delivery/Deployment. Quali strumenti possono essere utilizzati per implementarlo e quali attori dovrebbero essere coinvolti?

4. Descrivere il Test Driven Development (TDD) e il suo ciclo Red-Green-Refactor. Quali sono i vantaggi e gli svantaggi di questo approccio?

PARTE III: Prove Pratiche (12 punti)

Eseguire le seguenti prove pratiche. Ogni esercizio vale 3 punti.

1. Git e GitHub

a) Mostrare i comandi Git necessari per:

- Creare un nuovo repository locale
- Aggiungere un file README.md con il contenuto "# Progetto TOS"
- Committare il file
- Aggiungere un repository remoto su GitHub
- Pushare il commit al repository remoto

b) Come si può associare una commit a una issue specifica di GitHub? Fornire un esempio di messaggio di commit che chiude automaticamente l'issue #42.

2. JUnit

a) Scrivere una classe Java `Calcolatrice` con due metodi:

- `int somma(int a, int b)` : restituisce la somma di due numeri
- `int dividi(int a, int b)` : restituisce la divisione di `a` per `b` , lanciando un'eccezione se `b` è zero

b) Scrivere i test JUnit per verificare entrambi i metodi, incluso il test per l'eccezione.

3. Maven

a) Scrivere un file `pom.xml` base per un progetto con le seguenti caratteristiche:

- GroupId: `it.unipd.math`
- ArtifactId: `progetto-tos`
- Version: `1.0-SNAPSHOT`
- Dipendenza da JUnit 4.13
- Configurazione del plugin `maven-compiler-plugin` per utilizzare Java 11

b) Qual è il comando Maven per eseguire solo i test di unità? E per creare il pacchetto JAR del progetto?

4. GitHub Actions

Scrivere un file YAML di GitHub Actions (`.github/workflows/ci.yml`) che:

- Si attiva ad ogni push sul branch `main`
- Esegue il checkout del repository

- Configura un ambiente Java 11
- Compila il progetto con Maven
- Esegue i test di unità
- Produce un report dei test falliti

CHIAVE DI RISPOSTA - PARTE I

1. C
2. B, D
3. C
4. B, C
5. D
6. A, D, F
7. B
8. B
9. C
10. A, B
11. C
12. A
13. B
14. A, C
15. C
16. A, E
17. C
18. B
19. B
20. A
21. C
22. C
23. A, C
24. C
25. A, B, C
26. C
27. D
28. A, D
29. B
30. B