

Ohjelmistoarkkitehtuurit

Luento 1

13.10.2016

581358 Ohjelmistoarkkitehtuurit

1

Oppimistavoitteet

- Mitä ohjelmistoarkkitehtuuri on?
- Miten aihetta kurssilla käsitellään?

13.10.2016

581358 Ohjelmistoarkkitehtuurit

2

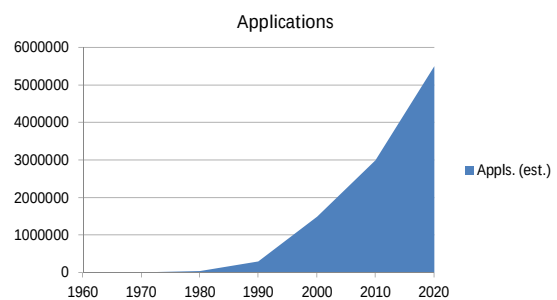
ALUKSI

13.10.2016

581358 Ohjelmistoarkkitehtuurit

3

Sovellusten määrä USA:ssa eri vuosikymmeninä (arvio)*



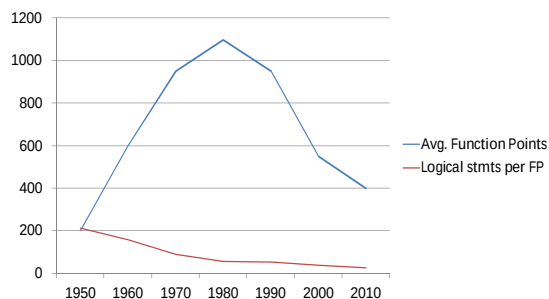
*) Capers Jones, *The Technical and Social History of Software Engineering*, Addison-Wesley Professional, 2013.

13.10.2016

581358 Ohjelmistoarkkitehtuurit

4

Sovellusten keskim. koko USA:ssa Function Point -kokomittarilla (arvio)*



*) Capers Jones, *The Technical and Social History of Software Engineering*, Addison-Wesley Professional, 2013.

13.10.2016

581358 Ohjelmistoarkkitehtuurit

5

Ohjelmistojen merkityksestä

- 2010-luvun tietoyhteiskunta: lukematon määrä palveluita toteutetaan tietojärjestelminä, joista monet ovat kooltaan ja käyttäjämääriltään maailmanlaajuisia
- Tietokonelaitteistot kehittyneet huomasti: laskentateho, nopea tiedonsiirto, valtavien tietomäärien hajautettu tallennus ja rinnakkainen käsittely, "esineiden Internet"
- Virtualisoinnin ja "pilvestä" saatavan laskenta- ja tallennuskapasiteetin ansiosta kokonaisiä järjestelmiä voidaan toteuttaa nyt puhtaasti *ohjelmistoratkaisuna* (ei omia kiinteitä laiteinvestointeja)
- Laadultaan hyvien ohjelmistojen kustannustehokas toteuttaminen on entistään tärkeämpää
 - Tarkoituksenmukaisuus, tehokkuus, luotettavuus, saatavuus, käyttökokemus, muunneltavuus, skaalautuvuus, turvallisuus,
 - Kilpailu on kovaa ja markkinat avoimet, time-to-market
- Ohjelmistoarkkitehtuuri on tärkeässä roolissa halutun laadun ja tuottavuuden saavuttamisessa

13.10.2016

581358 Ohjelmistoarkkitehtuurit

6

Ohjelmistotekniikan kehityksestä

2010	Websovelluslusrat, HTML5, pilvipalvelut, mobiili, Bigdata, IoT, Continuous X, containers	Lean, ekosysteemit, start-upit, experimentointi
2000	Tuotepuhearkkitehtuurit, MDA, väliohjelmistot, ohjelmistotallustat, aspektit, dynaamiset (skripti-) kielet, virtualisointi	Ketterät menetelmät (agile methods)
1995	CASE-välineet: uudelleenkäyttö, testaus, mittaus Patterns (ratkaisumallit), sovelluskehikset, ohjelmistoarkkitehtuurit, arkkitehtuurityylit, arkkitehtuurien kuvaukset, UML	Liiketoiminta, organisaatio, Unified Process & inkrementaalinen kehitys
1990	CASE-välineet: käyttöliittymien piirtäminen CASE-välineet: kaaviotyökalut (olioteknikaat)	Graafiset käyttöliittymät, käytettävyyt
1985	CASE-välineet: koodingenerointi (5GL)	Prosessit: CMM, spiraalimalli & riskien hallinta
1980	Oliot, uudelleenkäyttö, osiosuunnittelu, tietokone- ja järjestelmäarkkitehtuuri, relaatiotietokannat (4GL) Suunnittelumenetelmät: tietovirtakaaviot, ER-malli	
1970	Modulaarisuus, tiedon kätkeminen, (2GL, 3GL)	Prosessit: vesiputous
1950 - 60	Aliohjelmat (1GL)	

