

Ohjelmistoarkkitehtuurit

Luento 1

1.9.2015

581358 Ohjelmistoarkkitehtuurit

1

Oppimistavoitteet

- Mitä ohjelmistoarkkitehtuuri on?
- Mitä hyötyä siitä on?
- Miten aihetta kurssilla käsitellään?

1.9.2015

581358 Ohjelmistoarkkitehtuurit

2

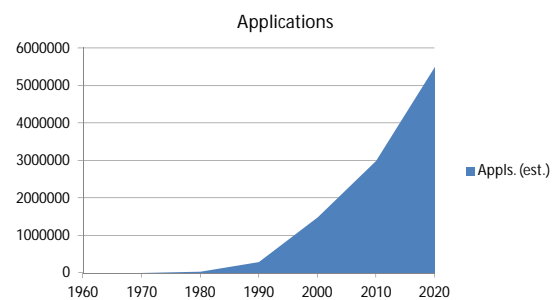
ALUKSI

1.9.2015

581358 Ohjelmistoarkkitehtuurit

3

Sovellusten määrä USA:ssa eri vuosikymmeninä (arvio)*



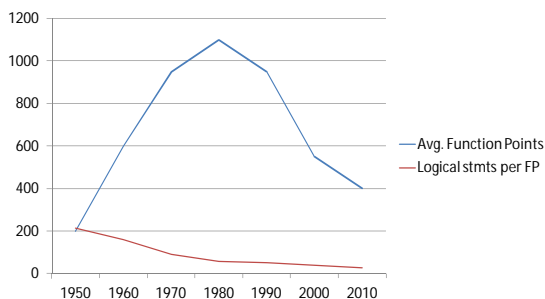
*)Capers Jones, *The Technical and Social History of Software Engineering*, Addison-Wesley Professional, 2013.

1.9.2015

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.

1.9.2015

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, valtaviin tietomääriin hajautettu tallennus ja rinnakkainen käsittely
- Virtualisoinnin ja "pilvestä" saatavan laskenta- ja tallennuskapasiteetin ansiosta kokonaisia järjestelmiä voidaan toteuttaa nyt puhtaasti *ohjelmistoratkaisuin*a (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

1.9.2015

581358 Ohjelmistoarkkitehtuurit

6

Ohjelmistotekniikan kehityksestä

<p>2010</p> <p>2000</p> <p>1995</p> <p>1990</p> <p>1985</p> <p>1980</p> <p>1970</p> <p>1950 - 60</p>	<p>Websovelluslустat, HTML5, pilvipalvelut, mobiili, Big data, IoT, <i>Continuous X</i></p> <p><i>Tuotepuhearkkitehtuurit, MDA</i>, valiohjelmitot, ohjelmistoalustat, aspektit, dynaamiset (skripti-) kielet</p> <p>CASE-valineet: uudelleenkäyttö, testaus, mittaus</p> <p><i>Patterns (ratkaisumallit)</i>, <i>sovelluskahykset</i>, <i>ohjelmistoarkkitehtuurit</i>, <i>arkkitehtuurityylit</i>, <i>arkkitehtuurien kuvaukskielet</i>, <i>UML</i></p> <p>CASE-valineet: käyttöliittymien piirtäminen</p> <p>CASE-valineet: kaaviotyökalut (oliotekniikat)</p> <p>CASE-valineet: koodingenerointi (5GL)</p> <p>Oliot, uudelleenkäyttö, olosuunnittelu Tietokone- ja järjestelmäarkkitehtuuri, Relaatiotietokannat (4GL)</p> <p>Suunnittelumenetelmät: tietovirtakaaviot, ER-malli</p> <p><i>Modulaarisuus</i>, <i>tiedon katkeminen</i>, (2GL, 3GL)</p> <p>Aliohjelmat (1GL)</p>	<p>Lean, ekosysteemit, start-upit, experimentointi</p> <p>Ketterät menetelmät (agile methods)</p> <p>Liiketoiminta, organisaatio, Unified Process & inkrementaalinen kehitys</p> <p>Graafiset käyttöliittymät, käytettävyys</p> <p>Prosessit: CMM, spiraalimalli & riskien hallinta</p> <p>Prosessit: vesiputous</p>	<p>7</p>
--	---	--	----------

