

Kordamisküsimused:

1. Mis on tarkvaratehnika?
2. Miks vajame tarkvaratehnikat?
3. Milleks tarkvaratehnika?
4. Tarkvaratehnika „point“?
5. Mis on tarkvara(toode)?
6. Kvaliteetse tarkvara atribuudid.
7. Tarkvaratehnika huvigrupid.
8. Tarkvaratehnika kui distsipliini eesmärgid.
9. Tarkvaratehnika kontekst.
10. Mis on süsteem?
11. Süsteemi kategooriad ja nende seletused.
12. Mis on süsteemitehnika?
13. Millised on parimad tarkvaratehnika meetodid?
14. Tarkvararakenduse liigid.
15. Mis on protsess?
16. Mis on tarkvara arendusprotsess e. tarkvaraprotsess?
17. Tarkvaraprotsessi mudel.
18. Plaanipõhine vs agiilne tarkvaraprotsess.
19. Tarkvaraprotsessi mudelite tüübid.
20. Kirjelda Kose mudeli koos puudustega ja eelistega.
21. Mis on modifitseeritud kose mudel?
22. Mis on iteratiivne arendamine? Eelised ja puudused?
23. Mis on komponendipõhine tarkvaraarendus, puudused ja eelised? Mis tüübid on olemas?
24. Mis on kõige parem tarkvaraprotsessi mudel?
25. Millist tarkvaraprotsessi mudelit kasutatakse kõige rohkem?
26. Mis on tarkvaraarenduskulud? Kuidas nad jaotuvad?
27. Tarkvarainseneri professionaalsus.
28. Inimloomus ja eetika.
29. Tarkvarainseneri professionaalse vastutuse aspektid.
30. ACM/IEEE eetikakoodid ja printsiibid ning nende seletused.
31. Eetilised dilemmad.
32. Eetiliste dilemmade põhipunktid.
33. Tarkvara nõuete koostamise tehnikad ja lähenemised. (Requirements Engineering)
34. Millest koosneb tarkvara nõuded?
35. Miks on vaja kirjutada nõuded?
36. Parimad praktikad nõuete kirjeldamiseks. Too näited.
37. Tuleta meelde, mis maailma probleemid olid käsitletud. Ja mis lahendused arvutite poolt olid pakutud. Nimeta need ja seleta lahti.
38. Nõuete esialgne definitsioon. Millised veel on nõuete definitsioonid?
39. Mis vahe nende mõistete vahel on: Süsteemi nõuded ja tarkvara nõuded?
40. Nõuete kolm mõõdikud koos seletustega.
41. Types of statements involved: descriptive vs. Prescriptive
42. Categories of requirements: functional vs. non-functional
43. A taxonomy of non-functional requirements
44. The requirements lifecycle: actors, processes, products
45. The requirement engineering process

46. Requirement engineering: an iterative process
47. Target qualities and defects to avoid
48. Types of software projects
49. Requirements in the software lifecycle
50. Relationship to other disciplines
51. The requirements problem: facts, data, citations
52. Role and stakes of requirement engineering.
53. Obstacles to good requirements engineering practice.
54. Millest koosneb tarkvarakvaliteet? Miks?
55. Mis võiks olla kvaliteet?
56. Millest koosneb tarkvaratoode?
57. Nimeta erinevate osapoolte erinevad nõuded.
58. Nimeta süsteemi kvaliteedi nõuded vastavalt EVS-ISO/IEC 25010:2011 standardile.
59. Mis on funktsionaalne nõue? Too näited.
60. Mis on mittefunktsionaalne nõue? Too näited.
61. Nimeta nõue, mis on testitav.
62. Nimeta nõue, mis on mittetestitav.
63. Mis on reaalne nõue?
64. Mis on mittereaalne nõue?
65. Milline nõue on hea nõue?
66. Kas nõue peab olema kodus testitav?
67. Mis on tarkvaraprotsessid?
68. Millistest komponentidest koosneb elutsüklimudel?
69. Kirjelda V-mudeli.
70. Mis on testimine?
71. Kui korraldada testimist, mida peale nõuetest ja protsessist peaks veel teadma?
72. Kuidas võiks jagada testimismeetodeid? Mis kaks põhilist meetodi on? Miks?
73. Kus leitud viga on kõige odavam parandada?
74. Räägi statistikast läbivaatuse kohta.
75. Valge kasti testimine.
76. Musta kasti testimine.
77. Halli kasti testimine.
78. Testimise tasandid. Nimeta need ja seleta lahti. Too näited.
79. Testimise tüübid. Nimeta ja seleta lahti. Too näited.
80. Millest koosneb testimise protsess?
81. Millal lõpetada testimist?
82. Mida teha, kui testida ei saa?
83. Kas igaüks võib olla testija?
84. Testijale abiks omadused ja oskused.
85. Valik jaotusi testimisele.
86. Testijaks olemise võlud.
87. Mis on tarkvara arhitektuur?
88. Erinevused funktsionaalsest disainist.
89. Mis on arhitektuuri erosioon.
90. Klient-server arhitektuur.
91. Komponentidel põhinev arhitektuur.
92. Kihiline arhitektuur.
93. Message bus.