13.10.2016

581358 Ohjelmistoarkkitehtuurit

7

OHJELMISTOARKKITEHTUURI

13.10.2016

581358 Ohjelmistoarkkitehtuurit

8

Määritelmiä

- Ohjelmistotyyppien kirjo on hyvin suuri
 - Sulautetut ohjelmistot pienlaitteissa – giga-aplikaatiot webissä miljoonine yhtäaikaisine käyttäjinene
- Myös *ohjelmistoarkkitehtuuri* on osoittautunut monitahoiseksi asiaksi, jota voidaan perustellusti tarkastella eri näkökulmista ja eri tavoittein
 - Johtaa hieman erilaisiin määritelmiin ja painotuksiin
- Kaksi "koulukuntaa" (1990's → 2010's)
 - Rakenteita* korostava koulukunta
 - Suunnittelupäätösten* merkitystä korostava koulukunta

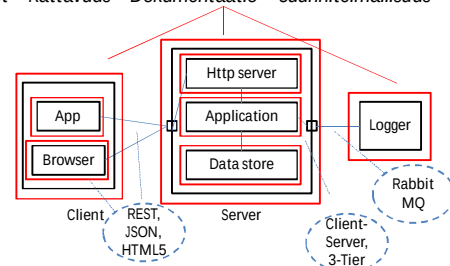
13.10.2016

581358 Ohjelmistoarkkitehtuurit

9

Rakenteet <> Suunnittelupäätökset

Rakenteet – Kattavuus – Dokumentaatio – Suunnitelmallisuus – Hallinta



Suunnittelupäätökset – Ominaisuudet – Ymmärrys – Inkrementaalisuus

13.10.2016

581358 Ohjelmistoarkkitehtuurit

10

I - Rakenteita korostava määritelmä

- Software Engineering Institute / Carnegie Mellon University:

"The software architecture of a system is the set of structures needed to reason about the system, which comprise software elements, relations among them, and properties of both."

"Järjestelmän ohjelmistoarkkitehtuuri tarkoittaa *rakenteita*, joita tarvitaan *päätelmien tekemiseksi järjestelmästä ja sen ominaisuuksista*. Rakenteet koostuvat ohjelmistoelementeistä, niiden välisistä suhteista, ja näiden molempien piirteistä."

(Katso myös esimerkiksi ISO/IEC 42010 standardin arkkitehtuurin määritelmä https://en.wikipedia.org/wiki/ISO/IEC_42010 ja IEEEExplore)

13.10.2016

581358 Ohjelmistoarkkitehtuurit

11

Mitä *rakenteet* ovat?

- Määritelmä jättää tämän avoimeksi
 - Ei yksi vaan *joukko* rakenteita
 - Mikä tahansa tarkoituksenmukainen kokoelma "elementtejä" ja niiden välisiä suhteita voi muodostaa rakenteen
 - Ohjelmistoja on monenlaisia ja monenkokoisia!
- Kurssilla opimme, minkälaisia rakenteita yleensä kannattaa tarkastella (arkkitehtuurin mallinnus)
 - Rakenteiden valinta riippuu sovelluksesta, sovellusalueesta ja toteutusteknologioista, "one size does not fit all"

13.10.2016

581358 Ohjelmistoarkkitehtuurit

12

Tyypillisiä elementtejä ja rakenteita

- Ohjelmistolla on kolme "olomuotoa", joissa esiintyy merkittäviä rakenteita
 - Staattinen, dynaaminen, operatiivinen
- Staattinen olomuoto
 - Ohjelmiston koodi tiedostoissa ja tiedostojärjestelmän kansioissa (*moduulit*)
 - konfiguraatio- ynnä muut alustus- ja datatiedostot ja – resurssit
 - Näiden väliset sisältyvyys ja viittaussuhteet (tiedosto- ja kooditasolla)

