

Ellenőrző kérdések a Rendszertervezés (VIMIM238) tárgy Scherer Balázs által tartott részéhez

1. Mutassa be, hogy hogyan alakult a *Standish Group* felmérése alapján a sikeresen végrehajtott projectek aránya az elmúlt években (arányokat írjon, pontos számok nem kellene). Kommentálja a felmérés eredményeit!
2. Mutassa be, hogy statisztikailag a fejlesztési ciklus mely szakaszában kerül a rendszerbe a legtöbb hiba, azok hol fedeződnek fel és mennyibe kerül a javításuk? Mit segíthet ezen egy jól alkalmazott fejlesztési módszertan?
3. A beágyazott rendszereket fejlesztő cégek miért törekednek az *ISO9001:2000* mellett más szabvány szerinti tanúsításra is. Milyen szabványok jöhetnek itt még szóba?
4. Röviden mutassa be a CMMI főbb jellegzetességeit és célját. Mi a különbség a *folytonos* és *lépcsőzetes* megközelítés között mikor melyiket célszerű használni?
5. Ismertesse a CMMI *lépcsőzetes* megközelítésének *érettségi* szintjeit!
6. Ismertesse a CMMI *folytonos* megközelítésének *képességi* szintjeit!
7. Milyen csoportokba sorolhatóak a CMMI folyamatai és ezek nagy csoportok milyen viszonyba vannak egymással?
8. Soroljon fel legalább 3, a CMMI *Maturity level 2* eléréséhez szükséges folyamatot!
9. Soroljon fel legalább 3 olyan folyamatot, amelyek a CMMI *Maturity level 2* eléréséhez nem szükségesek, a CMMI *Maturity level 3* -hoz ugyanakkor már igen!
10. Ismertesse a *Project menedzsment* folyamatokat lényegét összefoglalóan!
11. Mi az a Gantt diagramm? Milyen adatokat tartalmaz? Rajzoljon egy egyszerű fejlesztési folyamathoz illeszkedő Gantt diagrammot!
12. Milyen módszereket használnak a project tervezés alatt a szükséges idő meghatározására?
13. Mutassa be a COCOMO alap gondolatát (képletek nem kellene)!
14. Mutassa be a *Requirement Management* folyamatot!
15. Mi az a *Requirement tracking*? Mutasson rá példát!
16. Soroljon fel legalább 8 követelmény osztályt beágyazott rendszerek esetére, amivel a *Requirement Management* folyamatnak foglalkoznia kell!
17. Mi a konfigurációmenedzsment folyamat célja, milyen lépéseket tartalmaz?
18. Mik azok a konfigurációs komponensek és konfigurációs egységek (Mondjon példát mindkettőre)?
19. Mutassa be a konfigurációs egységek és konfigurációs komponensek viszonyát egy életszerű példán!
20. Mutassa be, hogy hogyan történik a konfigurációmenedzsmentben résztvevő komponensek azonosítása! Milyen szempontok vannak?
21. A konfiguráció menedzsment rendszernek milyen tárolási helyei szoktak lenni?
22. Mutassa be a Triviális verziókövetés („*Total Commander módszer*”) előnyeit és hátrányait!
23. Ismertesse a verziókövető rendszerekhez tartozó alapfogalmakat (*repository, working-copy, checkout, commit, update, merge*)!

