

1. Cosa è un metodo ?

☐ \_\_\_\_\_ Un sistema per fare qualcosa in modo migliore

**V** ☒ \_\_\_\_\_ Un procedimento generale per risolvere classi di problemi

☐ \_\_\_\_\_ Un particolare approccio o filosofia per fare qualcosa

2. Quale di queste affermazioni non è appropriata per il modello di processo a cascata ?

☐ \_\_\_\_\_ Le fasi del processo sono in progressione sequenziale

☐ \_\_\_\_\_ I semilavorati all'uscita di una fase sono congelati e non possono essere più modificati

**V** ☒ \_\_\_\_\_ Le funzionalità del sistema sono sviluppate in maniera incrementale

3. Un prototipo viene sviluppato per:

☐ \_\_\_\_\_ Ridurre i costi di sviluppo

☐ \_\_\_\_\_ Rilasciare rapidamente al cliente una prima versione del sistema

**V** ☒ \_\_\_\_\_ Interagire con il committente per convalidare i requisiti

4. Quale tipo di prototipazione parte con i requisiti meglio compresi ?

**V** ☒ \_\_\_\_\_ Prototipazione esplorativa

☐ \_\_\_\_\_ Prototipazione throw-away

☐ \_\_\_\_\_ Prototipazione mock-ups

5. Quale tipo di prototipazione tende a realizzare l'interfaccia utente?

☐ \_\_\_\_\_ Prototipazione esplorativa

☐ \_\_\_\_\_ Prototipazione breadboards

**V** ☒ \_\_\_\_\_ Prototipazione mock-ups

6. Quale di queste affermazioni è appropriata per il modello di processo incrementale ?

☐ \_\_\_\_\_ Ogni versione produce funzionalità/sottosistemi più affidabili

**V** ☒ \_\_\_\_\_ Ogni versione aggiunge nuove funzionalità/sottosistemi al sistema

☐ \_\_\_\_\_ Ogni versione raffina le funzionalità/sottosistemi che sono presenti fin dall'inizio

7. Quale di queste affermazioni è appropriata per il modello di processo iterativo ?

**V** ☒ \_\_\_\_\_ Ogni versione raffina le funzionalità/sottosistemi che sono presenti fin dall'inizio

☐ \_\_\_\_\_ Ogni versione aggiunge nuove funzionalità/sottosistemi al sistema

☐ \_\_\_\_\_ I requisiti a più alta priorità vengono rilasciati per primi

8. Cosa è un team ?

**V** ☐ \_\_\_\_\_ Un insieme di persone a cui è assegnato un task comune, ma che lavorano individualmente e senza necessità di interazione

☐ \_\_\_\_\_ Un insieme di persone che rivedono un work product e che propongono azioni

☐ \_\_\_\_\_ Un piccolo insieme di persone che lavorano in stretta interazione sulla stessa attività o task

9. Il tempo durante il quale un certo lavoro in un progetto deve essere fatto, viene denominato:

☐ \_\_\_\_\_ Work-product

**V** ☐ \_\_\_\_\_ Schedule

☐ \_\_\_\_\_ Task

10. Il lavoro che deve essere eseguito da un partecipante al progetto, viene denominato:

☐ \_\_\_\_\_ Work-product

☐ \_\_\_\_\_ Schedule

**V** ☐ \_\_\_\_\_ Task

11. Quale di queste affermazioni non è valida per una struttura organizzativa di reporting gerarchica ?

**V** ☐ \_\_\_\_\_ Lo stato è riportato dall'alto verso il basso

☐ \_\_\_\_\_ Le decisioni vengono comunicate dall'alto verso il basso

☐ \_\_\_\_\_ Le informazioni relative a stato e decisioni sono unidirezionali

12. Cosa è un liason all'interno di un team ?

☐ \_\_\_\_\_ Il leader del team

**V** ☐ \_\_\_\_\_ Il responsabile della comunicazione con un altro team

☐ \_\_\_\_\_ Il consulente amministrativo del team

13. Che tipo di ruolo è un configuration manager ?

☐ \_\_\_\_\_ Uno sviluppatore

☐ \_\_\_\_\_ Un manager

**V** ☐ \_\_\_\_\_ Un liason

15. Che cosa è un PERT?

☐ \_\_\_\_\_ Un grafico a barre che descrive lo schedule e la durata dei vari task

**V** ☐ \_\_\_\_\_ Un grafo che descrive le dipendenze tra i task del progetto

☐ \_\_\_\_\_ Un grafo che descrive le dipendenze tra i work-product del progetto

16. Quale dei seguenti **non** è una comunicazione pianificata?

☐ \_\_\_\_\_ Definizione del problema

☐ \_\_\_\_\_ Review di progetto

**V** ☐ \_\_\_\_\_ Richiesta di modifica

17. Quale dei seguenti è una comunicazione **non** pianificata?

**V** ☐ \_\_\_\_\_ Risoluzione di un problema

☐ \_\_\_\_\_ Ispezione

☐ \_\_\_\_\_ Rilascio

18. Quale dei seguenti meccanismi di comunicazione è asincrono?

☐ \_\_\_\_\_ Intervista strutturata

☐ \_\_\_\_\_ Riunione

**V** ☐ \_\_\_\_\_ Posta elettronica

19. Cosa è un modello ?

**V** ☐ \_\_\_\_\_ Un'astrazione che descrive il sistema o un sottoinsieme di un sistema

☐ \_\_\_\_\_ Un insieme di regole grafiche o testuali per rappresentare viste

☐ \_\_\_\_\_ Una vista di aspetti del sistema

20. Cosa è una vista ?

☐ \_\_\_\_\_ Un'astrazione che descrive il sistema o un sottoinsieme di un sistema

**V** ☐ \_\_\_\_\_ Una visualizzazione di particolari aspetti di un modello

☐ \_\_\_\_\_ Un insieme di regole grafiche o testuali

21. Cosa è una notazione ?

☐ \_\_\_\_\_ Un'astrazione che descrive il sistema o un sottoinsieme di un sistema

☐ \_\_\_\_\_ Una visualizzazione di particolari aspetti di un modello

**V** ☐ \_\_\_\_\_ Un insieme di regole grafiche o testuali per rappresentare viste

22. Cosa è UML ?

☐ \_\_\_\_\_ Una notazione grafica per progettare sistemi software

**V** ☐ \_\_\_\_\_ Un insieme di linguaggi per modellare software

☐ \_\_\_\_\_ Un modello astratto per descrivere sistemi software

23. Quale di questi diagrammi **non** è usato per descrivere il comportamento dinamico di un sistema software ?

☐ \_\_\_\_\_ Sequence diagram

☐ \_\_\_\_\_ Activity diagram

**V** ☒ \_\_\_\_\_ Use case diagram

24. I diagrammi dei casi d'uso:

**V** ☒ \_\_\_\_\_ Descrivono il comportamento funzionale del sistema così come visto dagli utenti

☐ \_\_\_\_\_ Descrivono il comportamento dinamico tra gli attori e il sistema e tra gli oggetti del sistema

☐ \_\_\_\_\_ Descrivono il comportamento dinamico di un sistema, in particolare il workflow

25. I diagrammi delle attività:

☐ \_\_\_\_\_ Descrivono il comportamento funzionale del sistema così come visto dagli utenti

☐ \_\_\_\_\_ Descrivono il comportamento dinamico tra gli attori e il sistema e tra gli oggetti del sistema

**V** ☒ \_\_\_\_\_ Descrivono il comportamento dinamico di un sistema, in particolare il workflow

26. I diagrammi di sequenza:

☐ \_\_\_\_\_ Descrivono il comportamento funzionale del sistema così come visto dagli utenti

**V** ☒ \_\_\_\_\_ Descrivono il comportamento dinamico tra gli attori e il sistema e tra gli oggetti del sistema

☐ \_\_\_\_\_ Descrivono il comportamento dinamico di un sistema, in particolare il workflow

27. I diagrammi di stato:

☐ \_\_\_\_\_ Descrivono la struttura statica del sistema: oggetti, attributi e relazioni

☐ \_\_\_\_\_ Descrivono il comportamento dinamico tra gli attori e il sistema e tra gli oggetti del sistema

**V** ☒ \_\_\_\_\_ Descrivono il comportamento dinamico di un singolo oggetto come una macchina a stati

finiti

28. I diagrammi delle classi:

**V** ☐ \_\_\_\_\_ Descrivono la struttura statica del sistema: oggetti, attributi e relazioni

☐ \_\_\_\_\_ Descrivono il comportamento dinamico tra gli attori e il sistema e tra gli oggetti del sistema

