

Ohjelmistoarkkitehtuurit

Luento 1

3.9.2013

581358 Ohjelmistoarkkitehtuurit

1

Oppimistavoitteet

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

3.9.2013

581358 Ohjelmistoarkkitehtuurit

2

ALUKSI

3.9.2013

581358 Ohjelmistoarkkitehtuurit

3

Ohjelmistotekniikan kehityksestä

2010	Websovellusalustat, pilvipalvelut, mobiili, Big data	Lean, ekosysteemit
2000	<i>Tuoteperhearkkitehtuurit, MDA, väliohjelmistot, ohjelmistoalustat, aspektit, dynaamiset (skripti-) kielet</i>	Ketterät menetelmät (agile methods)
1995	CASE-välineet: uudelleenkäyttö, testaus, mittaus <i>Patterns (ratkaisumallit), sovelluskehukset, ohjelmistoarkkitehtuurit, arkkitehtuurityylit, arkkitehtuurien kuvauskielet, UML</i>	Liiketoiminta, organisaatio
1990	CASE-välineet: käyttöliittymien piirtäminen CASE-välineet: kaaviotyökalut	Graafiset käyttöliittymät, käytettävyys
1985	CASE-välineet: koodingenerointi	Prosessit, CMM
1980	Oliot, uudelleenkäyttö, oliosuunnittelu Tietokone- ja järjestelmäarkkitehtuuri Suunnittelumenetelmät: tietovirtakaaviot, ER-malli	
1970	<i>Modulaarisuus, tiedon kätkeminen</i>	Prosessit: vesiputous
1960	Aliohjelmat	

3.9.2013

581358 Ohjelmistoarkkitehtuurit

4

Ohjelmistojen merkityksestä

- 2010-luvun tietoyhteiskunta: lukematon määrä palveluita toteutetaan tietojärjestelminä, joista monet ovat kooltaan ja käyttäjämääriltään maailmanlaajuisia
- Tietokonelaitteistot kehittyneet huimasti: laskentateho, nopea tiedonsiirto, valtaviin tietomääriin hajautettu tallennus ja rinnakkainen käsittely
- Virtualisoinnin ja "pilvestä" saatavan laskenta- ja tallennuskapasiteetin ansiosta kokonaisia järjestelmiä voidaan toteuttaa nyt puhtaasti *ohjelmistoratkaisuina* (ei laiteinvestointeja)
- Laadultaan hyvien ohjelmistojen kustannustehokas toteuttaminen on entistäkin tärkeämpää

3.9.2013

581358 Ohjelmistoarkkitehtuurit

5

OHJELMISTOARKKITEHTUURI

3.9.2013

581358 Ohjelmistoarkkitehtuurit

6

Mitä OA on?

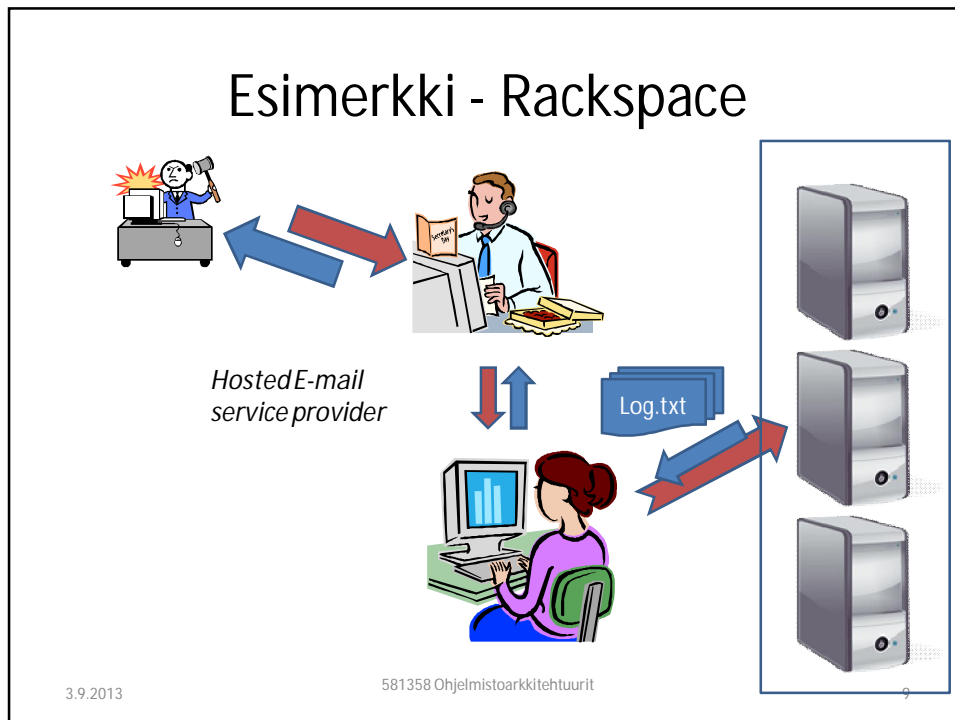
- Määritelmä (Software Engineering Inst. / CMU)

" 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."

