

Szoftverfejlesztési módszerek és paradigmák

Fejlesztő eszközök hatékony használata

1. A fejlesztői eszköz kiválasztásánál milyen szempontokat érdemes figyelembe venni és miért?
2. Jellemezd egy hatékony tervezési, fejlesztési és javítási mikociklust (mind a hármat)! Mire érdemes odafigyelni?
3. Mik a SOLID elvek? Röviden ismertesd mind az ötöt! Válassz ki egyet és jellemezd részletesen!

Módszertanok

4. Adj kritikát a vízésés és az evolúciós szoftverfejlesztési modellekre!
5. Adj jellemzést a különböző szerződés modellekről!
6. Írd le, hogy a különböző szerződés modellek mire motiválják a résztvevőket!
7. Írd le, hogy a különböző szerződés modellek kockázataira milyen kockázat-mérséklési technikákat ismersz!
8. A különböző szerződés modellek milyen elvárásokat támasztanak a módszertannal szemben?
9. Mit gondolsz az alkalmazásban megjelenő rugalmasságról?

Tesztelés

10. A tesztelésbe milyen szerepkörű résztvevők vannak bevonva? Mi a tipikus feladatuk a teszteléssel kapcsolatban? Jellemezd, hogy a projekt különböző fázisaiban mik a módosítások költségei!
11. Mi az ellenőrzés és mi a validáció? Mit tartalmaz a teszterv? Röviden jellemezd az elemeket! Mit értünk tesztelési stratégián?
12. Mi a unit tesztelés alapkonceptiója? Mik egy unit teszt tipikus lépései?
13. Ismertesd a jó unit teszt ismérveit, röviden jellemezd őket!
14. Röviden jellemezd az NUnit keretrendszert, ismertesd a főbb attribútumokat, használatukat, ellenőrző módszereket!
15. Jellemezd a moq keretrendszert! Add meg a főbb osztályok, mutasd be a legfontosabb kódolási mintákat (nem kell feltétlenül kód, elég a koncepció)!
16. Mi jelentenek az IoC és DI rövidítések? Mik a fogalmak jellemzői? Mit jelent ebben a kontextusban a resolution, resolution root, injection?
17. A DI keretrendszerek tipikusan milyen életciklus modelleket támogatnak? Mutasd meg a dryloc keretrendszer néhány kapcsolódó funkcióját!
18. Írd le, hogy egy adatbázis alapú alkalmazáshoz milyen módon érdemes unit tesztek készíteni és miért?
19. Jellemezd a Repository mintát! Mi az előnye, hátránya más megoldáshoz képest? Milyen viszonyban van az ORM technológiákkal?
20. Jellemezd a property és konstruktor alapú DI megoldásokat!
21. Jellemezd a különböző Factory mintákat és Service Locator mintát DI szempontból! Hasonlítsd össze a két megoldást!

22. Írd le, hogy az alkalmazást hogyan kell strukturálni ha unit-tesztelni szeretnénk? Milyen módszerek segítenek ebben? Mik a különböző megoldások előnyei, hátrányai?

Source control

23. Mik az előnyei a verzió kezelő eszközök használatának? Milyen verzionálási modelleket (nem elágaztatási stratégiát!) ismersz, mik ezek előnyei, hátrányai?
24. Ismertesd a központosított és elosztott verziókezelés előnyeit, hátrányait!
25. Milyen elágaztatási stratégiákat ismersz? Jellemezd őket röviden!
26. Jellemezd a git legfontosabb alap objektumait, adj példát egy módosítási művelet során végbemenő változásokra!
27. Jellemezd a git rebasing és a merge műveleteit!

Fejlesztési módszertanok

28. A szoftverfejlesztési folyamat tevékenységei. Foglald össze és röviden mutasd be a szoftverfejlesztési aktivitásokat!
29. Mutasd be a szoftverrendszerek, alkalmazások értékeit!
30. Mutasd be a szoftverprojektek életciklusának elemeit, azok jellemzőit!
31. Mutasd be a szoftverfejlesztési folyamat lépéseit, röviden elemezd őket!
32. Hasonlítsd össze a vízésés, a spirál és az iteratív fejlesztési módszereket!
33. Mutasd be az iteratív és inkrementális fejlesztési módszerek jellemzőit!
34. Mutasd be a *Capability Maturity Model Integration* (CMMI) különböző területeinek céljait és jellemzőit!
35. Mutasd be a *Capability Maturity Model Integration* (CMMI) különböző szintjeit és azok jellemzőit!

Követelményelemzés és szoftvertervezés

36. Foglald össze az *Structured Systems Analysis And Design Method* (SSADM) módszer lényegét, fázisait, előnyeit és hátrányait!
37. Foglald össze a követelményelemzés céljait, módszereit, azok előnyeit és hátrányait!
38. Mutasd be a követelményelemzés során végzett aktivitásokat és azok jellemzőit!
39. Mutasd be a követelményelemzés során előtérbe kerülő különböző követelménytípusok jellemzőit!
40. Mutasd be a követelményelemzés során előtérbe kerülő kihívásokat!
41. Foglald össze az üzleti elemzés céljait!
42. Foglald össze az üzleti elemzés területeit, technikáit és jellemzőit!

Szoftver tervezési módszerek

43. Mutasd be a szoftvertervezés során alkalmazott alapelveket és koncepciókat!
44. Mi a szoftvermodellézés szerepe a szoftvertervezés során? Milyen módszereket, technikákat ismersz?

45. Miért használunk a modellezés során eltérő absztrakciós szintű modelleket? Mutass rá példákat!

Agilis fejlesztési módszerek

46. Foglald össze az agilis módszerek értékeit és elveit!
47. Foglald össze az agilis tervezés jellemzőit és szintjeit!
48. Mutasd be az agilis módszerek során alkalmazott kiadás tervezést és annak jellemzőit! Térj ki a felhasználói sztorik szerepére!
49. Foglald össze az agilis iterációk jellemzőit és az iteráció tervezés lépéseit!
50. Mutasd be a napi *stand-up* esemény és a "Kész, kész" (done, done) jellemzőit!
51. Foglald össze az *eXtreme Programming* (XP) jellemzőit és eszközeit!
52. Foglald össze az *eXtreme Programming* (XP) értékeit és elveit!
53. Mutasd be a *Scrum* fejlesztési folyamatot és jellemzőit!
54. Mutasd be a *Scrum* ceremóniák jellemzőit!
55. Mutasd be a *Scrum* szerepköröket és a sprint tervezés jellemzőit!

Projektmenedzsment

56. Jellemezd a szervezeti stratégiát és viszonyát a projekthez!
57. Jellemezd a lineáris-funkcionális szervezeti formát!
58. Jellemezd a projekt orientált szervezeti formát!
59. Jellemezd a mátrix szervezeti formát!
60. Írj a projekt tipológiáról!
61. Mi a projekt karakterisztika, kik egy projekt résztvevői?
62. Sorold fel a PMBOK folyamatcsoportjait és tudásterületeit!