☐ \_\_\_\_\_ Descrivono il comportamento dinamico di un singolo oggetto come una macchina a stati

finiti

29. In UML le classi e gli oggetti (istanze) sono rappresentate con:

**V** ☐ \_\_\_\_\_ Rettangoli

☐ \_\_\_\_\_ Ovali

☐ \_\_\_\_\_ Rettangoli con angoli arrotondati

30. In UML i casi d'uso sono rappresentati con:

☐ \_\_\_\_\_ Rettangoli

**V** ☐ \_\_\_\_\_ Ovali

☐ \_\_\_\_\_ Rettangoli con angoli arrotondati

31. In UML gli stati sono rappresentati con:

☐ \_\_\_\_\_ Rettangoli

☐ \_\_\_\_\_ Ovali

**V** ☐ \_\_\_\_\_ Rettangoli con angoli arrotondati

32. Quale di questi elementi non è parte di un diagramma dei casi d'uso:

☐ \_\_\_\_\_ Attore

☐ \_\_\_\_\_ Caso d'uso

**V** ☐ \_\_\_\_\_ Classe

33. Quale di questi elementi non è parte di un diagramma di sequenza:

☐ \_\_\_\_\_ Attore

**V** ☐ \_\_\_\_\_ Caso d'uso

☐ \_\_\_\_\_ Oggetti

34. Che tipo di relazione può esistere tra un attore e un caso d'uso:

**V** ☐ \_\_\_\_\_ Associazione

☐ \_\_\_\_\_ Dipendenza

☐ \_\_\_\_\_ Generalizzazione

35. Che tipo di relazione può esistere tra due attori:

☐ \_\_\_\_\_ Associazione

☐ \_\_\_\_\_ Dipendenza

**V** ☒ \_\_\_\_\_ Generalizzazione

36. Quale di queste relazioni **non** esiste tra due casi d'uso:

**V** ☒ \_\_\_\_\_ Associazione

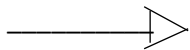
☐ \_\_\_\_\_ Dipendenza

☐ \_\_\_\_\_ Generalizzazione

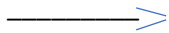
37. Quali di queste relazioni rappresenta una dipendenza:

----->

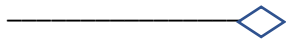
38. Quali di queste relazioni rappresenta una generalizzazione:



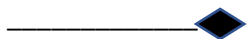
39. Quali di queste relazioni rappresenta una associazione:



40. Quali di queste relazioni rappresenta una aggregazione



41. Quali di queste relazioni rappresenta una composizione:



42. Quali di questi è un requisito funzionale:

**V** ☒ \_\_\_\_\_ Il sistema deve visualizzare l'ora in base alla sua locazione

☐ \_\_\_\_\_ Il tempo di risposta deve essere meno di un secondo

☐ \_\_\_\_\_ Il linguaggio di implementazione deve essere Java

43. Quali di questi è un requisito non funzionale:

☐ \_\_\_\_\_ Il sistema deve visualizzare l'ora in base alla sua locazione

**V** ☒ \_\_\_\_\_ Il tempo di risposta deve essere meno di un secondo

☐ \_\_\_\_\_ Il linguaggio di implementazione deve essere Java

44. Cosa si intende per correttezza dei requisiti ?

**V** ☒ \_\_\_\_\_ I requisiti rappresentano la vista dell'utente

☐ Sono descritti tutti i possibili scenari del sistema

☐ Non ci sono requisiti funzionali e non funzionali che si contraddicono

45. Cosa si intende per completezza dei requisiti ?

☐ I requisiti rappresentano la vista dell'utente

**V** ☒ Sono descritti tutti i possibili scenari del sistema

☐ Non ci sono requisiti funzionali e non funzionali che si contraddicono

46. Cosa si intende per consistenza dei requisiti ?

☐ I requisiti rappresentano la vista dell'utente

☐ Sono descritti tutti i possibili scenari del sistema

**V** ☒ Non ci sono requisiti funzionali e non funzionali che si contraddicono

47. Cosa si intende per chiarezza dei requisiti ?

**V** ☒ Non ci sono ambiguità nei requisiti

☐ I requisiti possono essere implementati e rilasciati

☐ Non ci sono requisiti funzionali e non funzionali che si contraddicono

48. Cosa si intende per realismo dei requisiti ?

☐ Non ci sono ambiguità nei requisiti

**V** ☒ I requisiti possono essere implementati e rilasciati

☐ Per ogni funzione del sistema è possibile individuare un insieme di requisiti funzionali

49. Cosa si intende per tracciabilità dei requisiti ?

☐ Non ci sono ambiguità nei requisiti

☐ I requisiti possono essere implementati e rilasciati

**V** ☒ Per ogni funzione del sistema è possibile individuare un insieme di requisiti funzionali

50. Quale di questi requisiti vanno affrontati durante il design ma non durante l'implementazione ?

☐ Requisiti ad alta priorità

**V** ☒ Requisiti a media priorità

☐ Requisiti a bassa priorità

51. Cosa si intende per interface engineering ?

☐ Lo sviluppo parte da zero, non esiste un sistema precedente

☐ \_\_\_\_\_ Riprogettazione o reimplementazione di un sistema esistente con nuove tecnologie

**V** ☐ \_\_\_\_\_ Fornire i servizi di un sistema esistente in un nuovo ambiente operativo

52. Che tipo di scenari sono usati per descrivere un sistema futuro?

☐ \_\_\_\_\_ as-is scenarios

**V** ☐ \_\_\_\_\_ visionary scenarios

☐ \_\_\_\_\_ training scenarios

53. Che tipo di scenari sono usati per descrivere un sistema esistente ?

**V** ☐ \_\_\_\_\_ as-is scenarios

☐ \_\_\_\_\_ visionary scenarios

☐ \_\_\_\_\_ training scenarios

54. Che tipo di scenari sono usati per guidare un utente nuovo nell'utilizzo del sistema ?

☐ \_\_\_\_\_ as-is scenarios

☐ \_\_\_\_\_ visionary scenarios

**V** ☐ \_\_\_\_\_ training scenarios

55. Perché si usano gli scenari ?

**V** ☐ \_\_\_\_\_ Per individuare i requisiti del sistema

☐ \_\_\_\_\_ Per analizzare le associazioni tra gli oggetti del sistema

☐ \_\_\_\_\_ Per analizzare il flusso di eventi di un caso d'uso

56. Come si descrive un caso d'uso ?

☐ \_\_\_\_\_ Mediante uno scenario

**V** ☐ \_\_\_\_\_ Mediante un flusso di eventi

☐ \_\_\_\_\_ Mediante un insieme di requisiti funzionali

57. Cosa esprime una relazione di inclusione tra casi d'uso ?

☐ \_\_\_\_\_ Una variante del normale flusso di eventi di un caso d'uso

☐ \_\_\_\_\_ Una specializzazione di un caso d'uso

**V** ☐ \_\_\_\_\_ Una decomposizione funzionale di un caso d'uso

58. Cosa esprime una relazione di estensione tra casi d'uso ?

**V** ☐ \_\_\_\_\_ Una variante del normale flusso di eventi di un caso d'uso



☐ \_\_\_\_\_ Una specializzazione di un caso d'uso

☐ \_\_\_\_\_ Una decomposizione funzionale di un caso d'uso

59. Cosa esprime una relazione di generalizzazione tra casi d'uso ?

☐ \_\_\_\_\_ Una variante del normale flusso di eventi di un caso d'uso

**V** ☒ \_\_\_\_\_ Una specializzazione di un caso d'uso

☐ \_\_\_\_\_ Una decomposizione funzionale di un caso d'uso

60. Un oggetto che interagisce con un attore in un sequence diagram è:

☐ \_\_\_\_\_ Un entity object

**V** ☒ \_\_\_\_\_ Un boundary object

☐ \_\_\_\_\_ Un control object

61. Quale di queste categorie di requisiti non funzionali indica la facilità di cambiamenti al sistema dopo il rilascio ?