1.9.2015 581358 Ohjelmistoarkkitehtuurit

OHJELMISTOARKKITEHTUURI

1.9.2015 581358 Ohjelmistoarkkitehtuurit 8

Määritelmiä

- Ohjelmistotyyppien kirjo on hyvin suuri
 - Sulautetut ohjelmistot – On-line giga-aplikaatiot
- Myös *ohjelmistoarkkitehtuurion* 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

1.9.2015 581358 Ohjelmistoarkkitehtuurit 9

Rakenteet <> Suunnittelupäätökset

Rakenteet – Kattavuus – Dokumentaatio – Suunnitelmällisuus – Hallinta

Suunnittelupäätökset – Ominaisuudet – Ymmärrys – Inkrementaalisuus

1.9.2015 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."

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

1.9.2015 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
- Kurssilla opimme, minkälaisia rakenteita yleensä kannattaa tarkastella (arkkitehtuurin mallinnus)
 - Rakenteiden valinta riippuu sovelluksesta, sovellusalueesta ja toteutusteknologiosta, "one size does not fit all"

1.9.2015 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)

1.9.2015

581358 Ohjelmistoarkkitehtuurit

13

Tyypillisiä elementtejä ja rakenteita

- Dynaaminen olomuoto
 - Ohjelmiston suoritusajana (tietokoneen muistiin ja käyttöjärjestelmään) luomat aktiiviset ja passiiviset oliot, prosessit ja säikeet, data-entiteetit sekä näiden kommunikaatorajapinnat ja –yhteydet (*komponentit ja konektorit*)
 - Olioiden (ja prosessien) väliset data- ja kontrollivuotot, muut viestintä- ja yhteistoimintamekanismit
 - Usean oliion 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

1.9.2015

581358 Ohjelmistoarkkitehtuurit

14

Tyypillisiä elementtejä ja rakenteita

- Operatiivinen olomuoto (sijoittelu, allokatio)
 - Dynaamisten olioiden ja palveluiden sijoittelu ja suoritus laitteistossa osana *tiettyä 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

1.9.2015

581358 Ohjelmistoarkkitehtuurit

15

Pohdittavaksi

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

1.9.2015

581358 Ohjelmistoarkkitehtuurit

16

Abstraktit rakennekuvaukset

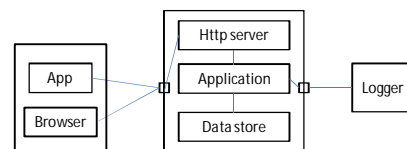
- Jonkin rakenteen dokumentoitu kuvaus (arkkitehtuurinäkökulma, *view*) on usein *abstrakti*
 - Tarkasteltavan asian kannalta epäoleelliset yksityiskohdat jätetään pois
 - Jos kyse on suunnitelmasta, ei toteutuksen yksityiskohtia vielä voida tai edes haluta kiinnittää
 - Tarkastelu tapahtuu yleensä *järjestelmätasolla*,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

1.9.2015

581358 Ohjelmistoarkkitehtuurit

17

Abstraktit rakennekuvaukset



1.9.2015

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, loppukäyttäjä, 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*

1.9.2015

581358 Ohjelmistoarkkitehtuurit

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.

1.9.2015

581358 Ohjelmistoarkkitehtuurit

20

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 osoittautua kalliiksi, jos menee pieleen
 - Tasapainottelee eri suuntiin vetävien voimien välillä
 - Ohjaa ja/tai rajoittaa muita suunnittelupäätöksiä

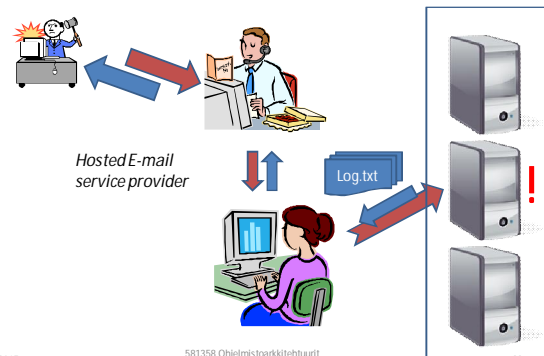
1. Grady Booch, 2006 https://www.ibm.com/developerworks/community/blogs/gradybooch/entry/on_design