13.10.2016

581358 Ohjelmistoarkkitehtuurit

13

Tyypillisiä elementtejä ja rakenteita

- Dynaaminen olomuoto
 - Ohjelmiston suoritusaikana (tietokoneen muistiin ja käyttöjärjestelmään) luomat aktiiviset ja passiiviset oliot, prosessit ja säikeet, data-entiteetit sekä näiden kommunikaatorajapinnat ja –yhteydet (*komponentit* ja *konnektorit*)
 - Olioiden (ja prosessien) väliset data- ja kontrollivuot, muut viestintä- ja yhteistoimintamekanismit
 - Usean olion muodostamat, roolinsa ja toimintansa perusteella selkeästi erottuvat kokonaisuudet (alijärjestelmä, komponentti)
 - Application, Server, Load Balancer, Business Process, Container, Bean
 - Palvelu ≈ Olio- ja prosessijoukko
 - *Huomaa: ei välttämättä yksi-yhteen vastaavuutta* staattisten rakenteiden kanssa

13.10.2016

581358 Ohjelmistoarkkitehtuurit

14

Tyypillisiä elementtejä ja rakenteita

- Operatiivinen olomuoto (sijoittelu, allokatio)
 - Dynaamisten olioiden ja palveluiden sijoittelu ja suoritus laitteistossa osana *tietyä järjestelmää*
 - (Virtuaali-) palvelimet, tallennusvälineet, tietoliikennekanavat
 - Palveluiden ylös- ja alasajo, päivitykset, monitorointi, skaalautuvuuden ja saatavuuden hallinta, pääsynvalvonta, transaktiot
 - Ohjelmiston paketointi, konfigurointi ja piirteiden (features) valinta tiettyä käyttökontekstia (asiakasta) varten

13.10.2016

581358 Ohjelmistoarkkitehtuurit

15

Pohdittavaksi

- Mitä elementtejä ja rakenteita voisi olla Java-tekniikoilla toteutetussa web-sovelluksessa tai mobiiliapplikaatiossa?
 - Staattinen, dynaaminen, operatiivinen olomuoto

13.10.2016

581358 Ohjelmistoarkkitehtuurit

16

Abstraktit rakennekuvaukset

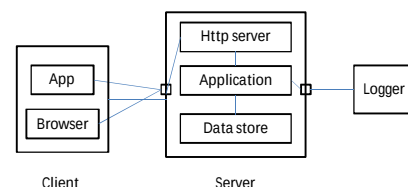
- Jonkin rakenteen dokumentoitu kuvaus (arkkitehtuurinäköymä, *view*) on usein *abstrakti*
 - Tarkasteltavan asian kannalta epäoleelliset yksityiskohdat jätetään pois
 - Jos kyse on suunnitelmasta, ei toteutuksen yksityiskohtia vielä voida tai edes haluta kiinnittää
 - Tarkastelu tapahtuu yleensä *järjestelmätasolla*, jossa rakenneosat ovat vastuitaan (toiminnot, tietosisältö) suhteellisen suuria (esim. asiakas – palvelin - tietovarasto)
 - Näkymät muodostavat usein *hierarkian*, jossa alemman tason näköymä *tarkentaa* jotain osaa ylemmän tason näköymästä

13.10.2016

581358 Ohjelmistoarkkitehtuurit

17

Abstraktit rakennekuvaukset



13.10.2016

581358 Ohjelmistoarkkitehtuurit

18

Rakenteet ja ohjelmiston ominaisuudet

- Oleellista on, että rakenteiden perusteella voidaan tehdä päätelmiä jollekin *sidosryhmälle tärkeistä ohjelmiston makrotason ominaisuuksista*
 - Tarkoituksenmukaisuus, luotettavuus, tehokkuus, helppokäyttöisyys, turvallisuus, muutettavuus, ulkoiset riippuvuudet, skaalautuvuus, ...
 - Asiakas, loppukäyttäjät, kehittäjä, ylläpitäjä, liiketoiminnan johto, viranomainen, ...
- Ohjelmiston halutut ominaisuudet vaikuttavat siis siihen, mitä rakenteita arkkitehtuurissa tarkastellaan
 - Ja kuinka tarkkoja tai yksityiskohtaisia kuvauksia tarvitaan
- Arkkitehtuuri ei siis ole itsetarkoitus vaan *väline*

19

II - Suunnittelupäätöksiä korostava määritelmä

- “We propose to view a software architecture as a set of *explicit architectural design decisions*. In this perspective, the *software architecture is the result of the architectural design decisions made over time*”¹
- Ohjelmistoarkkitehtuuri koostuu joukosta selkeitä suunnittelupäätöksiä. Arkkitehtuuri on ajan myötä tehtyjen päätösten tulos.

