

581358 Ohjelmistoarkkitehtuurit  
Koekysymyksiä S2016  
Antti-Pekka Tuovinen

Kokeeseen tulee yksi laajempi, opittujen asioiden soveltamista vaativa tehtävä ja 3 – 4 suppeampaa peruskysymystä.

6 pisteen peruskysymyksiä (vastauksen pituus n.½ - 1 sivua)

1. Anna *Ohjelmistoarkkitehtuurin* määritelmä ja selitä se lyhyesti. (jompikumpi riittää, eli rakenteita (I) tai suunnittelupäätöksiä (II) korostava määritelmä)
2. Luonnehdi arkkitehtuuria koskevia suunnittelupäätöksiä. Mistä ne tunnistaa? Mikä erottaa ne muista suunnittelupäätöksistä? Minkälaisia konkreettisia asioita arkkitehtuuripäätökset tyypillisesti koskevat (pari esimekkiä riittää)?
3. Selosta ohjelmistoarkkitehtuurin ohjaavaa (prescriptive) ja kuvailevaa (descriptive) käyttöä ohjelmistokehityksessä.
4. Selosta lyhyesti George Fairbanks'in tunnistamat *kolme ohjelmistoarkkitehtuurin suunnittelun ja käytön tapaa* ohjelmistokehityksessä. Mikä on tunnusomaista kullekin tavalle?
5. Selitä mitä tarkoittaa ohjelmistoarkkitehtuurin käyttö *laatuviipuna*. Anna jokin esimerkki.
6. Selosta syitä, miksi *arkkitehtuurityötä* (arkkitehtuurin suunnittelua ja arviointia) ei aina tehdä ohjelmistoprojekteissa.
7. Minkälaisissa tilanteissa tai minkä tyyppisissä ohjelmistoprojekteissa *arkkitehtuurityö* (eli arkkitehtuurin mallintaminen, suunnittelu ja analysointi) on tärkeää?
8. Selitä mitä tarkoittaa *Information Hiding / Separation of Concerns / Coupling*. Anna jokin esimerkki, jossa se ilmenee.
9. Selitä lyhyesti A. Cockburn'in esittelemä *Walking Skeleton* –projektipatternin idea. Miten tämä liittyy ohjelmistoarkkitehtuurin kehittämiseen?
10. Mitä Frank Buschmann tarkoittaa ASR:llä eli arkkitehtuurin kannalta tärkeällä vaatimuksella (Architecturally Significant Requirement)? Mikä on niiden rooli ja merkitys arkkitehtuurin vaiheittaisessa suunnittelussa ja toteuttamisessa?
11. Selitä <jokin arkkitehtuurityyli/patterni> perusidea, sen käytöstä seuraavia hyviä ominaisuuksia ja soveltuvuutta (minkälaisiin tilanteisiin/sovelluksiin se sopii).

12. Selitä miten suunnittelutaktiikka eroaa arkkitehtuurityylistä tai arkkitehtuuripatternista. Anna esimerkki jostakin laatutekijästä ja sitä koskevasta suunnittelutaktiikasta.
13. Selosta perustellen, miksi ja milloin ohjelmistoarkkitehtuurista kannattaa laatia *malleja*?
14. Selitä, ohjelmistoarkkitehtuurin sovellusaluemallin (domain model) / suunnittelumallin (design model) tyypillinen rakenne, eli kerro, mistä osista malli koostuu. Kuvaa lyhyesti kunkin mallin osan pääasiallinen tarkoitus/informaatio sisältö.
15. Mikä on *komponentti*? Selosta lyhyesti komponentin keskeiset ominaisuudet.
16. Mitä *näkymä* (view) tarkoittaa ohjelmistoarkkitehtuurin mallinnuksessa/dokumentaatioissa? Entä *näkymätyyppi* (viewtype)?
17. Selitä miksi suunnittelun (arkkitehtuurin) ja toteutuksen (koodin) välillä on usein kuilu. Miksi arkkitehtuuri ei aina näy koodissa?
18. Selosta tapoja kaventaa suunnittelun (arkkitehtuurin) ja toteutuksen (koodin) välistä kuilua.
19. Miten *ohjelmistokehitys* (software framework) eroaa (luokka- tai funktio-) kirjastosta (esimerkiksi tiedostojen ja hakemistojen käsittelyyn tarkoitettu luokkakirjasto)?
20. Mikä on *tuoterunko* ja mitä *tuoterunkopohjainen ohjelmistokehitys* tarkoittaa? Selitä lyhyesti tuoterunkopohjaisen ohjelmistokehityksen päävaiheet (prosessit).
21. Miten muunneltavuudella (variability) tarkoitetaan tuoterunkopohjaiseen ohjelmistokehityksen (eli tuoteperheiden) yhteydessä? Minkälaista muunneltavuutta tuoterungoissa tyypillisesti on?
22. Mikä on *arkkitehtuurin arvioinnin* (architectural analysis) tarkoitus, minkälaisia tavoitteita sillä voi olla ja mihin ohjelmiston ominaisuuksiin se pääasiassa kohdistuu?
23. Selosta *skenaarioihin perustuvan arkkitehtuurin arvioinnin* perusideat. Kuvaile lyhyesti ATAM (Architecture Trade-off Analysis Method) –menetelmän mukaisen arvioinnin kulku (arvioinnin vaiheet ja niiden päätavoitteet).

#### Laajempia kysymyksiä (10 – 12 pistettä, n. 1 – 2 sivua)

24. Selosta ohjelmiston *laatuvaatimusten* (quality attribute requirements, non-functional requirements) suhdetta ohjelmiston arkkitehtuuriin.
25. Selosta tyylin/patternin X perusidea. Selitä X:ssä määriteltyjen komponenttien ja konnektoreiden vastuut/roolit sekä analysoi, millä tavalla näissä malleissa ilmenevät Separation of Concerns ja Information Hiding –periaatteet.

26. <Sanallinen kuvaus jostakin palvelusta/järjestelmästä>. Analysoi ja perustele, mitkä kurssilla käsitellyt arkkitehtuurityylit/patternit sopisivat järjestelmän ohjelmistoarkkitehtuurin perustaksi.
27. <Sanallinen kuvaus jostakin palvelusta/järjestelmästä>. Arvioi järjestelmän toteutukseen liittyviä riskejä (voit käyttää mielikuvitustasi täydentämään edellä esitettyä kuvausta) ja selitä perustellen, miten arkkitehtuurityö voi auttaa tunnistamiesi riskien vähentämisessä.