☐ \_\_\_\_\_ Reliability

☐ \_\_\_\_\_ Performance

**V** ☒ \_\_\_\_\_ Supportability

62. Quale di queste categorie di pseudo-requirements indica vincoli legati alle modalità di rilascio del sistema?

☐ \_\_\_\_\_ Interface requirements

**V** ☒ \_\_\_\_\_ Packaging requirements

☐ \_\_\_\_\_ Operations requirements

63. Una trasformazione applicata agli oggetti di una classe e definita in fase di analisi è chiamata:

**V** ☒ \_\_\_\_\_ Operazione

☐ \_\_\_\_\_ Signature

☐ \_\_\_\_\_ Metodo

64. Una connessione tra due istanze di oggetti è chiamata:

☐ \_\_\_\_\_ Associazione

**V** ☒ \_\_\_\_\_ Link

☐ \_\_\_\_\_ Relazione

65. Una associazione è:

☐ \_\_\_\_\_ Una connessione tra due istanze di oggetti

☐ \_\_\_\_\_ Una dipendenza tra classi

**V** ☒ \_\_\_\_\_ Un mapping bidirezionale tra classi

66. Quale di questi tipi di oggetti cambia meno frequentemente in un sistema software ?

☐ \_\_\_\_\_ Boundary object

☐ \_\_\_\_\_ Control object

**V** ☒ \_\_\_\_\_ Entity object

67. Cosa indica un ruolo ?

**V** ☒ \_\_\_\_\_ Una estremità dell'associazione

☐ \_\_\_\_\_ La navigabilità dell'associazione

☐ \_\_\_\_\_ La direzione del nome dell'associazione

68. Cosa indica la direzione di una associazione ?

☐ \_\_\_\_\_ Una estremità dell'associazione

**V** ☒ \_\_\_\_\_ La navigabilità dell'associazione

☐ \_\_\_\_\_ La direzione del nome dell'associazione

69. Nel diagramma seguente, "filename":

☐ \_\_\_\_\_ E' un attributo della classe directory

**V** ☒ \_\_\_\_\_ E' un attributo della classe file

☐ \_\_\_\_\_ E' il nome della associazione

70. Cosa indicano le frecce in un diagramma di sequenza ?

**V** ☒ \_\_\_\_\_ Eventi inviati da un oggetto ad un altro

☐ \_\_\_\_\_ Relazioni tra oggetti

☐ \_\_\_\_\_ Dipendenze tra oggetti

71. Cosa indica la linea tratteggiata verticale sotto un oggetto in un diagramma di sequenza ?

☐ \_\_\_\_\_ L'invio di un messaggio all'oggetto

☐ \_\_\_\_\_ Il periodo di attivazione di una operazione dell'oggetto

**V** ☒ \_\_\_\_\_ Il periodo durante il quale è possibile inviare un messaggio all'oggetto

72. Cosa indica un rettangolo verticale sotto un oggetto in un diagramma di sequenza ?

☐ \_\_\_\_\_ L'invio di un messaggio all'oggetto

**V** ☐ \_\_\_\_\_ Il periodo di attivazione di una operazione dell'oggetto

☐ \_\_\_\_\_ Il periodo durante il quale è possibile inviare un messaggio all'oggetto

73. Cosa dovrebbe contenere la prima colonna di un diagramma di sequenza ?

**V** ☐ \_\_\_\_\_ L'attore che inizia il caso d'uso

☐ \_\_\_\_\_ Un boundary object

☐ \_\_\_\_\_ Il control object che gestisce il caso d'uso

74. Cosa dovrebbe contenere la seconda colonna di un diagramma di sequenza ?

☐ \_\_\_\_\_ L'attore che inizia il caso d'uso

**V** ☐ \_\_\_\_\_ Un boundary object

☐ \_\_\_\_\_ Il control object che gestisce il caso d'uso

75. Cosa dovrebbe contenere la terza colonna di un diagramma di sequenza ?

☐ \_\_\_\_\_ L'attore che inizia il caso d'uso

☐ \_\_\_\_\_ Un boundary object

**V** ☐ \_\_\_\_\_ Il control object che gestisce il caso d'uso

76. Quali di queste affermazioni **non** è corretta ?

☐ \_\_\_\_\_ Un boundary object accede ad un entity object

**V** ☐ \_\_\_\_\_ Un entity object accede ad un control object

☐ \_\_\_\_\_ Un control object crea un boundary object

77. In un diagramma di stato, quando viene valutata la condizione di guardia su una transizione ?

☐ \_\_\_\_\_ Quando il sistema si trova nello stato sorgente della transizione

☐ \_\_\_\_\_ Quando si verifica l'azione indicata sulla transizione

**V** ☐ \_\_\_\_\_ Quando si verifica l'evento indicato sulla transizione

78. Quale di queste affermazioni non si riferisce ad **un'attività in** un diagramma di stato ?

☐ \_\_\_\_\_ Operazione che prende tempo per essere completata

**V** ☐ \_\_\_\_\_ E' associata ad un evento

☐ \_\_\_\_\_ E' associata ad uno stato

79. Quale di queste affermazioni non si riferisce **ad un'azione** in un diagramma di stato ?

**V** ☐ \_\_\_\_\_ Operazione che prende tempo per essere completata

☐ \_\_\_\_\_ E' associata ad uno stato

☐ \_\_\_\_\_ E' associata ad un evento

80. Quale di queste affermazioni si riferisce ad un diagramma di stato ?

☐ \_\_\_\_\_ Serve ad individuare le relazioni temporali tra oggetti nel tempo

☐ \_\_\_\_\_ Serve ad individuare la sequenza di operazioni come risposta ad uno o più eventi

**V** ☒ \_\_\_\_\_ Serve ad individuare i cambiamenti degli oggetti nel tempo

81. Quale di queste affermazioni **non** si riferisce ad un diagramma di sequenza ?

**V** ☒ \_\_\_\_\_ Serve ad individuare i cambiamenti degli oggetti nel tempo

☐ \_\_\_\_\_ Serve ad individuare le relazioni temporali tra oggetti nel tempo

☐ \_\_\_\_\_ Serve ad individuare la sequenza di operazioni come risposta ad uno o più eventi

82. Quale di questi prodotti dell'analisi dei requisiti vanno usati durante la fase di definizione degli obiettivi di design ?

☐ \_\_\_\_\_ Modello dei casi d'uso

☐ \_\_\_\_\_ Modello a oggetti

**V** ☒ \_\_\_\_\_ Requisiti non funzionali

83. Quale di questi prodotti dell'analisi dei requisiti vanno usati durante la fase di decomposizione del sistema ?

**V** ☒ \_\_\_\_\_ Modello dei casi d'uso

☐ \_\_\_\_\_ Modello a oggetti

☐ \_\_\_\_\_ Requisiti non funzionali

84. Quale di questi prodotti dell'analisi dei requisiti vanno usati durante la fase di design relativa a mapping hardware/software e a gestione dei dati persistenti ?

☐ \_\_\_\_\_ Modello dei casi d'uso

**V** ☒ \_\_\_\_\_ Modello a oggetti

☐ \_\_\_\_\_ Requisiti non funzionali

85. Come viene rappresentato un sottosistema in UML ?

☐ \_\_\_\_\_ Con una classe

☐ \_\_\_\_\_ Con un caso d'uso

**V** ☒ \_\_\_\_\_ Con un package

86. Cosa è un servizio di un sottosistema ?

☐ \_\_\_\_\_ Un insieme di operazioni con signature completamente specificata

**V** ☒ \_\_\_\_\_ Un gruppo di operazioni che condividono uno scopo comune

☐ \_\_\_\_\_ Un insieme di associazioni, eventi e vincoli legati tra di loro

87. Cosa è l'interfaccia di un sottosistema ?

**V** ☒ \_\_\_\_\_ Un insieme di operazioni con signature completamente specificata

☐ \_\_\_\_\_ Un gruppo di operazioni che condividono uno scopo comune

☐ \_\_\_\_\_ Un insieme di associazioni, eventi e vincoli legati tra di loro

88. Quali di queste affermazioni è vera ?

☐ \_\_\_\_\_ L'accoppiamento misura le dipendenze tra le classi di un sottosistema

**V** ☒ \_\_\_\_\_ In un sistema con elevato accoppiamento le modifiche ad un sottosistema hanno forte impatto sugli altri sottosistemi

☐ \_\_\_\_\_ L'obiettivo del system design è massimizzare l'accoppiamento

89. Quali di queste affermazioni è vera ?

☐ \_\_\_\_\_ La coesione misura le dipendenze tra i sottosistemi di un sistema

☐ \_\_\_\_\_ In sottosistemi con elevata coesione le modifiche ad un sottosistema hanno forte impatto sugli altri sottosistemi

**V** ☒ \_\_\_\_\_ Le classi di un sottosistema con elevata coesione eseguono task simili

90. A cosa servono le partizioni di un sistema ?

☐ \_\_\_\_\_ A dividere verticalmente un sistema in sottosistemi debolmente accoppiati

**V** ☒ \_\_\_\_\_ A dividere orizzontalmente un sistema in sottosistemi a diversi livelli di astrazione

☐ \_\_\_\_\_ A dividere un sistema sia orizzontalmente che verticalmente in sottosistemi indipendenti

91. A cosa servono i layer di un sistema ?

**V** ☒ \_\_\_\_\_ A dividere verticalmente un sistema in sottosistemi debolmente accoppiati

☐ \_\_\_\_\_ A dividere orizzontalmente un sistema in sottosistemi a diversi livelli di astrazione

☐ \_\_\_\_\_ A dividere un sistema sia orizzontalmente che verticalmente in sottosistemi indipendenti

92. A chi fornisce servizi un layer di un sistema ?

☐ \_\_\_\_\_ Ai layer di livello più basso

☐ \_\_\_\_\_ Ai layer dello stesso livello

**V** ☒ \_\_\_\_\_ Ai layer di livello più alto

93. In un'architettura software aperta:

☐ \_\_\_\_\_ I sottosistemi di un livello possono accedere solo ai sottosistemi del livello immediatamente inferiore

**V** ☒ \_\_\_\_\_ I sottosistemi di un livello possono accedere ai sottosistemi di qualunque livello inferiore

☐ \_\_\_\_\_ I sottosistemi di un livello possono accedere sia ai sottosistemi dei livelli inferiori che a quelli dei livelli superiori