24. Ismertesse a Lock-Modify-Unlock megközelítés és annak előnyeit, hátrányait!
25. Ismertesse a Copy-Modify-Merge megközelítés, annak előnyeit, hátrányait!
26. Mutassa be az SVN-nél is alkalmazott Trunks, Tags, Branch ágak szerepét, egy példán keresztül.
27. Mutassa be röviden a *Vizesés* életciklus modellt! Ismertesse annak tulajdonságait!
28. Mutassa be röviden a *Spirál* életciklus modellt! Ismertesse annak tulajdonságait!
29. Mutassa be az alap *V-modell*t! Miben tér el ez a *Vizesés* modelltől?
30. Ismertesse egy tényleges komplett rendszer tervezésénél alkalmazott hierarchia szinteket, hogyan képesek ezek a szintek beépülni a *V-modell* szemléletébe?
31. Mutasson be egy tipikus rendszer fejlesztését *V-modell* alapján, amely iterációkat is tartalmaz!
32. Mutassa be a *Követelményanalízis, és a Logikai rendszerterv elkészítése* lépést.
33. Mutassa be egy ábrán a logikai rendszerterv és technikai rendszerterv közötti kapcsolatot!
34. Ismertesse a technikai rendszer architektúra specifikálásának lépéseit!
35. Mit a jelentősége a technikai rendszer architektúra specifikálása során elvégzett „szabályozási körök elemzése és specifikációja” lépésnek.
36. Mit a jelentősége a technikai rendszer architektúra specifikálása során elvégzett „Real-time követelmények feltérképezése és specifikációja” lépésnek?
37. Mit a jelentősége a technikai rendszer architektúra specifikálása során elvégzett „Elosztott működés elemzése, az elosztott rendszerek és hálózatok specifikálása” lépésnek?
38. Mit a jelentősége a technikai rendszer architektúra specifikálása során elvégzett „A megbízhatóság és biztonság szempontjából fontos részek azonosítása” lépésnek?
39. Mutassa be a „biztonságossági és megbízhatósági analízis” lépéseit!
40. Mi az a „Limit Risk (kockázati határ)” és mi köze van a rendszertervezéshez?
41. Mi az SIL *Safety Integrity Level* és mik alapján kerül meghatározásra?
42. Hogyan befolyásolja a meghatározott SIL (*Safety Integrity Level*) besorolás a rendszertervezés további lépéseit? Mutasson rá példát!
43. Mutassa be a *Szoftver architektúra megtervezésének* lépéseit!
44. Mutassa be egy általános beágyazott rendszer szoftverrétegei és magyarázza meg az egyes rétegek szerepét!
45. Ismertesse egy általános beágyazott rendszer tipikus működési módjait és azok kapcsolatait! Röviden ismertesse az egyes működési módokban elvárt viselkedést!
46. Ismertesse a *Szoftver komponensek specifikálása* lépést!
47. Indokolja, hogy miért célszerű sok esetben a szoftverkomponensek tervezésénél viselkedési és az adatmodell szétválasztani!
48. Ismertesse a szoftver komponensek, illetve rendszerek real-time viselkedésének tervezésénél alkalmazott fogalmakat és módszereket!
49. Ismertesse a „*A szoftver komponensek implementálása*” lépéssel kapcsolatos problémákat, illetve nehézségeket!

50. Mi a jelentősége a *megfelelő implementációs környezet kiválasztásának* a szoftverkomponensek implementációja során? Mik az implementációs környezet kiválasztásának főbb szempontjai?
51. Mutassa be egy általános *Coding style guide* által tárgyalt területeket!
52. Mutasson példát arra, hogy egy *Coding style guide*-ban hogyan szabályozzák a forrásfájlok nevét és belső szerkezetét!
53. Mutasson példát arra, hogy egy *Coding style guide*-ban hogyan szabályozzák a változók elnevezését!
54. Milyen szabályhalmazokat ismer az implementációs nyelvi készlet korlátozására?
55. Mutassa be röviden a MISRA-C szerepét és céljait!
56. Röviden foglalja össze, hogy a V-modell tesztelési ágánál melyik hierarchia szinten milyen jellegű tesztelési eljárásokat használnak! Mutasson példát minden hierarchia szintre!
57. Mutassa be, hogy a *Szoftverkomponensek tesztelése* milyen főbb lépésekből áll!
58. Mutassa be az szoftverkomponensek, rendszerek adatmodelljének tesztelésére leggyakrabban alkalmazott módszereket!
59. Mutassa be az szoftverkomponensek, rendszerek viselkedési modelljének tesztelésére leggyakrabban alkalmazott módszereket!
60. Mutassa be röviden, hogy a *Szoftverkomponensek integrációjánál* milyen lépéseket kell végrehajtani.
61. Mutassa be a rendszerintegráció utáni *integrációs tesztelés* főbb lépéseit!
62. Mi az az ISTQB (International Software Testing Qualifications Board)?
63. Soroljon fel legalább 5 darab SVN parancsot és ismertesse ezek működését!
64. Rajzolja fel egy azonos programrészen dolgozó két fejlesztő konfliktusának és a konfliktus feloldásának folyamatát SVN parancsok segítségével!
65. Mi a *doxygen* és milyen problémákra próbál megoldást adni és hogyan?
66. Soroljon fel legalább 8 *doxygen* parancsot!
67. Adja meg egy tipikus fájl fejlécének kommentezését *doxygen* segítségével!
68. Adja meg egy tipikus függvény fejlécének kommentezését *doxygen* segítségével!
69. Mire szolgál az *@addtogroup* parancs és miért fontos a dokumentáció elkészítése során, hogyan jelenik meg a dokumentációban? Mutasson példát rá!
70. Mutasson példát arra, amikor a C fordító működésének specifikátlansága problémát tud okozni!
71. Mutasson példát arra, amikor ugyanazt a kódot, ami egy 32 bites vezérlőn hibátlanul futott egy 8 bites vezérlőre lefordítva és azon futtatva a végrehajtásnál hibát okoz, vagy okozhat!
72. Mutasson példát arra, amikor ugyanazt a kódot, ami egy 8 bites vezérlőn hibátlanul futott egy 32 bites vezérlőre lefordítva és azon futtatva a végrehajtásnál hibát okoz, vagy okozhat!