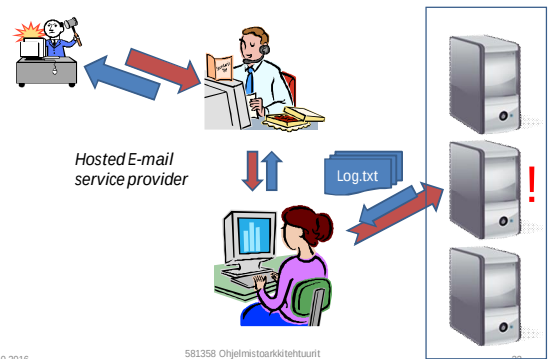
1. Jansen, A., & Bosch, J. (2005). Software architecture as a set of architectural design decisions. In *Software Architecture, 2005. WICSA 2005. 5th Working IEEE/IFIP Conference on* (pp. 109-120). IEEE.

Arkkitehtuuritason suunnittelupäätös?

- "All architecture is design, but not all design is architecture."¹
- Tunnusmerkkejä
 - Päätös koskee järjestelmätason asioita ja vaikuttaa pääasiassa muiden vaatimusten kuin järjestelmän tarjoamia palveluja tai toimintoja koskevien vaatimusten toteutumiseen
 - Voi osoittaa kalliiksi, jos menee pieleen
 - Tasapainottelee eri suuntiin vetävien voimien välillä
 - Ohjaa ja/tai rajoittaa muita suunnittelupäätöksiä

1. Grady Booch, 2006 https://www.ibm.com/developerworks/community/blogs/gradybooch/entry/on_design
13.10.2016 581358 Ohjelmistoarkkitehtuurit

Esimerkki - Rackspace



Tutkittava järjestelmä

- Sähköpostipalvelimien tuottamien lokitietojen tallennus ja käsittely asiakastukea varten
- Tehtävä (missio)
 - Asiakkaan sähköpostin toimintaa koskevien ongelmien selvittämisen tukeminen sähköpostipalvelimien tuottamien lokitietojen perusteella
 - Tarjottava ajantasaista tietoa riittävän pitkältä ajanjaksolta
- Kolme järjestelmäversiota, kolme erilaista arkkitehtuuria

23

V. 1

- *S-postipalvelinten paikalliset lokitiedostot*
 - Teknisen tuen pitää pyytää operaattoria ottamaan yhteys asiakkaan postipalvelimeen ja tutkimaan sen paikallista lokia ongelmien selvittämiseksi
 - Rackspace toteutti skriptin, joka automaattisesti ottaa yhteyden joukkoon palvelimia ja suorittaa haun (grep) halutun tyyppisen palvelimen lokista; operaattorit voivat muokata hakuheitoja
- Ongelmia:
 - Palvelinten, asiakkaiden ja tukipyyntöjen määrän kasvaessa yhä useammin suoritettavat haut alkoivat vaikuttaa haitallisesti palvelinten suorituskykyyn
 - Hakujen tekemiseen tarvittiin aina operaattoria, mikä vei jatkuvasti kasvavan osan heidän työajastaan

24

V. 2

- **Lokitietojen keskittäminen yhteen tietokantaan**
 - S-postipalvelimet lähettävät muutaman minuutin väliajoin lokitietonsa ladattavaksi samaan relaatiotietokantaan tk-palvelimelle
 - Tekniikoilla on suora web-yhteys tietokantapalvelimelle ennalta ohjelmoitujen hakujen suorittamista varten
 - Yksittäisten palvelimien lähettämät päivitykset kootaan yhteen säännöllisesti suoritettavaksi massapäivitykseksi (suoritetaan 10 min. välein) tietokantapalvelimen suorituskäyvän ylläpitämiseksi
- **Ongelmia:**
 - Datan ja kyselyiden määrän jatkuvasti kasvaessa tietokantapalvelimen suorituskäyky ja kapasiteetti muodostuivat pullonkaulaksi, häiriöt lisääntyivät
 - Haut hidastuivat ja dataa hävisi välillä, vain muutaman päivän edestä lokitietoja kyettiin säilyttämään, varmuuskopioita ei ollut