1.9.2015

581358 Ohjelmistoarkkitehtuurit

21

Esimerkki - Rackspace



1.9.2015

581358 Ohjelmistoarkkitehtuurit

22

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

1.9.2015

581358 Ohjelmistoarkkitehtuurit

23

V. 1

- *S-postipalvelinten paikalliset lokitiedostot*
 - Teknisen tuen pitää pyytää operaattoria ottamaan yhteys asiakkaan postipalvelimeen ja tutkimaan lokitietoja 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 tukipyynnöiden määrän kasvaessa yhä useammin suoritettavat haut alkoivat vaikuttaa palvelinten suorituskykyyn
 - Hakujen tekemiseen tarvittiin aina operaattoria, mikä vei jatkuvasti kasvavan osan heidän työajastaan

1.9.2015

581358 Ohjelmistoarkkitehtuurit

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 suorituskyvyn ylläpitämiseksi
- **Ongelmia:**
 - Datan ja kyselyiden määrän jatkuvasti kasvaessa tietokantapalvelimen suorituskyky ja kapasiteetti muodostuivat pullonkaulaksi, häiriöt lisääntyivät
 - Haut hidastuivat ja dataa hävisi välillä, vain muutaman päivän edestä lokitietoja kyettiin säilyttämään, varmuuskopioita ei ollut

1.9.2015

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

1.9.2015

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

1.9.2015

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

1.9.2015

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

1.9.2015

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

1.9.2015

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

1.9.2015

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

1.9.2015

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 suorituskykyyn, skaalautuvuuteen, saatavuuteen, resurssien käyttöön ja turvallisuuteen vaikuttavat ratkaisut
 - Palvelimet, prosessit ja säikeet, tieto- ja tapahtumavirrat, etäyhteydet, valvonta, tunnistaminen, sijoittelu laitteistoon
- J:n kehitystä ja ylläpitoa tukevat ratkaisut
 - Järjestelmällinen uudelleenkäyttö, tuoterunon käyttö, komponenttien vaihdettavuus ja siirrettävyys, diagnostiikka, testattavuus
- Jonkin teknologia-alustan ja sen referenssiarkkitehtuuri(e)n käyttö

1.9.2015

581358 Ohjelmistoarkkitehtuurit

33

RAJANVETO ARKKITEHTUURIN JA MUUN SUUNNITTELUN VÄLILLÄ

1.9.2015

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, pattersien 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

1.9.2015

581358 Ohjelmistoarkkitehtuurit

35

Arkkitehtuuriratkaisut ja muut suunnittelupäätökset



1.9.2015

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

1.9.2015

581358 Ohjelmistoarkkitehtuurit

37

Kuningas näkee pelin toisin



<http://img.yle.fi/urheilu/jalkapallo/artikkelit/978622/ALTERNATIVES/w580/studiokuva%20ittmenen%20kuussa.JPG>
581358 Ohjelmistoarkkitehtuurit

1.9.2015

38

OA:N MÄÄRITELMISTÄ

1.9.2015

581358 Ohjelmistoarkkitehtuurit

39

Ohjelmistoarkkitehtuurin määritelmien yhteensovittaminen

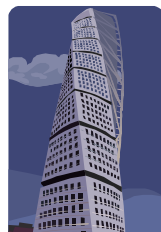
- Molemmat ovat sinänsä järkeviä näkökulmia ja täydentävät toisiaan
- Rakenteita tarvitaan ohjelmiston monimutkaisuuden yleiseen hallintaan ja suunnittelupäätöksiä perusteluineen (rationale) tarvitaan ratkaisujen syvällistä ymmärtämistä ja arviointia varten
 - Dokumentaatio- ja kontrollikeskeiset prosessit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

1.9.2015

581358 Ohjelmistoarkkitehtuurit

40

Ohjelmistoarkkitehtuuri vs. Arkkitehtuuri



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

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

Venustas (!), Firmitas(?), Utilitas(??)