94. In un'architettura software chiusa:

**V** ☒ \_\_\_\_\_ I sottosistemi di un livello possono accedere solo ai sottosistemi del livello immediatamente inferiore

☐ \_\_\_\_\_ I sottosistemi di un livello possono accedere solo ai sottosistemi dello stesso livello

☐ \_\_\_\_\_ I sottosistemi di un livello possono accedere ai sottosistemi di qualunque livello inferiore

95. Quali sono gli obiettivi di design di un'architettura software aperta ?

☐ \_\_\_\_\_ Manutenibilità

**V** ☒ \_\_\_\_\_ Efficienza

☐ \_\_\_\_\_ Affidabilità

96. Quali sono gli obiettivi di design di un'architettura software chiusa ?

☐ \_\_\_\_\_ Usabilità

☐ \_\_\_\_\_ Efficienza

**V** ☒ \_\_\_\_\_ Manutenibilità

97. Quale di queste affermazioni è falsa?

☐ \_\_\_\_\_ In un'architettura client-server, il server fornisce i servizi al client

**V** ☐ In un'architettura client-server, il server conosce l'interfaccia del client

**V** ☐ Una repository architecture è un caso particolare di architettura client-server

98. Quali di queste affermazioni è vera ?

**V** ☐ In un'architettura peer-to-peer non c'è differenza tra client e server

☐ Un'architettura peer-to-peer favorisce una gestione centralizzata dei dati

☐ Un'architettura peer-to-peer consente di evitare problemi di deadlock

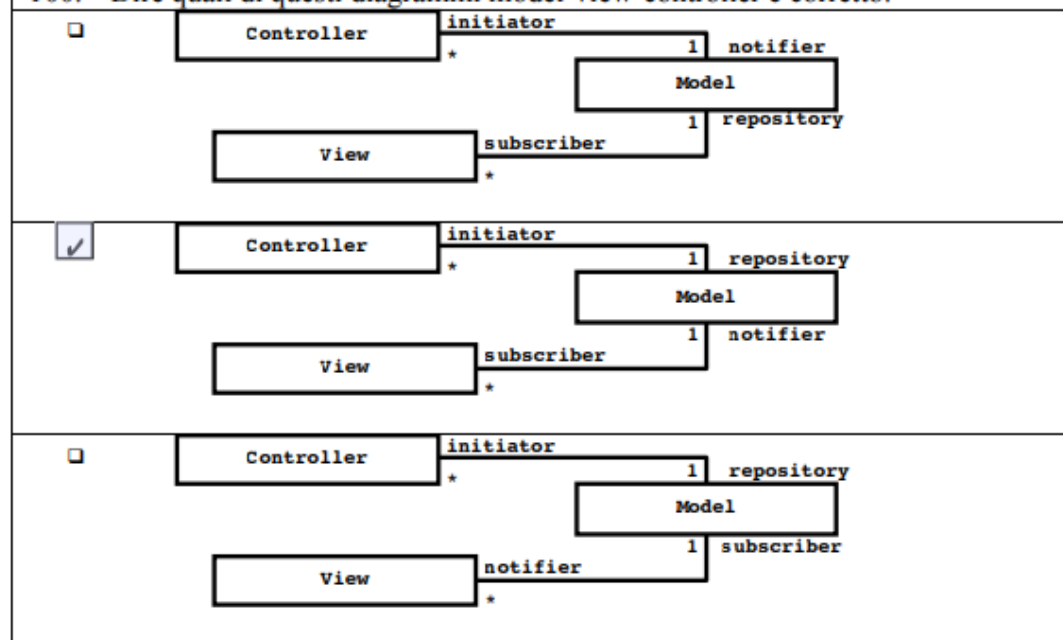
99. Un'architettura model-view-controller, il sottosistema "model":

**V** ☐ E' responsabile della conoscenza del dominio applicativo

☐ E' responsabile della visualizzazione degli oggetti del dominio applicativo

☐ E' responsabile della sequenza dei interazioni con l'utente

100. Dire quali di questi diagrammi model-view-controller è corretto:



101. Le relazioni tra i componenti in un component diagram sono

☐ associazioni

**V** ☐ dipendenze

☐ generalizzazioni

102. Quale di queste affermazioni relative ad un component diagram è falsa ?

**V** ☐ Mostra il mapping hardware/software

☐ \_\_\_\_\_ Mostra la struttura del sistema a compilation time

☐ \_\_\_\_\_ Mostra la struttura a design time

103. Quale di queste affermazioni relative ad un deployment diagram è falsa ?

☐ \_\_\_\_\_ Mostra la struttura del sistema a run-time

☐ \_\_\_\_\_ Mostra il mapping hardware/software

**V** ☒ \_\_\_\_\_ Mostra le dipendenze tra componenti e interfacce dei sottosistemi

104. Le relazioni tra i nodi di un deployment diagram sono:

**V** ☒ \_\_\_\_\_ associazioni

☐ \_\_\_\_\_ dipendenze

☐ \_\_\_\_\_ generalizzazioni

105. In un thin client model

**V** ☒ \_\_\_\_\_ Il sistema è two thier e la logica applicativa è eseguita sul server

☐ \_\_\_\_\_ Il sistema è two thier e la logica applicativa è eseguita sul client

☐ \_\_\_\_\_ Il sistema è three thier e la logica applicativa è eseguita sul client

106. Se il mio obiettivo è fornire implementazioni diverse per uno stesso sottosistema

**V** ☒ \_\_\_\_\_ Uso un bridge pattern

☐ \_\_\_\_\_ Uso un adapter pattern

☐ \_\_\_\_\_ Uso un façade pattern

107. Se il mio obiettivo è quello di realizzare una architettura chiusa

☐ \_\_\_\_\_ Uso un adapter pattern

☐ \_\_\_\_\_ Uso un bridge pattern

**V** ☒ \_\_\_\_\_ Uso un façade pattern

108. Un invariante è

☐ \_\_\_\_\_ Un predicato che deve essere vero prima dell'invocazione di un metodo di una classe

☐ \_\_\_\_\_ Un predicato che è vero dopo l'invocazione di un metodo di una classe

**V** ☒ \_\_\_\_\_ Un predicato che è vero prima e dopo l'invocazione di un metodo di una classe

109. Quale di queste affermazioni relative a JavaDoc è corretta ?

**V** ☒ \_\_\_\_\_ Il commento che precede un metodo consente di specificare la precondizione del metodo



☐ \_\_\_\_\_ Il commento che precede un metodo consente di specificare la postcondizione del metodo

☐ \_\_\_\_\_ Il commento che precede un metodo consente di specificare l'invariante della classe

110. Se devo realizzare una associazione qualificata uso come struttura dati

**V** ☒ \_\_\_\_\_ Una tabella

☐ \_\_\_\_\_ Un insieme

☐ \_\_\_\_\_ Una lista

111. Se devo estendere una classe mi interessa conoscere

☐ \_\_\_\_\_ Tutti i membri privati, protetti e pubblici della classe

**V** ☒ \_\_\_\_\_ Solo i membri protetti e pubblici della classe

☐ \_\_\_\_\_ Solo i membri pubblici della classe

112. La realizzazione delle associazioni del modello a oggetti

**V** ☒ \_\_\_\_\_ E' una trasformazione del modello a oggetti

☐ \_\_\_\_\_ E' una trasformazione di tipo forward engineering

☐ \_\_\_\_\_ E' una trasformazione di refactoring

113. Se devo memorizzare dati voluminosi e necessari per un periodo breve di tempo

☐ \_\_\_\_\_ uso una struttura dati in memoria

**V** ☒ \_\_\_\_\_ uso un file

☐ \_\_\_\_\_ uso un database

114. Se devo effettuare una trasformazione prima sul codice sorgente e poi sul modello a oggetti faccio

☐ \_\_\_\_\_ Prima operazioni di forward engineering e poi di refactoring

☐ \_\_\_\_\_ Prima operazioni di object model transformation e poi di forward engineering

**V** ☒ \_\_\_\_\_ Prima operazioni di refactoring e poi di reverse engineering

115. Se eredito un contratto in UML quale di queste affermazioni **non è corretta** ?

**V** ☒ \_\_\_\_\_ La preconditione del metodo nella sottoclasse consente al metodo di gestire meno casi del

corrispondente metodo nella superclasse

☐ \_\_\_\_\_ Il metodo nella sottoclasse deve assicurare la stessa postcondizione del corrispondente

metodo nella superclasse

☐ \_\_\_\_\_ L'invariante della sottoclasse può essere più restrittivo dell'invariante della superclasse

116. Quali di questi operazioni non è opportuno implementare durante la realizzazione di contratti ?

☐ \_\_\_\_\_ Controllare che la preconditione all'inizio di un metodo sia soddisfatta

**V** ☐ \_\_\_\_\_ Controllare che la postcondizione all'uscita di un metodo sia soddisfatta

☐ \_\_\_\_\_ Incapsulare il codice di controllo in metodi separati ai fini del riuso