13.10.2016

581358 Ohjelmistoarkkitehtuurit

25

V. 3

- **Lokitiedostojen rinnakkainen indeksointi hajautetussa tiedostojärjestelmässä**
 - S-postipalvelimet syöttävät lokitiedostonsa usealle yksinkertaiselle palvelimelle (cluster) hajautettuun tiedostojärjestelmään (Hadoop DFS)
 - Hadoop map-reduce –operaatio indeksoi lokitiedostot ja muodostaa kaiken kattavan indeksin 15 min. välein
 - Tekniikoilla on web-liittymä hakujen tekemiseen kuten ennenkin
 - Hakujen suoritus indeksistä on nopeaa, uudenlaisen haun ohjelmointi ja suoritus kestää kuitenkin joitain tunteja
 - Täydellinen varmuuskopiointi, lokitietojen säilytys 6 kk:n ajalta

13.10.2016

581358 Ohjelmistoarkkitehtuurit

26

Ratkaisujen vertailua

	V. 1	V. 2	V. 3
Toiminnallisuus	✓	✓	✓
Skaalautuvuus – datan ja hakujen määrä	heikko	välttävä	hyvä
Viive – lokitietojen saanti s-palvelimilta	välitön	10 min.	15 min.
Joustavuus – uudet haut	hyvä	hyvä	tydyttävä

13.10.2016

581358 Ohjelmistoarkkitehtuurit

27

Huomioita 1

- Järjestelmän tekniselle tuelle näkyvät *toiminnot ovat käytännössä samat* kaikissa versioissa
 - Tukihenkilö sai kaikissa tapauksissa tarvitsemansa (samat) tiedot lokeihin kirjautuneista tapahtumista
- Järjestelmän toiminnallisuus (*mitä järjestelmä tekee*) ei siis ohjannut teknisiä ratkaisuja
 - 1. versioonkin olisi voinut toteuttaa web-liittymän hakujen pyytämiseen operaattoreilta

13.10.2016

581358 Ohjelmistoarkkitehtuurit

28

Huomioita 2

- Esimerkkijärjestelmän *arkkitehtuuriin* vaikuttavat eniten vaaditut *laatuominaisuudet* (quality attributes, -ilities, 'non-functional requirements')
 - Skaalautuvuus, tiedonsaannin viive, joustavuus uusien hakujen määrittelyssä
- Ylivoimaisesti tärkein järjestelmän ominaisuus on *skaalautuvuus* datan ja hakujen määrän suhteen
 - Huonosti skaalautuva järjestelmä ei lainkaan kykene täyttämään tehtävänsä toiminnan laajentuessa

13.10.2016

581358 Ohjelmistoarkkitehtuurit

29

Huomioita 3

	V. 1	V. 2	V. 3
Toiminnallisuus	✓	✓	✓
Skaalautuvuus – datan ja hakujen määrä	heikko	välttävä	hyvä
Viive – lokitietojen saanti s-palvelimilta	välitön	10 min.	15 min.
Joustavuus – uudet haut	hyvä	hyvä	tydyttävä

13.10.2016

581358 Ohjelmistoarkkitehtuurit

30

Huomioita 3

- Skaalautuvuuden parantuessa versiosta toiseen muut ominaisuudet hieman kärsivät
 - Viive kasvaa (lokidatan tilannekuvan ikä) -> 15 min
 - Joustavuus vähenee, koska uusien hakujen lisääminen vaatii enemmän työtä
- Joistakin vaadituista ominaisuuksista täytyy tinkiä toisten, tärkeämpien, hyväksi (*trade off*)
 - Laatuominaisuuksien *priorisointi* esimerkissä:

Skaalautuvuus > Viive > Joustavuus

13.10.2016

581358 Ohjelmistoarkkitehtuurit

31

...entäs se arkkitehtuuri?