1.9.2015

581358 Ohjelmistoarkkitehtuurit

41

OHJELMISTOARKKITEHTUURIN HYÖDYT JA KÄYTTÖ

1.9.2015

581358 Ohjelmistoarkkitehtuurit

42

Arkkitehtuurin ilmeneminen

- Arkkitehtuuri on siis valikoitu joukko ohjelmisto-elementtien ja niiden välisten suhteiden muodostamia rakenteita tai joukko näitä koskevia suunnittelupäätöksiä
- Konkreettisesti nämä rakenteet / päätökset voivat ilmetä
 - Ideoina ja periaatteina (kehittäjien mielessä)
 - Konkreettisina rajoitteina (suunnittelulle ja toteutukselle)
 - Dokumentteina (muodollisina tai vapaamuotoisina)
 - Malleina (UML, formaalit mallinnuskielet)
 - Koodina (kirjastot, kehykset, alustat, esimerkit ja prototyypit, lähdekoodin struktuuri ja riippuvuudet, paketointi ja nimentä) ja suoritusajaisina objekteina
 - Implementaation ilmentämä arkkitehtuuri on tietysti lopullinen totuus (ei välttämättä vastaa dokumentoituja suunnitelmia)

1.9.2015

581358 Ohjelmistoarkkitehtuurit

43

Arkkitehtuurin käyttö

- Arkkitehtuurin käyttötavat ohjelmistokehityksessä voidaan jakaa karkeasti kahteen kategoriaan
 - *Preskriptiivinen* eli *ohjaava* käyttö
 - *Deskriptiivinen* eli *kuvaileva* käyttö

1.9.2015

581358 Ohjelmistoarkkitehtuurit

44

Ohjaava käyttö

- Arkkitehtuuri määrittää järjestelmän perusrakenteen
 - Analogiana eläimen luuranko
 - Arkkitehtuurin lähtökohdaksi voidaan valita *tyyli* (architectural style) tai *patterni* (architectural pattern), joka kiinnittää elementtien tyypit/vastuut/roolit ja niiden välisten liitosten ja yhteyksien ominaisuudet
 - Ei voida sanoa, onko jokin tyyli absoluuttisesti parempi kuin toinen, vaan täytyy arvioida tyylin *sopivuutta* kehitettävän ohjelmiston tarpeisiin nähden

1.9.2015

581358 Ohjelmistoarkkitehtuurit

45

Ohjaava käyttö

- Arkkitehtuuri vaikuttaa laatuominaisuuksiin
 - Arkkitehtuuri mahdollistaa haluttujen ominaisuuksien saavuttamisen
 - Sopimaton arkkitehtuuri voi myös estää sen !
- Arkkitehtuuri on (enimmäkseen) riippumaton toiminnallisuudesta
 - Saman toiminnallisuuden voi toteuttaa melkein minkälaisella arkkitehtuurilla tahansa
 - *Mutta*: huonosti valittu arkkitehtuuri voi tehdä toiminnallisuuden toteuttamisen vaikeaksi ja kalliiksi (joskus jopa mahdottomaksi) !

1.9.2015

581358 Ohjelmistoarkkitehtuurit

46

Ohjaava käyttö

- Arkkitehtuuri ohjaa toteutusta rajoitteiden avulla (guide rails)
 - Esimerkiksi käytettävyyden tai tietoturvallisuuden vuoksi halutaan suoraan kieltää eräät ominaisuuden kannalta huonoksi tiedetyt ratkaisut
 - Rajoitteet auttavat kehittäjiä monin tavoin
 - Kokemuksen siirto tiivistetyssä muodossa asiantuntijoilta, jotka ovat perustelluista syistä asettaneet rajoitteet
 - Käsitteellisen eheyden (conceptual integrity!) vahvistaminen ja kompleksisuuden vähentäminen "tarpeetonta luovuutta" rajoittamalla
 - Koodin ajonaikaisen käyttäytymisen helpompi ymmärtäminen