...mutta mitä se tarkoittaa?

- Määritelmä ei kerro mitä rakenteet ovat
 - Ei yksi vaan *joukko* rakenteita
 - Mikä tahansa kokoelma "elementtejä" (koodimoduuleja, komponentteja, virtuaalikoneita) ja niiden välisiä suhteita (riippuvuuksia, käyttösuhteita, omistussuhteita) voi muodostaa rakenteen
- Oleellista on, että rakenteiden perusteella voidaan tehdä päätelmiä jollekin *sidosryhmälle tärkeistä* o:n *makrotason* ominaisuuksista
 - Asiakas, loppukäyttäjä, kehittäjä, ylläpitäjä, liiketoiminnan johto, viranomainen, ...
 - Tarkoituksenmukaisuus, luotettavuus, tehokkuus, helppokäyttöisyys, turvallisuus, muutettavuus, ulkoiset riippuvuudet, ...



Tutkittava järjestelmä

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

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
- Ongelmia:
 - Palvelinten, asiakkaiden ja tukipyyntöjen määrän kasvaessa yhä useammin suoritettavat haut alkoivat vaikuttaa palvelinten suorituskykyyn
 - Hakujen tekemiseen tarvittiin aina operaattoria, mikä vei jatkuvasti kasvavan osan heidän työajastaan

V. 2

- *Lokitietojen keskittäminen yhteen tietokantaan*
 - S-postipalvelimet lähettävät muutaman minuutin väliajoin lokitietonsa ladattavaksi samaan relaatiotietokantaan
 - Teknikoilla 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

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
 - Teknikoilla 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. ajalta

3.9.2013

581358 Ohjelmistoarkkitehtuurit

13

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

3.9.2013

581358 Ohjelmistoarkkitehtuurit

14

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

3.9.2013

581358 Ohjelmistoarkkitehtuurit

15

Huomioita 2

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

3.9.2013

581358 Ohjelmistoarkkitehtuurit

16

Huomioita 3

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

Skaalautuvuus > Viive > Joustavuus

3.9.2013

581358 Ohjelmistoarkkitehtuurit

17

...entäs se arkkitehtuuri?

- Esimerkin järjestelmäversioiden kuvausten yhteydessä mainitut tekniset ratkaisut (suunnittelupäätökset) kuuluvat järjestelmän arkkitehtuuriin
- Ohjelmistoarkkitehtuuriin kuuluvat siis ne ohjelmiston *suunnittelupäätökset*, jotka *merkittävästi vaikuttavat* ohjelmiston *laadullisten ominaisuuksien* saavuttamiseen
- *Sovellusalue* ja kunkin sovelluksen/järjestelmän *vaatimukset* vaikuttavat siihen, mitkä tekniset seikat milloinkin nousevat arkkitehtuuriasioiksi

3.9.2013

581358 Ohjelmistoarkkitehtuurit

18

Arkkitehtuuri ja muu suunnittelu

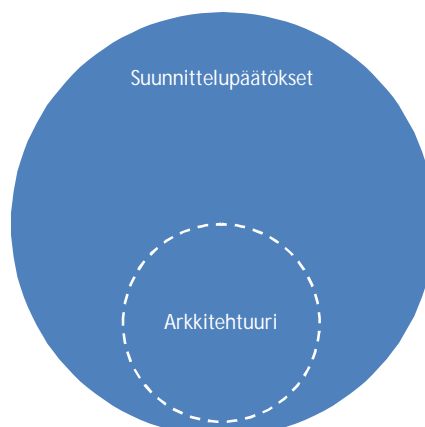
- Miten vedetään raja arkkitehtuurisuunnittelun ja muun suunnittelun (detailed design) välille?
 - Täytyykö raja vetää?
 - Joskus detaljit voivat olla tärkeitä, esimerkiksi komponenttistandardin yksityiskohtaiset muotovaatimukset ladattavien komponenttien rajapinnan koodaukselle
- Arkkitehtuurin kannalta merkittävät suunnittelupäätökset on kuitenkin syytä tunnistaa niiden *laajan vaikutuksen* takia

3.9.2013

581358 Ohjelmistoarkkitehtuurit

19

Arkkitehtuuriratkaisut ja muu suunnittelu



3.9.2013

581358 Ohjelmistoarkkitehtuurit

20

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 on arkkitehtuuri
 - On toki parempi, että arkkitehtuuriratkaisut ovat *harkittuja* kuin sattumalta syntyneitä (vrt. "Big Ball of Mud")
 - Kokeneella ammattilaisella on mielessään *ajatusmalleja* (conceptual model), jotka auttavat häntä tunnistamaan arkkitehtuurin kannalta tärkeitä päätökset -> näkee metsän puilta