- Esimerkin järjestelmäversioiden kuvausten yhteydessä mainitut tekniset ratkaisut, eli *suunnittelupäätökset*, määrittävät järjestelmän arkkitehtuurin
- Ohjelmistoarkkitehtuuriin kuuluvat siis ne ohjelmiston suunnittelupäätökset, jotka *merkittävästi vaikuttavat* ohjelmiston *laadullisten ominaisuuksien* saavuttamiseen
 - Suunnittelupäätösten pohjalta voidaan rakentaa järjestelmä, jolla on halutut ominaisuudet
- *Sovellusalue* ja kunkin sovelluksen/järjestelmän *vaatimukset* vaikuttavat siihen, mitkä tekniset seikat milloinkin nousevat arkkitehtuuriasioiksi ja mitä rakenteita on syytä tarkastella

13.10.2016

581358 Ohjelmistoarkkitehtuurit

32

Tyypillisiä arkkitehtuuripäätöksiä

- Järjestelmän jakaminen osajärjestelmiin (pääkomponentteihin) ja niiden roolien, toimintojen ja keskinäisten riippuvuuksien määrittely
 - Osajärjestelmien jako päätason komponentteihin jne.
- J:n käsittelemän ja säilyttämän tiedon tallennus- ja saantiratkaisut
- J:n rajapintojen (käyttöliittymät, ohjelmalliset rajapinnat) tunnistaminen ja erottaminen niiden kautta käytettävien toimintojen sisäisestä toteutuksesta
- J:n ajoaikaiseen suorituskäyttöön, skaalautuvuuteen, saatavuuteen, resurssien käyttöön ja turvallisuuteen vaikuttavat ratkaisut
 - Palvelimet, prosessit ja säikeet, tieto- ja tapahtumavirrat, etäyhteydet, valvonta, tunnistaminen, sijoittelu laitteistoon, osien välinen viestintä
- J:n kehitystä ja ylläpitoa tukevat ratkaisut
 - Järjestelmällinen uudelleenkäyttö, tuoterungon käyttö, komponenttien vaihdettavuus ja siirrettävyys, diagnostiikka, testattavuus
- Jonkin teknologia-alustan ja sen referenssiarkkitehtuuri(e)n käyttö

13.10.2016

581358 Ohjelmistoarkkitehtuurit

33

RAJANVETO ARKKITEHTUURIN JA MUUN SUUNNITTELUN VÄLILLÄ

13.10.2016

581358 Ohjelmistoarkkitehtuurit

34

Arkkitehtuuri ja muu suunnittelu (design, detailed design)

- Suunnittelu (design) – päätetään, minkälaisista elementeistä ohjelmiston toteutus koostuu
 - ohjelmistoelementtien vastuut, rajapinnat, riippuvuudet
 - Esim. luokat/oliot ja niiden roolit ja rajapinnat, metodit ja niiden semantiikka, patternien soveltaminen, keskeiset algoritmit, luokkien paketointi, tietomallit, konfigurointi jne.
- Miten vedetään raja arkkitehtuurisuunnittelun ja muun suunnittelun (detailed design) välille?
 - Täytyykö raja vetää?
 - Joskus detaljit voivat olla tärkeitä; esimerkiksi komponenttistandardin muotovaatimukset ladattavien komponenttien rajapinnan koodaukselle ja niiden ulkoisten riippuvuuksien määrittelylle
- Arkkitehtuuriset suunnittelupäätökset
 - Vaikuttavat ohjelmiston keskeisten *laatuvaatimusten toteutumiseen*
 - Tai muuten *vaikuttavat laajasti* ohjelmiston osien suunnitteluun

13.10.2016

581358 Ohjelmistoarkkitehtuurit

35

Arkkitehtuuriratkaisut ja muut suunnittelupäätökset



13.10.2016

581358 Ohjelmistoarkkitehtuurit

36

Arkkitehtuuri ja muu suunnittelu

- Rajaa arkkitehtuurin ja (yksityiskohtien) suunnittelun välille ei ole aina helppo vetää, mutta järjestelmän laatuominaisuuksiin vaikuttavat suunnittelupäätökset ovat yleensä tunnistettavissa
 - Kaikilla ohjelmistoilla siis on arkkitehtuuri
 - On toki parempi, että arkkitehtuuriratkaisut ovat *harkittuja* kuin sattumalta syntyneitä (vrt. "Big Ball of Mud")
- Kokeneella ammattilaisella on mielessään tietynlaisten *ongelmien hahmoja* ja niiden *ratkaisutapoja* (conceptual model), jotka auttavat häntä tunnistamaan arkkitehtuurin kannalta tärkeitä päätökset -> näkee metsän puilta