¹ "A single good idea consistently applied is better than several brilliant ideas scattered across a system" (Fred Brooks)

1.9.2015

581358 Ohjelmistoarkkitehtuurit

47

Kuvaileva käyttö

- Ohjelmiston ja suunnitteluratkaisujen ymmärtäminen
 - Abstrahointi yksityiskohtia pois suodattamalla
 - Sopivasti valitut rakenteet ja niitä tietystä näkökulmasta esittävät *näkymät* (view) ovat erinomaisia ymmärryksen lähteitä
 - Uudet kehittäjät, johto, asiakkaat, alihankkijat, jne. !
- Liiketoiminnallisten tavoitteiden toteutumisen seuraaminen
 - Tuoteperheet, COTS-komponenttien käyttö, integrointi ulkoisiin palveluihin, standardointi, lisensointi jne.

1.9.2015

581358 Ohjelmistoarkkitehtuurit

48

Kuvaileva käyttö

- Rajoitteiden noudattamisen valvonta
 - Esimerkiksi näkymän muodostaminen rakenneosien (komponenttien, kerrosten) välisistä riippuvuuksista, jolloin voidaan havaita arkkitehtuuriin rajoitteiden kieltämät riippuvuudet
 - Esim. kerrosarkkitehtuurissa alemman kerroksen komponentti kutsuu ylemmän kerroksen palvelua muuten kuin palvelun asettaman takaisinkutsurajapinnan kautta, **riippuvuuskytkien kieltäminen** !
- Organisaation kehittäminen
 - Vahvasti toisiinsa kytkeytyvien elementtien kehittämissvastuun jako kannattaa miettiä tarkkaan kommunikaatio-ongelmien välttämiseksi
 - Conwayn laki – organisaatio ja arkkitehtuuri muistuttavat ennen pitkää toisiaan

1.9.2015

581358 Ohjelmistoarkkitehtuurit

49

Muita hyötyjä

- Riskien hallinta
 - Arkkitehtuurityö kannattaa keskittää tunnistettujen riskien eliminoimiseen tai lieventämiseen
 - Kiinnitetään jatkuvaa huomiota tärkeimpiin teknisiin ughiin
- Vaatimusten täsmentäminen
 - Vaadittujen laatuominaisuuksien analysointi ja arkkitehtuurin suunnittelu niiden saavuttamiseksi auttaa huomaamaan ristiriitaisuuksia ja epätasallisuksia vaatimuksissa ja määrittelyissä
 - Arkkitehtuurin suunnitteluun liittyvä *laatuominaisuuksien tasapainottelu* (trade off) pakottaa *priorisoimaan laatuvaatimukset*

1.9.2015

581358 Ohjelmistoarkkitehtuurit

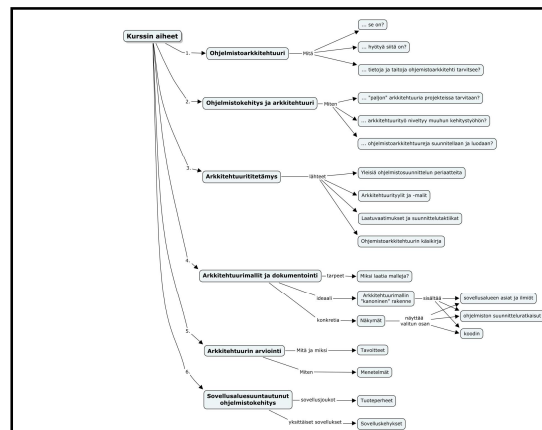
50

KURSSIN SISÄLTÖ

1.9.2015

581358 Ohjelmistoarkkitehtuurit

51



1.9.2015

581358 Ohjelmistoarkkitehtuurit

52

Pääoppimistavoitteet

- Tietää mitä ohjelmistoarkkitehtuuri on ja mihin sitä käytetään
- Tietää ja osaa soveltaa muutamia keskeisiä arkkitehtuurityylejä ja -patroneja 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)

1.9.2015

581358 Ohjelmistoarkkitehtuurit

53