581358 Ohjelmistoarkkitehtuurit

Harjoitus 2

1. Pohdi ja selitä, miten yleiset suunnitteluperiaatteet *Separation of Concerns* ja *Information Hiding* ilmenenvät Kolmistasoarkkitehtuurissa (N-Tier), MVC-arkkitehtuurissa (Web-MVC tai alkuperäinen käyttöliittymien toteuttamiseen liittyvä MVC) sekä Putket ja suotimet -arkkitehtuurissa (Pipes and Filters).
2. Tutustu Microsoftin MVVM (Model-View-ViewModel) patterniin ja vertaa sitä alkuperäiseen (käyttöliittymätason) MVC:hen. Miten MVVM eroaa perus-MVC:stä? Minkä takia? Mitä tarkoittaa ”data binding”? Voimistuuko vai heikkeneekö elementtien välinen kytkentä (suorat ja epäsuorat riippuvuudet) MVVM –patternissa MVC:hen verrattuna?
    - http://en.wikipedia.org/wiki/Model_View_ViewModel
3. Selvitä, mitä tarkoittaa *kokonaisarkkitehtuuri* eli *yritysarkkitehtuuri* (*Enterprise Architecture)*. Mitä Suomen *tietohallintolaki* sanoo kokonaisarkkitehtuurista ja julkisesta hallinnosta (kts. esim. Valtiovarainministeriön sivuilta: Vastuualueet / Julkisen hallinnon ICT)?
4. Kolmistasoarkkitehtuuri (3-tier, N-tier) on hyvin yleisesti käytetty arkkitehtuuri tietojärjestelmissä (*de-facto* standardi yritysjärjestelmissä). Mitä vanhempi järjestelmä, sitä todennäköisemmin se noudattaa tätä arkkitehtuuria (ns. perinne- eli legacy -järjestelmät). Nykyaikaiset web-ohjelmistot taas perustuvat yleensä web-MVC –arkkitehtuuriin tai sen muunnelmiin. Ajatellaan tilannetta, jossa yrityksen X perinnetietojärjestelmään Y halutaan kehittää moderni web-pohjainen käyttöliittymä. Ajatuksena on muuttaa järjestelmän Y arkkitehtuuria ottamalla käyttöön web-MVC-arkkitehtuuria noudattava kehys (framework), jonka avulla toteutetaan uusi käyttöliittymä. Liiketoimintaprosessien olemassa olevaa toteutusta ei kuitenkaan haluta tehdä uudestaan, vaan käyttää jo vanhan kolmitasoratkaisun sovellus- ja datatasojen ratkaisuja sellaisenaan. Hahmottele järjestelmän Y uudelle versiolle arkkitehtuuri, joka yhdistää web-MVC- ja kolmitasoarkkitehtuurin (sillä tarkkuustasolla, kuin luentomateriaalissa on käytetty).

    web-MVC: http://advancedkittenry.github.io/koodaaminen/arkkitehtuuri/index.html
5. Eoin Woods toimittaa *The Pragmatic Architect* –kirjoitussarjaa *IEEE Software* –lehdessä. Hänen tuorein juttunsa käsittelee arkkitehtuurityön ja ketterien tiimien yhdistämistä:
    - Woods, Eoin. "Aligning Architecture Work with Agile Teams." IEEE Software 5 (2015): 24-26. http://www.computer.org/csdl/mags/so/2015/05/mso2015050024.pdf

    Ron Jeffries on yksi *Agile Manifeston* alkuperäisistä allekirjoittajista, ja hän puolestaan on kirjoittanut suunnittelun emergenssistä (Emergent Design) omalla sivustollaan: http://ronjeffries.com/xprog/articles/emergent-design/

    Lue artikkelit, ja pohdi seuraavia kysymyksiä. Mitä yhtäläisyyksiä ja toisaalta eroja näet kirjoittajien lähestymistavoissa? Miten eroja on esimerkiksi arkkitehdin roolissa? Mistähän erot voisivat johtua?