117. L'evento percepito dall'utente come differenza tra comportamento atteso di un sistema software e comportamento esibito dal sistema software viene denominato

**V** ☐ \_\_\_\_\_ Failure

☐ \_\_\_\_\_ Fault

☐ \_\_\_\_\_ Error

118. Nel testing di integrazione bottom-up ho bisogno di realizzare

**V** ☐ \_\_\_\_\_ Test Driver

☐ \_\_\_\_\_ Test Stub

☐ \_\_\_\_\_ Sia test driver che test stub

119. Quale di queste affermazioni è falsa ?

☐ \_\_\_\_\_ Nel testing white box la derivazione dei casi di test mira a coprire la maggior parte di una classe di elementi della struttura del codice

☐ \_\_\_\_\_ Nel testing white box il numero di casi di test da eseguire dipende dal criterio di copertura adottato

**V** ☐ \_\_\_\_\_ Nel testing white-box non si controlla se l'output prodotto in corrispondenza di un input è uguale all'output atteso

120. Quando si effettua il testing di regressione ?

☐ \_\_\_\_\_ Dopo aver effettuato il test di integrazione e prima di effettuare il test di sistema

☐ \_\_\_\_\_ Dopo aver effettuato il testing di sistema e prima di effettuare il testing di accettazione

1. **Che cosa è l'ingegneria del software secondo Bruegge, si può definire mediante 4 attività fondamentali. Elencarne almeno 3.**

L'ingegneria del software è una collezione di attività, metodologie e strumenti che aiutano nella produzione del software nei vincoli di tempo, costo e qualità imposti mentre occorre il cambiamento. Le attività fondamentali sono: modellazione, problem solving, acquisizione della conoscenza, fondamenti logici come guida.

2. **le seguenti categorie di requisiti estendono il modello FURPS di base:**  
interfaccia, packaging, legali, operazione
3. **un work product è:**  
un artefatto prodotto durante l'attività di sviluppo
4. **in un diagramma di stato (state chart) un'attività è:**  
un comportamento che è eseguito quando un oggetto risiede in uno stato
5. **un caso d'uso specifica**  
tutti i possibili scenari per una data funzionalità del sistema

6. **indicare i tre tipi di comunicazione pianificata e esprimere brevemente lo scopo di ognuna:**

presentazione del problema: viene spiegato il problem statement riesame tra pari: vi sono due tipi: informale (walkthrough) dove il codice viene sottoposto al giudizio di altri sviluppatori e formale (ispezione) dove il codice viene sottoposto ad una serie di criteri. Brainstorming: si cercano tutte le soluzioni possibili ad un problema, anche quelle sbagliate perché potrebbero generare nuove soluzioni.

7. **Indicare cosa rappresentano i seguenti oggetti:**

entity object: rappresenta l'informazione persistente all'interno del sistema boundary object: rappresenta l'oggetto di comunicazione tra l'utente e il sistema control object: rappresenta l'oggetto che si occupa del controllo del flusso degli eventi (la logica del sistema)

8. **Le schede CRC servono per definire:**

struttura e relazioni delle classi del sistema

9. **L'attività che identifica concetti più specifici a partire da concetti di più alto livello viene detta:**  
specializzazione

1. **Fornire la definizione di coesione ed accoppiamento**

L'accoppiamento è l'insieme delle dipendenze tra sottosistemi. I sistemi possono essere debolmente o fortemente accoppiati.

La coesione è l'insieme delle dipendenze interne ai sottosistemi, in particolare tra le classi.

L'accoppiamento deve essere debole, mentre la coesione alta.

2. **Fanno parte della requirement elicitation le seguenti attività:**

identificare requisiti non funzionali, identificare gli scenari

3. **Quale tra le seguenti affermazioni è falsa? Un modello è utile per:**

considerare i dettagli del sistema

**4. La specifica delle seguenti informazioni “nome task, ruolo assegnato, descrizione task, input, output” qualifica:**

un task

**5. la domanda “quanti task può eseguire il sistema in un certo periodo di tempo?” fa riferimento al criterio di disegno di:**

throughput

**6. fanno parte della fase di “start” del processo di project management le seguenti attività:**

team formation, project kickoff

**7. fanno parte del “client sign-off”:**

schedule,RAD,un processo di revisione

**8. un test stub:**

simula un componente chiamato da un altro sottotest

**9. una versione che viene resa disponibile a altri sviluppatori durante un progetto viene detta:**

promotion

**10. elencare le attività del system design**

decomposizione del sistema in sottosistemi

mapping hw/sw

polizza di controllo degli accessi

scelta dell'architettura software

flusso di controllo globale

gestione dei dati persistenti

gestione delle condizioni limite

**1. Definire la nozione di “work product”**

Un work product è un artefatto prodotto durante la realizzazione di un task o di un'attività.

Il work product può essere interno (promotion) se è rivolto agli sviluppatori, deliverable se è da consegnare all'utente.

**2. Durante l'analisi gli sviluppatori mirano a produrre un modello del sistema che è:**

completo, consistente,non ambiguo

**3. Elencare i nomi delle sezioni che compongono un caso d'uso in forma testuale**

Nome del caso d'uso

Attori partecipanti

Condizioni d'entrata

Flusso di eventi

Condizioni di uscita

Requisiti di qualità

**4. Il processo teso a ricercare un certo numero di soluzioni a un problema per poi valutarle viene detto:**

brainstorming

**5. Estendono il modello FURPS al modello FURPS+:**

requisiti di implementazione, requisiti legali

**6. Un diagramma di sequenza contiene le stesse informazioni di un:**

diagramma di attività

**7. Elencare almeno cinque attività che vengono svolte durante l'analisi dei requisiti**

Identificare oggetti boundary, entity, control, attributi, aggregazioni, associazioni

**8. Nei diagrammi di attività le swimlane servono a:**

raggruppare le azioni svolte dai ruoli

**9. Possono estendere i diagrammi già presenti in UML:**

stereotipi, vincoli

**1. L'acquisizione della conoscenza è:**

un processo non lineare

**2. I diagrammi delle classi rappresentano:**

la struttura del sistema

**3. Tra una classe e una sua istanza esiste una relazione di:**

generalizzazione

**4. Uno schedule costituisce:**

il mapping di un task sull'asse dei tempi

**5. Quale tra i seguenti requisiti è verificabile?**

Il sistema fornirà risposte in meno di un secondo

**6. In UML un'associazione tra due classi è:**

la rappresentazione di un insieme di collegamenti tra oggetti

**7. Una richiesta di chiarimenti costituisce un evento di comunicazione:**

sincrono e non pianificato

**8. In UML un diagramma state chart è una notazione per:**

descrivere la sequenza di stati di un oggetto

**9. Nei diagrammi di sequenza quale delle seguenti euristiche è inadatta?**

Oggetti entità accedono a oggetti di controllo

**10. Lo scopo dell'analisi è quello di:**

strutturare e formalizzare i requisiti utente

**1. L'identificazione degli obiettivi di design (design goal)**

Permette di stabilire le necessità hw/sw

**2. Una classe C1 nel package P1 usa una classe C2 nel package P2. La relazione tra P1 e P2 è:**

di dipendenza

**3. Nei diagrammi di stato un'azione è definita come:**

un comportamento atomico eseguito nella macchina a stati

**4. La comunicazione pianificata include:**

status meeting, peer review

**5. Definire la nozione di metodo**

Un metodo è un procedimento generale per risolvere uno specifico problema.

**6. In java i contratti per la specifica delle interfacce possono essere implementati con il meccanismo di:**

meccanismo delle eccezioni

**7. Il pilot testing**

È effettuato da un selezionato numero di utenti

**8. Quali di queste attività fanno tutte parte dell'ispezione dei componenti?**

Overview, preparation, follow-up

**9. Nel contesto del configuration management, una versione di un configuration item che è stata formalmente riesaminata e poi approvata che può cambiare solo mediante una change request viene detta:**

baseline

**10. Definire il concetto di baseline nel contesto del configuration management**

È una versione di un configuration item che è stata formalmente riesaminata e successivamente approvata, che può essere cambiata solo dopo una richiesta di cambiamento.

**1. Un work product è un artefatto prodotto durante lo sviluppo software e si distingue in "interno"**

**destinato a usi propri del team di sviluppo e “deliverable” da consegnare invece al cliente.**

**Associare a ogni artefatto il tipo appropriato:**

- a. manuale d'uso: deliverable
- b. manuale dei test: interno
- c. diagramma delle classi: interno
- d. specifiche software: deliverable

**2. fornire la definizione di metodo**

un metodo è un procedimento generale per risolvere uno specifico problema.