13.10.2016

581358 Ohjelmistoarkkitehtuurit

37

Kuningas näkee pelin toisin



http://img.yle.fi/urheilu/jalkapallo/article/979622_ece/ALTERNATES/w580/studioduvarin2011menen%20kuusela.JPG
581358 Ohjelmistoarkkitehtuurit

13.10.2016

38

OA:N MÄÄRITELMISTÄ

13.10.2016

581358 Ohjelmistoarkkitehtuurit

39

Ohjelmistoarkkitehtuurin määritelmien yhteensovittaminen

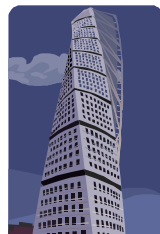
- Molemmat ovat sinänsä järjeviä näkökulmia ja täydentävät toisiaan
- Rakenteita tarvitaan ohjelmiston monimutkaisuuden yleiseen hallintaan ja suunnittelupäätöksiä perusteluineen (rationale) tarvitaan ratkaisujen syvällistä ymmärtämistä ja arviointia varten
 - Dokumentaatio- ja kontrollikeskeiset prosessit pitävät rakenteiden kattavaa määrittelyä tärkeänä
 - Suunnittelupäätöksiin keskittyvä arkkitehtuurityö tyypillistä ketterille kehitysmenetelmille: vain todella tärkeät asiat dokumentoidaan, rakennekuvauksia voidaan generoida automaattisesti tarvittaessa

13.10.2016

581358 Ohjelmistoarkkitehtuurit

40

Ohjelmistoarkkitehtuuri vs. Arkkitehtuuri



Suomen arkkitehtiiliiton puheenjohtaja Erkki Rautio *Tietoviikon* haastattelussa 29.4.2012: "Ei se herätä suuttumusta, vaan ehkä enemmänkin sääliä, kun it-ammattilaisilla ei ole omia ammattinimikkeitä."

Huippuarkkitehtuuri on *taidemuoto*, jossa rakennusten käyttäjien tarpeet usein jäävät vähemmälle huomiolle kuin taiteelliset näkökohdat

Venustus (!), Firmitas(?), Utilitas(??)

13.10.2016

581358 Ohjelmistoarkkitehtuurit

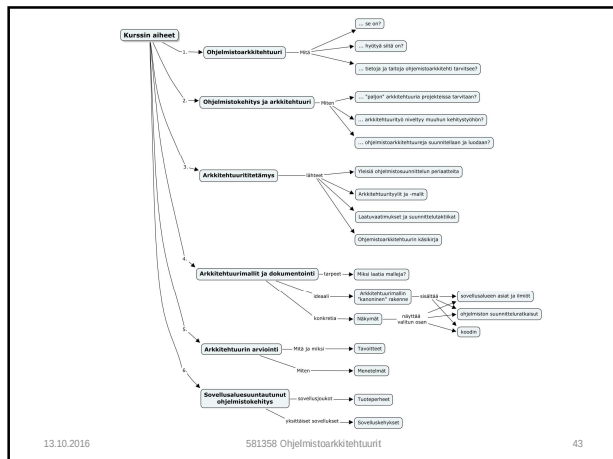
41

KURSSIN SISÄLTÖ

13.10.2016

581358 Ohjelmistoarkkitehtuurit

42



Pääoppimistavoitteet

- Tietää mitä ohjelmistoarkkitehtuuri on ja mihin sitä käytetään
- Tietää ja osaa soveltaa muutamia keskeisiä arkkitehtuurityylejä ja –patteja ohjelmiston arkkitehtuurin suunnitteluun
- Tietää miksi ja milloin arkkitehtuurimalleja laaditaan ja osaa muodostaa "kanonisen rakenteen" mukaisia näkymiä arkkitehtuuriin
- Hallitsee arkkitehtuurin arvioinnin ja ohjelmistokehysten (framework) suunnittelun perusideat (syvennetään harjoitustyössä)

13.10.2016

581358 Ohjelmistoarkkitehtuurit

44