

Helsingin yliopisto
Tietojenkäsittelytieteen laitos

581358 Ohjelmistoarkkitehtuurit
Koekysymyksiä S2015
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 G.Fairbanks'in kuvaamat kolme ohjelmistoarkkitehtuurin suunnittelun ja käytön tapaa.
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. Selitä <jokin arkkitehtuurityyli/patterni> perusidea, sen käytöstä seuraavia hyviä ominaisuuksia ja soveltuvuutta (minkälaisiin tilanteisiin/sovelluksiin se sopii).
11. Selitä miten suunnittelutaktiikka eroaa arkkitehtuurityylistä tai arkkitehtuuripatternista. Anna esimerkki jostakin laatutekijästä ja sitä koskevasta suunnittelutaktiikasta.

12. Selosta perustellen, miksi ja milloin ohjelmistoarkkitehtuurista kannattaa laatia *malleja*?
13. 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.
14. Mitä *näkymä* (view) tarkoittaa ohjelmistoarkkitehtuurin mallinnuksessa/dokumentaatioissa? Entä *näkymätyyppi* (viewtype)?
15. Selitä miksi suunnittelun (arkkitehtuurin) ja toteutuksen (koodin) välillä on usein kuilu. Miksi arkkitehtuuri ei aina näy koodissa?
16. Selosta tapoja kaventaa suunnittelun (arkkitehtuurin) ja toteutuksen (koodin) välistä kuilua.
17. Miten *ohjelmistokehys* (software framework) eroaa (luokka- tai funktio-) kirjastosta (esimerkiksi tiedostojen ja hakemistojen käsittelyyn tarkoitettu luokkakirjastosta)?
18. Mitä *tuoterunkopohjainen ohjelmistokehitys* tarkoittaa? Selitä lyhyesti päävaiheet (prosessit).
19. Miten muunneltavuuden hallinta (variability management) liittyy tuoterunkopohjaiseen ohjelmistokehitykseen (eli tuoteperheisiin)?
20. 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)

21. Selosta ohjelmiston *laatuvaatimusten* (quality attribute requirements, non-functional requirements) suhdetta ohjelmiston arkkitehtuuriin.
22. 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.
23. <Sanallinen kuvaus jostakin palvelusta/järjestelmästä>. Analysoi ja perustele, mitkä kurssilla käsitellyt arkkitehtuurityylit/patternit sopisivat järjestelmän ohjelmistoarkkitehtuurin perustaksi.
24. <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.