**3. il dominio applicativo riguarda tutti gli aspetti del problema dell'utente. Il dominio applicativo include:**

ambiente fisico, utenti, processi

**4. i diagrammi dei casi d'uso possono includere i seguenti tipi di relazione:**

comunicazione, estensione, generalizzazione

**5. un task è definito come l'unità di lavoro assegnabile a:**

un ruolo

**6. lo scopo del project review è fornire informazioni:**

al project manager per stimare stato del progetto; ai team per esaminare interfacce

**7. l'ingegneria delle interfacce riguarda:**

il ridisegno di interfacce utente in un sistema pre-esistente

**8. elencare i tre modelli che compongono il modello d'analisi**

modello funzionale, modello a oggetti, modello dinamico

**9. fanno parte del “client sign-off”:**

schedule, RAD, un processo di revisione

**10. quando è possibile progettare test ripetibili per dimostrare che il sistema soddisfa la specifica, si dice che:**

la specifica dei requisiti è verificabile

**1. “system”, “model”, “document” sono sottoclassi dirette della classe:**

work product

**2. Un “ruolo” è definito come:**

un insieme di task di tipo manageriale e tecnico che sono assegnati ad un singolo o ad un team.

**4. Il dominio delle soluzioni rappresenta:**

lo spazio di modellazione di tutti i possibili sistemi.

**5. In una diagramma delle classi un'associazione molti a molti rappresenta:**

una molteplicità 0..n oppure 1..n su entrambi i lati.

**6. Una specifica dei requisiti che non contraddice se stessa viene detta:**

consistente

**7. Descrivere (scopi e partecipanti) la comunicazione pianificata "problem inspection"**

L'ispezione è un tipo di comunicazione pianificata detta "riesame tra pari" che avviene tra sviluppatori. Durante questo tipo di comunicazione viene sottoposto il codice sorgente ad una serie di criteri per verificare se li rispetta.

**1. sia W l'insieme dei work product e D l'insieme dei deliverable di un progetto software. Qual è la relazione tra W e D?**

deliverable è sottoinsieme di work product

**2. quale tra le seguenti affermazioni è falsa? Un modello è utile per:**

studiare la proprietà emergente del sistema

**3. un tipo di dato astratto è specificato in:**

un linguaggio a oggetti

**4. i diagrammi dei casi d'uso descrivono il comportamento di un sistema come percepito da:**

utenti del sistema

**5. in UML un link rappresenta:**

una connessione tra oggetti

**6. la specifica delle seguenti informazioni "nome task, ruolo assegnato, descrizione task, input, output" qualifica:**

un task

**7. il V-Model si occupa della pianificazione di attività di testing ed esecuzione dei test case relativamente a:**

tutte le fasi del ciclo di sviluppo

**8. sono esclusivamente di pianificazione i seguenti insiemi di documenti:**

test design, test case

**9. le seguenti attività gestionali "configuration item identification, release management, variant management" fanno parte del:**

change management

**10. quale tra le seguenti caratteristiche è estranea alla definizione di progetto?**



Serve a creare un prodotto o un servizio

**1. i seguenti elementi fanno parte della specifica di un work package:**

nome task, descrizione task, dipendenze sugli input

**2. un vincolo è una regola collegata a un elemento di modellazione UML che:**

restringe la semantica dell'elemento modellato

**3. gli scopi del client review:**

confermare i cambi ai requisiti del sistema

**4. quale tra le seguenti attività non fa parte della requirement elicitation?**

Identificare i dati persistenti

**5. una baseline è definita come una versione di un work product che è:**

una versione di work item formalmente riesaminata e approvata

**6. il testing di un insieme comune di funzionalità presso un ristretto e selezionato gruppo di utenti viene detto:**

pilot testing

**7. nell'ambito della specifica delle interfacce un contratto è interpretabile come un accordo tra:**

l'utente della classe e l'implementatore della classe

**8. quale delle seguenti tecniche è rivolta esclusivamente a scoprire staticamente gli errori (senza l'esecuzione del programma o dei modelli che lo rappresentano)?**

Fault avoidance

**9. un processo aziendale di change management può essere modellato mediante:**

activity diagram

**10. quale tra le seguenti attività è estranea al project management?**

Gestione qualità del software

**1. fornire una definizione di metodologia**

una metodologia è un insieme di metodi che servono per risolvere classi di problemi

**2. quale domanda non è attinente al requisito di supportability?**

Quanti utenti concorrenti deve supportare il sistema?

**3. il modello dell'analisi deve essere**

completo, consistente, non ambiguo

**4. qual è lo scopo dei diagrammi di deployment?**

Mostrare le relazioni fisiche tra i componenti hw e sw

5. un constraint è una regola associata a un elemento di modellazione UML che:

6. ne restringe la semantica

7. la politica delle protezioni in un sistema sw può essere definita tramite la matrice degli accessi.

Elencare i tre possibili approcci per la rappresentazione della matrice degli accessi riportandone le caratteristiche essenziali.

La matrice si può rappresentare attraverso tre approcci:

tabella d'accesso globale: rappresenta esplicitamente ogni cella nella matrice come una tripla (attore, classe, operazione). Se tale tripla non è presente l'accesso è negato.

Lista di controllo degli accessi: associa una coppia (attore, operazione) a ciascuna classe.

Capability: associa una coppia (classe, operazione) a ciascun attore.

8. Gli strumenti software definite "same time, different place groupware" consentono di:

collaborare tra utenti in rete in modalità sincrona

9. il documento che contiene : inputs, drivers, stubs, e risultati attesi dei test viene chiamato:

test case specification

10. giustificare l'uso di uno strumento software per la gestione delle configurazioni

1. fornire una definizione di work package

è la specifica di come deve essere realizzato il work product. Al suo interno sono indicati nome del task, input, output, schedule.

2. quale domanda tra le seguenti è inadatta a individuare gli attori di un sistema?

Chi paga per la realizzazione del sistema?

3. Un diagramma state chart è una notazione per:

descrivere la sequenza di stati di un oggetto

4. Quali concetti non sono propri della fase di system design?

Generalizzazione e specializzazione

5. La "differenza tra quanto specificato e comportamento osservato" fa riferimento al criterio di: affidabilità

6. Il testing funzionale, di usabilità e di prestazioni eseguito dal cliente nell'ambiente di sviluppo viene definito:

acceptance testing

7. La comunicazione pianificata include:

problem presentation, project review, peer review

8. Una versione di un configuration item che è stata formalmente riesaminata e che può essere

cambiata solamente tramite una richiesta di cambiamento viene chiamata:

baseline

#### 9. Giustificare l'uso del linguaggio OCL

L'OCL è un linguaggio che consente l'uso dei vincoli per specificare formalmente gli elementi in un singolo modello. Un vincolo è espresso con un'espressione booleana che restituisce true/false.

#### 1. Fornire una definizione di modello e giustificarne l'uso in ingegneria del software

Un modello è un'astrazione della realtà. Un'astrazione è utile perché si focalizza sui dettagli rilevanti ignorando quelli non rilevanti. Un modello è utilizzato quando abbiamo a che fare sistemi troppo piccoli, troppo grandi, troppo complessi o anche troppo costosi poiché un modello è una rappresentazione economica della realtà. In ingegneria del software usiamo i modelli per dominare la complessità.

#### 2. Un dominio applicativo:

rappresenta tutti gli aspetti del problema utente

#### 3. Un link rappresenta:

una connessione tra oggetti

#### 4. Uno schedule costituisce:

il mapping di un task sull'asse dei tempi

#### 5. Nei diagrammi statechart uno stato rappresenta:

una condizione soddisfatta da un insieme di attributi di un oggetto

#### 6. In UML un'associazione tra due classi è:

la rappresentazione di un insieme di collegamenti tra oggetti

#### 7. La comunicazione non pianificata include:

request for change, request for clarification, issue resolution

#### 8. Il modello FURPS è stato esteso includendo i seguenti requisiti non funzionali:

operation requirements, legal requirements

#### 9. La seguente domanda "i casi d'uso sono stati tutti nominati usando frasi verbali?" fa riferimento a:

non ambiguità della specifica

#### 10. Giustificare l'uso delle schede CRC durante l'analisi dei requisiti

Le schede CRC si usano per individuare le classi, le responsabilità e le collaborazioni tra le classi.

Vengono usate perché sono uno strumento facile da utilizzare e modellare.

11. Riprodurre l'indice del RAD commentando brevemente le finalità di ogni paragrafo

- Introduzione: introduce il sistema
- Requisiti funzionali: si indicano le funzionalità che deve supportare il sistema
- Requisiti non funzionali: si indicano i vincoli che non sono relati alle funzioni del sistema
- Modelli del sistema
  - Scenari: istanze di caso d'uso
  - Casi d'uso: rappresentazione funzionale del sistema. Descrive il sistema come flusso di eventi.
  - Modello a oggetti: descrizione statica del sistema. Comprende diagramma delle classi.
  - Modello dinamico: descrizione dinamica del sistema. Comprende i diagrammi di attività, i sequence diagram e gli state-chart diagram.
  - User interface: comprende i navigational path e i mock-up
- Glossario: definizione dei termini utilizzati all'interno del documento. Serve per evitare le incomprensioni e le ambiguità.