3.9.2013

581358 Ohjelmistoarkkitehtuurit

21

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

3.9.2013

581358 Ohjelmistoarkkitehtuurit

22

OHJELMISTOARKKITEHTUURIN HYÖDYT JA KÄYTTÖ

3.9.2013

581358 Ohjelmistoarkkitehtuurit

23

Arkkitehtuurin ilmeneminen

- Arkkitehtuuri on siis valikoitu joukko ohjelmisto-elementtien ja niiden välisten suhteiden muodostamia rakenteita
- Konkreettisesti nämä rakenteet voivat ilmetä
 - Ideoina ja periaatteina (kehittäjien mielessä)
 - Konkreettisinä rajoitteina (suunnittelulle ja toteutukselle)
 - Dokumentteina (muodollisina tai vapaamuotoisina)
 - Malleina (UML, formaalit mallinnuskielet)
 - Koodina (kirjastot, kehykset, alustat, esimerkit ja prototyypit, lähdekoodin struktuuri ja nimentä)

3.9.2013

581358 Ohjelmistoarkkitehtuurit

24

Arkkitehtuurin käyttö

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

Ohjaava käyttö

- Arkkitehtuuri määrittää järjestelmän perusrakenteen
 - Analogiana eläimen luuranko
 - Arkkitehtuurin lähtökohdaksi voidaan valita *tyyli* (architectural style) tai *malli* (architectural pattern), joka kiinnittää elementtien 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

Ohjaava käyttö

- Arkkitehtuuri vaikuttaa laatuominaisuuksiin
 - Arkkitehtuuri voi mahdollistaa ja toisaalta myös estää haluttujen ominaisuuksien saavuttamisen
- Arkkitehtuuri on (enimmäkseen) riippumaton toiminnallisuudesta
 - Saman toiminnallisuuden voi toteuttaa melkein minkälaisella arkkitehtuurilla tahansa
 - Mutta: huonosti valittu arkkitehtuuri voi tehdä toiminnallisuuden toteuttamisen ja laatuvaatimusten saavuttamisen vaikeaksi ja kalliiksi (tai mahdottomaksi)

3.9.2013

581358 Ohjelmistoarkkitehtuurit

27

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 eksperteiltä
 - Käsitteellisen eheyden (conceptual integrity¹) 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)

3.9.2013

581358 Ohjelmistoarkkitehtuurit

28

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.

3.9.2013

581358 Ohjelmistoarkkitehtuurit

29

Kuvaileva käyttö

- Rajoitteiden noudattamisen valvonta
 - Esim. elementtien välisten kiellettyjen riippuvuuksien tekeminen näkyviksi: kerrosarkkitehtuurissa alempi kerros kutsuu ylemmän kerroksen palvelua muuten kuin takaisinkutsurajapinnan kautta
- Organisaation kehittäminen
 - Vahvasti toisiinsa kytkeytyvien elementtien kehittämisvastuun jako kannattaa miettiä tarkkaan kommunikaatio-ongelmien välttämiseksi

3.9.2013

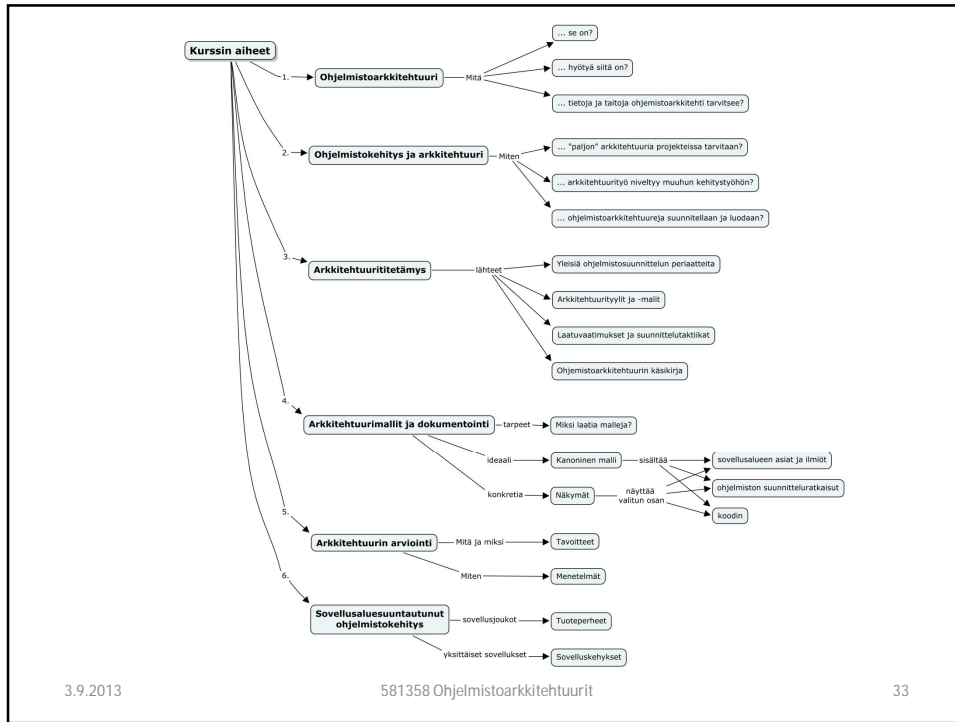
581358 Ohjelmistoarkkitehtuurit

30

Muita hyötyjä

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

KURSSIN SISÄLTÖ



3.9.2013

581358 Ohjelmistoarkkitehtuurit

33