

Softversko inženjerstvo – popis pitanja za usmeni

I. Uvod u softversko inženjerstvo

1. Što je to softverski produkt i koji su važniji atributi njegove kvalitete?
2. Što je softverski proces i što je softversko inženjerstvo?
3. Model vodopada, prednosti i mane.
4. Model evolucijskog razvoja, prednosti i mane
5. Model formalnog razvoja, prednosti i mane
6. Model usmjeren na ponovnu upotrebu, prednosti i mane
7. Model inkrementalnog razvoja, prednosti i mane
8. Upravljanje softverskim projektom i njegove osobine, poslovi softverskog menadžera.

II. Zahtjevi i specifikacija

9. Što su zahtjevi, kakve vrste postoje i kakvi trebaju biti?
10. Pod-aktivnosti unutar specifikacije i kako se one provode.
11. Koji dokumenti o zahtjevima postoje, tko ih čita?
12. Problemi i savjeti kod pisanja specifikacija.
13. Što je model sustava i kada nastaje? Navesti važnije perspektive kod modeliranja.
14. Opisati model toka podataka i dati primjer.
15. Opisati model entiteta, veza i atributa, te dati primjer.
16. Opisati model promjene stanja, dati primjer.
17. Opisati objektne modele, nabrojiti vrste, dati primjer jednog od njih.
18. Što je prototip, čemu služi i kakva je korist od njega?
19. Nabrojiti poznatije načine za prototipiranje i opisati svaki od njih.
20. Što je to formalna specifikacija, kako se ona u stvarnosti provodi?

III. Oblikovanje i implementacija

21. Definirati oblikovanje i implementaciju. Navesti glavne pod-aktivnosti.
22. Navesti dva pristupa oblikovanju, dati primjer za svaki od njih.
23. Kako oblikovanje utječe na kvalitetu sustava?
24. Oblikovanje arhitekture sustava i njegove pod-aktivnosti, dati primjer arhitekture nekog sustava.
25. Nabrojiti tri najpoznatija modela arhitekture sustava i ukratko opisati svaki od njih.
26. Navesti četiri uobičajena modela kontrole sustava i ukratko opisati svaki od njih.
27. Što je distribuirani sustav i koje su njegove prednosti i mane?
28. Opisati arhitekturu klijent-poslužitelj. Dati primjer takve arhitekture.
29. Opisati arhitekturu s distribuiranim objektima, navesti poznatije standarde za 'middleware'.
30. Opisati pear-to-pear arhitekturu i dati primjer.
31. Što je to objektni pristup oblikovanju i koje prednosti ima, što je objekta, a što klasa?
32. Koja svojstva ima objektno-oblikovani sustav?
33. Navesti pod-aktivnosti unutar objektno oblikovanja te ukratko opisati svaku od njih na nekom primjeru.
34. Što je to oblikovanje korisničkog sučelja i koje vrste takvog sučelja postoje?
35. Koji su principi oblikovanja korisničkog sučelja?
36. Koji su načini interakcije korisnika sa sustavom, ukratko opisati svaki od njih.
37. Na koji načine se prikazuju informacije korisniku, dati primjere.
38. Opisati vođenje korisnika kroz sustav, kako se ono realizira?
39. Oblikovanje uz ponovnu upotrebu softvera. Koji su nužni uvjeti za takav razvoj?
40. Opisati razvoj zasnovan na komponentama. Što je komponenta i kakva sučelja ima?
41. Definirati stupnjeve veličine komponente i dati primjer za svaki od njih.
42. Što su porodice aplikacija i kako nastaju novi članovi porodice?
43. Što su obrasci za oblikovanje i kako se oni zapisuju. Navesti jedan takav obrazac.

IV. Verifikacija i validacija

44. Što su verifikacija i validacija. Koja je razlika između ta dva pojma?

45. Koje su metode za verifikaciju i validaciju.
46. Gdje je mjesto verifikacije i validacije u softverskom procesu?
47. Što je plan za testiranje softvera i od čega se sastoji?
48. Što je proces 'debugiranja', čemu služi i kako se provodi?
49. Što je statička verifikacija. Nabrojiti poznatije tehnike statičke verifikacije.
50. Opisati proces inspekcije programa.
51. Opisati kako se provodi automatska statička analiza. Primjer alata za to.
52. Što je to formalna verifikacija, kako se provodi.
53. Što je testiranje, navesti vrste testiranja.
54. Kako se testiraju greške, načini testiranja dijelova softvera.
55. Opisati funkcionalno testiranje na nekom primjeru.
56. Opisati strukturalno testiranje na nekom primjeru.
57. Opisati testiranje po putovima na nekom primjeru.
58. Kako se provodi testiranje integracije? Kakve vrste grešaka tu postoje?
59. Navesti dobre savjete za testiranje integracije.
60. Koji CASE alati nam mogu pomoći kod testiranja?

V. Održavanje i evolucija

61. Što je održavanje, što je evolucija, koje strategije mijenjanja softvera postoje?
62. Što nastaje kao rezultat održavanja i koje vrste održavanja postoje?
63. Nabrojiti 'Lehmanove' zakone vezane za dinamiku održavanja softvera.
64. Koji faktori utječu na cijenu održavanja softvera?
65. Što je to upravljanje konfiguracijom i koje aktivnosti uključuje?
66. Kako se upravlja promjenama sustava, dati neki primjer.
67. Kako se identificiraju verzije sustava, koja je razlika između verzije i izdanja?
68. Kako se vrši automatska gradnja sustava i navesti neke poznatije alate za to.
69. Što je baštinjeni softver i koje osobine ima.
70. Čemu služi softversko re-inženjerstvo i koje pod-aktivnosti uključuje?
71. Koji je cilj arhitekturne transformacije i kako se ona provodi?