1. Giustificare la seguente affermazione: "l'acquisizione della conoscenza è un processo non lineare"

Con l'aggiunta di un singolo dato si può invalidare l'intero modello

2. Nei diagrammi state chart un'azione esprime sempre:

un comportamento atomico eseguito in punti specifici nella macchina a stati

3. I tre tipi principali di interazione tra i partecipanti a un progetto sono:

reporting, decision, communication

4. quale attività non fa parte delle attività di requirements elicitation?

identificare gli oggetti boundary

5. qual è l'esempio corretto di bilanciamento (trade-off) tra obiettivi di progetto?

Tempo di consegna vs qualità

6. l'attività di specifica delle interfacce include:

specificare i contratti, identificare le boundary conditions

7. fornire una definizione di test case

è un insieme di input e risultati aspettati che esercita un componente di test con lo scopo di causare fallimenti e rilevare errori.

8. Il path testing costituisce una tecnica di:

whitebox testing

9. I cambi a un configuration item associati con una singola revisione di una configurazione vengono chiamati:

change set

1. Il lavoro di un team cross-funzionale:

ha impatto sullo sviluppo di molti sottosistemi

2. Quali delle seguenti affermazioni è falsa. Il riesame del cliente (client review) è usato per:  
ispezionare il codice sorgente

3. Quale di queste affermazioni è falsa? Un project review è condotto  
Durante il test di accettazione del cliente

4. Quale tra i seguenti concetti non è collegato al pattern Facade?  
Pattern proxy

5. Le boundary condition:  
definiscono come il sistema viene inizializzato

6. Il re factoring:  
migliora la leggibilità del codice sorgente

7. cosa contiene il test report summary:  
la lista di tutte le failure riscontrate nel testing

8. l'interfaccia di un sistema include:  
nome operazioni, tipi input e output

9. una capability list risponde alla domanda:  
a quali oggetti questo attore può accedere

10. quali dei seguenti concetti fanno parte del Configuration Management:  
version, repository, software library

1. in UML uno stato è:  
una condizione soddisfatta dagli attributi di un oggetto

2. una specifica software si dice realistica quando:  
il sistema software può essere implementato entro i vincoli

3. in UML un'associazione tra due classi è:  
la rappresentazione di un insieme di collegamenti tra oggetti

4. uno stereotipo è un:

meccanismo di estensione per classificare elementi UML

5. descrivere l'attività di acquisizione della conoscenza in ingegneria del software

(requirement elicitation) è la prima attività che si realizza durante lo sviluppo software.

Vengono raccolti tutti i dati relativi al problema e successivamente vengono formalizzati in vere e proprie informazioni. L'acquisizione della conoscenza è un'attività non lineare in quanto un'introduzione di una nuova conoscenza può invalidare il modello.

6. La domanda "quale documentazione dovrebbe essere messa a disposizione dell'utente?" fa parte dei requisiti di:

usabilità

7. Quali tra le seguenti specifiche non vengono definite durante la fase di scomposizione di un sistema software?

Casi d'uso limite, obiettivi del progetto

8. Elencare le attività della fase "specificazione delle interfacce":

specificare precondizioni, post condizioni, invarianti

indicare i metodi: firma, tipo di parametri e tipo di ritorno

9. Un diagramma PERT è?

Uno schedule rappresentato come grafo aciclico

1. Quale tra i seguenti è un requisito funzionale?

Il sistema dovrà rilasciare titoli di viaggio

2. in UML una vista è:

un sottoinsieme del modello del sistema

3. quale dei seguenti aspetti non attiene alla validazione dei requisiti?

Decomponibilità

4. un oggetto partecipante è un oggetto introdotto da:

diagramma di sequenza

5. quale tra i seguenti tipi di work product non è interno?

Modello funzionale

6. in UML le classi di analisi rappresentano:

un'astrazione ben definita del dominio di applicazione

7. in un diagramma delle attività una ricongiunzione (join) sincronizza due o più flussi

concorrenti quando:

ha molte transizioni in ingresso e una in uscita

8. i diagrammi di collaborazione rappresentano la sequenza di messaggi:

dall'alto verso il basso

9. quali tra le seguenti domande è estranea alla determinazione del requisito non funzionale di affidabilità?

Quanti utenti concorrenti dovrà supportare il sistema?

1. Quale tra i seguenti è un requisito non funzionale?

Il sistema dovrà essere facile da utilizzare

2. In quale dei seguenti casi è inopportuno usare il procedimento di falsificazione?

Durante la raccolta dei requisiti

3. Quale delle seguenti affermazioni è sbagliata? La struttura di UML è costituita da:

linguaggi orientati agli oggetti

4. Quale domanda, tra le seguenti, è inadatta a individuare gli attori di un sistema?

Chi paga per la realizzazione di un sistema?

5. Quale tra i seguenti requisiti è verificabile?

Il sistema fornirà risposte in meno di un secondo

6. In UML un'associazione tra due classi è:

la rappresentazione di un insieme di collegamenti tra oggetti

7. Nei diagrammi delle classi ogni classe può essere composta da:

nome, attributi, operazioni

8. Un diagramma state chart è una notazione per:

descrivere la sequenza di stati di un oggetto

9. Nei diagrammi di sequenza quale delle seguenti euristiche è inadatta?

Oggetti entità accedono a oggetti di controllo

10. Quali dei seguenti elenchi di concetti vengono tutti gestiti dal project manager per organizzare un progetto?

Ruoli, work product, tasks, schedule

1. Quale delle seguenti affermazioni è corretta:

- Il sequence diagram è adatto alla comunicazione con il cliente

- Il sequence diagram è più intuitivo degli use case
- **V** Il sequence diagram fornisce una prospettiva diversa che consente di individuare oggetti mancanti e aree non chiare nelle specifiche

2. (Sequence Diagram) quali di queste affermazioni non sono corrette:

- **V** Gli oggetti sono attivati quando vengono istanziati
- Una classe stabilisce i messaggi a cui un oggetto può rispondere
- **V** I messaggi sono inviati da oggetti

3. Un class diagram è tipicamente usato per modellare:

- Le relazioni tra oggetti dei sequence diagram individuati
- **V** Il glossario di un sistema: sono prese decisioni relativamente alle astrazioni da considerare
- **V** Lo schema concettuale di un database

4. Quale/i delle seguenti affermazioni è/sono corretta/e? In una architettura client/server le funzioni eseguite dal server di database sono:

- Gestione decentralizzata dei dati
- **V** Garantire la sicurezza del Database
- **V** Gestire la concorrenza delle operazioni (multiple user access)
- Elaborazioni decentralizzate

5. Un Boundary use case descrive (indicare la/e risposte corrette);

- **V** La configurazione del sistema
- **V** Le scelte relative allo startup, allo shutdown
- Le condizioni limite

6. In che modo possono essere specificati i contratti in UML?

- In Javadoc
- **V** La Object Contract Language

7. Se devo effettuare una trasformazione prima sul modello a oggetti e poi ottengo il codice, faccio:

- Prima operazioni di forward engineering e poi di refactoring
- **V** Prima operazioni di object model transformation e poi di forward engineering
- Prima operazioni di refactoring e poi di reverse engineering



8. Durante l'ispezione ai fini del fault detection (indicare le risposte corrette):
- Lo sviluppatore presenta gli artefatti
  - Si controlla l'efficienza degli algoritmi con le richieste funzionali
  - **V** Lo sviluppatore interviene solo se si richiedono chiarimenti
9. Il testing mira a valutare l'aderenza del sistema ai requisiti non funzionali viene denominato:
- Integration testing
  - Acceptation testing
  - **V** Performance testing
10. Quale di queste affermazioni è vera:
- Il testing white-box consente di capire se non ho implementato delle funzionalità
  - Il testing black-box consente di individuare caratteristiche non richieste
  - **V** Il testing black-box consente di capire se non ho implementato delle funzionalità
11. Quale delle seguenti affermazioni è/sono corretta/e?
- Un programma è esercitato da un caso di test (insieme di dati di input)
  - Un test è formato da un caso di test
  - **V** L'esecuzione del test consiste nell'esecuzione del programma per tutti i casi di test
  - Un test ha successo se non rileva malfunzionamenti del programma
12. Quale di queste affermazioni è vera?
- Con il top down integration testing ho bisogno di driver
  - **V** Con il bottom up integration testing ho bisogno di driver
  - Con il sandwich testing non ho bisogno di driver

**Quale di queste affermazioni relativa ad un package in UML è falsa ?**

- Consente di decomporre un sistema in sottosistemi
- Consente di raggruppare classi Java
- E' un meccanismo per strutturare e migliorare la leggibilità dei modelli

**Gli stati di un diagramma di stati UML sono rappresentati con:**

- Rettangoli
- Ovali
- Rettangoli con angoli arrotondati

**Una entità fuori dal sistema che si sta modellando è**

- Una classe
- Una istanza
- Un attore

**Cosa si intende per backward traceability dei requisiti ?**

- Il sistema garantisce la compatibilità con il sistema precedente
- Dato un requisito funzionale è possibile individuare le funzioni del sistema che lo realizzano
- Per ogni funzione del sistema è possibile individuare un insieme di requisiti funzionali