94. N-tier.
95. Objekt orienteeritud arhitektuur.
96. DDD arhitektuur.
97. Teenus orienteeritud arhitektuur.
98. Monoliitne arhitektuur.
99. Mikroteenused.
100. Kuidas valida arhitektuuri?
101. Arhitektuuri disain.
102. Nimeta ja seleta lahti arhitektuuri disainimise protsessid.
103. Kes on arhitektuuri omanik?
104. Kuidas toimud töötamine arhitektuuriga suures meeskonnas?
105. Arhitektuuri testimine.
106. Agiilne arhitektuur.
107. Veebirakenduse arhitektuur.
108. Nimeta universaalsed põhimõtted ja seleta need lahti.
109. Hea arhitektuuri eelised ja nende seletused.
110. HTTP protokoll.
111. HTTP päring
112. HTTP vastus
113. Nimeta veebirakenduste liike ja seleta need lahti.
114. Veebirakendus JAVAs.
115. Servlek API
116. Model view controller
117. JSP
118. JAX-RS
119. Starter KITS
120. Alternatiivid servletile
121. Mineta tööriistu ja raamistike veebirakenduse jaoks.
122. Riskide võrdlus
123. Nimeta manifesto agiilse tarkvaraarendamiseks ja seleta need lahti.
124. Agiilse meetodi printsiibid. Seleta lahti.
125. Agiilsete metodoloogiate maastik.
126. eXterme Programming.
127. XP väärtused.
128. Scrum? Kõik, mis oli loengus. Seletada lahti kõik mõisted, tegevused ja osalejad.
129. Mis on kasutuslugu?
130. Kanban.
131. Minimal Marketable Feature (MMF)
132. Eesmärgi mudel
133. Kandan vs Scrum
134. Mis on mudel? Kus kasutatakse? Milleks on mudeleid vaja?
135. Tarkvaratehnika vaated.
136. Liikumine ühelt vaatel järgmisele.
137. Tarkvaraprotsessi etapid.
138. Iteratiivne arendamine: Rational Unified Process (RUP)
139. Modelleerimine.
140. Mudelite klassifikatsioon.
141. Käitumise analüüs.

142. Interaktsioonide analüüs.
143. Struktuuri analüüs.
144. Interaktsioonide disain.
145. Struktuuri disain.
146. Käitumise disain.
147. Kuidas on varem arhitektuurist mõeldud
148. Moodsa süsteemiarhitektuuri mõtte juured
149. Arhitektuuri olemus
150. Arhitektuuri roll
151. Mis on arhitektuur?
152. Mis on keerukus ja miks on see oluline?
153. Riigi kui selle arhitektuur
154. Riigi infosüsteemi kui terviku kontekstis ja miks on see oluline?
155. Süsteemid ja nende dünaamika
156. ICBM ja nende ohutus
157. Tänapäevased keerulised süsteemid
158. Süsteemi definitsioon ja süsteemimõtlemine
159. Arhitektuur süsteemimõtlemise kontekstis
160. Vorm, funktsioon ja kontseptsioon
161. Arhitekti roll
162. Miks koodi disain on vajalik?
163. Clean Code
164. Koodi lihtsa disaini neli elementi ja näited
165. Objektorienteeritud disain
166. Disaini mustrid
167. Finants- või tehnoloogiaettevõtted?
168. «Äri», kelle jaoks IT taust on juba eeldus.
169. Lean põhimõtted.
170. Arenduse infrastruktuur ja konfiguratsioonihaldus
171. Mida endast kujutab arenduse infrastruktuur?
172. Milleks hallata nõudeid?
173. Milleks planeerida?
174. Versioonihaldus.
175. Build/Deploy. Toodangusse minek.
176. Virtualiseerimine.