**Quale di queste affermazioni non si riferisce ad un'attività in un diagramma di stato ?**

- Operazione che prende tempo per essere completata
- E' associata ad un evento
- E' associata ad uno stato

**A cosa serve un qualifier ?**

- A specificare la molteplicità di una associazione
- A ridurre la molteplicità di una associazione
- A identificare una estremità di una associazione

**Cosa è un Lower-CASE ?**

- Uno strumento per la generazione automatica di interfacce utente
- Uno strumento che supporta le fasi di programmazione, testing e debugging
- Uno strumento che supporta le attività delle fasi di analisi dei requisiti e di progettazione

**Quale di queste affermazioni relativa ad un package in UML è falsa ?**

- Consente di decomporre un sistema in sottosistemi
- Consente di raggruppare classi Java
- E' un meccanismo per strutturare e migliorare la leggibilità dei modelli

**Come viene rappresentato un invio condizionato di un messaggio in un diagramma di sequenza UML ?**

- Una condizione racchiusa tra parentesi angolari precede il nome del messaggio
- Una condizione racchiusa tra parentesi quadre precede il nome del messaggio
- Un asterisco precede il nome del messaggio

**Quale di queste affermazioni non è valida per una struttura organizzativa di reporting gerarchica ?**

- Lo stato è riportato dall'alto verso il basso

- Le decisioni vengono comunicate dall'alto verso il basso
- Le informazioni relative a stato e decisioni sono unidirezionali

### **Cosa si intende per forward traceability dei requisiti ?**

- Il sistema garantisce la compatibilità con il sistema precedente
- Dato un requisito funzionale è possibile individuare le funzioni del sistema che lo realizzano
- Per ogni funzione del sistema è possibile individuare un insieme di requisiti funzionali

### **Quale di queste affermazioni non si riferisce ad un diagramma di sequenza ?**

- Serve ad individuare i cambiamenti degli oggetti nel tempo
- Serve ad individuare le relazioni temporali tra oggetti nel tempo
- Serve ad individuare la sequenza di operazioni come risposta ad uno o più eventi

### **Cosa non va individuato in fase di analisi dei requisiti ?**

- Le classi del dominio applicativo
- Le classi del dominio delle soluzioni
- Le operazioni delle classi

### **Se devo realizzare un'associazione qualificata, quale struttura dati Java utilizzo?**

- Una mappa (Map)
- Un insieme (Set)
- Una lista (List)

### **Se voglio realizzare l'associazione uno-a-molti "ha-superato" tra le classi Studente ed Esame (uno studente può aver superato più esami), quale mapping è più adatto?**

- Una Collection sulla classe Esame in cui memorizzare gli studenti che hanno superato l'esame ed un riferimento sulla classe Studente in cui memorizzare l'esame superato
- Una Collection sulla classe Studente in cui memorizzare gli esami superati ed un riferimento sulla classe Esame in cui memorizzare lo studente che ha superato l'esame
- Una Collection sulla classe Esame in cui memorizzare gli studenti che hanno superato l'esame ed una Collection sulla classe Studente in cui memorizzare gli esami superati

**Se voglio realizzare l'associazione multi-a-molti "ha-superato" tra le classi Studente ed Esame (uno studente può aver superato più esami e ciascun esame può essere stato superato da più studenti), quale mapping è più adatto?**

-Una Collection sulla classe Esame in cui memorizzare gli studenti che hanno superato l'esame ed un riferimento sulla classe Studente in cui memorizzare l'esame superato

-Una Collection sulla classe Studente in cui memorizzare gli esami superati ed un riferimento sulla classe Esame in cui memorizzare lo studente che ha superato l'esame

-Una Collection sulla classe Esame in cui memorizzare gli studenti che hanno superato l'esame ed una Collection sulla classe Studente in cui memorizzare gli esami superati

-----FILE GIANLUCA-----

**Quale di queste affermazioni non è appropriata per il modello di processo a spirale?**

-Per ogni sistema è possibile utilizzare diversi modelli di processo

-Non è adatto per la manutenzione del software

-è difficile definire il contratto

**Quale di queste affermazioni non è vera?**

-I prototipi possono essere utili per ridurre i rischi del progetto

-Il modello a cascata prende in considerazione la gestione sistematica dei rischi del progetto

-Il rischio è legato alla qualità e alla quantità di informazioni possedute

**Quale di queste affermazioni non è vera?**

-Il cammino critico è formato da attività che sono particolarmente complesse

-Il ritardo di un attività sul cammino critico ritarda l'intero progetto

-Ogni attività sul cammino critico ha uguale minimo tempo di completamento e massimo tempo di completamento

**Quale di queste affermazioni non è vera?**

-Nel forwarding engineering si parte dal codice per derivare i modelli del software

-nel greenfield engineering si parte da zero non esiste un sistema precedente

-Il software re-engineering è realizzato per ridurre i costi di manutenzione

**Quale di queste affermazioni è vera?**

-Un deliverable è la descrizione di un task

-Un wordpackage è la descrizione di un milestone

-Una attività può essere scomposta in task che possono essere svolti in parallelo

-Un task dura per tutta la durata del progetto

**Quale di queste affermazioni non è vera?**

-Il modello a cascata consente di ridurre il rischio di overrun

-Il modello a cascata non ha una buona visibilità

-Nel modello a cascata le fasi del processo sono in progressione sequenziale

**Quale di queste affermazioni è vera?**

-Lo sviluppo esplorativo è adatto per sistemi interattivi a vita lunga

-Il processo di sviluppo esplorativo non consente di ridurre il rischio legato alla cattiva comprensione dei requisiti

-nello sviluppo esplorativo vengono costruite varie versioni del sistema sottoposte a validazione da parte dell'utente

**Quale di queste affermazioni non è vera?**

-La coesione misura le dipendenze tra i sottosistemi di un sistema

-In sottosistemi con elevata coesione le modifiche ad un sottosistema hanno forte impatto sugli altri sottosistemi

-Le classi di un sottosistema con elevata coesione eseguono task simili/strettamente collegati

**Dall'informazione della matrice di tracciabilità in cui ho collegato i requisiti ai relativi test case**

-posso ricavare quali requisiti sono stati testati

-posso ricavare quali requisiti non sono stati realizzati

-posso determinare se ho realizzato un testing white-box o black-box

**Quale di queste affermazioni è vera?**

-in un'architettura a repository il client rappresenta il collo di bottiglia

-in un'architettura a repository è possibile aggiungere un numero qualsiasi di client se il repository conosce l'interfaccia del client

-un'architettura a repository consente una gestione centralizzata dei dati

**Quale di queste affermazioni è vera?**

-il ruolo del project manager in SCRUM è proposto a facilitare e proteggere il team

-in SCRUM i requisiti non possono cambiare durante lo sprint

-in SCRUM il focus factor viene utilizzato per definire la team velocity

**(use case diagram) Quale di queste affermazioni non è vera?**

-è possibile definire una relazione di estensione tra attori

-una relazione di estensione tra casi d'uso consente di esprimere casi d'uso per la gestione delle boundary condition

-è possibile definire una relazione di estensione tra attore e caso d'uso

**Quale tipo di prototipazione parte con i requisiti meglio compresi ?**

-Prototipazione esplorativa

-Throw-away

-Mock-ups

**Cosa non va individuato in fase di analisi dei requisiti ?**

-Le classi del dominio applicativo

-Le classi del dominio delle applicazioni

-Le operazioni delle classi

**Qual è l'obiettivo della prototipazione della prototipazione throw-away?**

-Costruire un sistema in modo rapido

-Ridurre i costi di sviluppo e manutenzione

-Avere una chiara comprensione dei requisiti

**Che tipo di relazione può esistere tra due attori in un diagramma dei casi d'uso?**

-Associazione

-Dipendenza

-Generalizzazione

**Cosa si intende per consistenza dei requisiti?**

-i requisiti rappresentano la vista dell'utente

-sono descritti tutti i possibili scenari del sistema

-non ci sono requisiti che si contraddicono

**Cosa esprime una relazione di estensione tra casi d'uso?**

-una variante del normale flusso di eventi in un caso d'uso

-una specializzazione di un caso d'uso

-una decomposizione funzionale di un caso d'uso

**Quando si effettua il testing di integrazione?**

-Dopo aver effettuato il testing di sistema e prima di effettuare il testing di accettazione

-Dopo aver effettuato delle modifiche al codice per correggere un fault

-Dopo aver effettuato il test di unità e prima di effettuare il test di sistema

**Il testing che mira a valutare l'aderenza del sistema ai requisiti non funzionali viene denominato?**

- Integration testing
- Acceptance testing
- Performance testing

**Quale di queste affermazioni è vera?**

- Con il top down integration testing ho bisogno di driver
- con il bottom up integration testing ho bisogno di driver
- con il sandwich testing non ho bisogno di driver

**L'evento percepito dall'utente come differenza tra comportamento atteso di un sistema software e comportamento esibito dal sistema software viene denominato?**

- Failure
- Fault
- Error