

Ohjelmistoarkkitehtuurit

Arkkitehtuurin arviointi

Luento 13



16.10.2014

581385 Ohjelmistoarkkitehtuurit

1



Arkkitehtuurin arviointi (architectural analysis)

- Arkkitehtuurin arvioinnin tarkoituksena on muodostaa käsitys kehitettävän ohjelmiston tärkeistä ominaisuuksista
 - Nyt ja *tulevaisuudessa*
- Arvioinnin rooli korostuu *suunnitelmallisuutta vaativissa kehitysprojekteissa*, joihin liittyy huomattavia riskejä
 - Kaikki keinot käyttöön riskien minimoimiseksi
- Voi olla myös osa ohjelmiston laajempaa evaluointia
 - kun ollaan hankkimassa ohjelmistoratkaisua
 - kun olemassa oleva ohjelmisto tulee elinkaarensa johonkin käännekohtaan

16.10.2014

581385 Ohjelmistoarkkitehtuurit

2

Arkkitehtuurin arviointi (architectural analysis)



- Täsmällisemmin tavoitteina voi olla saavuttaa
 - aikainen arvio järjestelmän koosta, monimutkaisuudesta ja kustannuksista
 - käsitys ohjelmistoon kohdistuvien vaatimusten (sekä toiminnallisten että erityisesti laadullisten) toteutumisesta
 - käsitys suunnitteluohjeiden ja sääntöjen noudattamisesta
 - käsitys dokumenttien ja toteutuksen vastaavuudesta, jne...
- Arvioinnin kohteena
 - Komponentit ja niiden väliset kytkennät
 - Järjestelmä kokonaisuutena ja osajärjestelmät
 - Tiedot ja tiedonkulku osajärjestelmien välillä
 - Arkkitehtuurit
 - *Vaihtoehtoiset / suhde referenssiarkkitehtuuriin*

16.10.2014

581385 Ohjelmistoarkkitehtuurit

3

Arkkitehtuurin arviointi



- Arvioinnissa päähuomio on **laadullisissa vaatimuksissa**
- Jos jotain laadullista ominaisuutta ei voida arkkitehtuurin (kuvauksen) perusteella arvioida, arkkitehtuuri(kuvaus) on tältä osin puutteellinen
- **Poikkeuksia**
 - Käytettävyys riippuu pitkälti käyttöliittymän yksityiskohdista, jotka eivät näy arkkitehtuurikuvauksessa
 - Toteutustason tietorakenne- ja algoritmiratkaisut vaikuttavat suoritussykyyn
- Arkkitehtuurin on kuitenkin mahdollistettava laatuvaatimusten täyttyminen tunnettujen toteutustapojen puitteissa

16.10.2014

581385 Ohjelmistoarkkitehtuurit

4

Arkkitehtuurin arviointi



- Arvioitavat yleiset laatuvaatimukset on aina täsmennettävä ja priorisoitava
 - esim. muunneltavuus → minkä asioiden suhteen järjestelmän tulee olla muunneltava
 - esim. suorituskykyvaatimusten priorisointi keskenään ja suhteessa muunneltavuusvaatimuksiin
- Myös toiminnallisia vaatimuksia voidaan arvioida arkkitehtuurin pohjalta
 - Ovatko halutut toiminnot toteutettavissa arkkitehtuurin komponenttien välisellä vuorovaikutuksella?

16.10.2014

581385 Ohjelmistoarkkitehtuurit

5

Arkkitehtuurin arviointi



- Arviointi auttaa ennustamaan järjestelmän evoluutiota ja ylläpitokustannuksia
 - Hyödyllistä tietoa johdolle resursointia varten
- Arviointi auttaa suunnittelijoita ymmärtämään järjestelmää paremmin ja näkemään sen ongelmakohtia
 - Hyödyllistä myös jälkikäteen tehtynä

16.10.2014

581385 Ohjelmistoarkkitehtuurit

6

Arkkitehtuurin arviointi



- Arkkitehtuurin arviointi osana ohjelmistoprosessia
 - Pitäisi olla rutiininomainen osa normaalia ohjelmistokehitysprosessia (riskiarvion perusteella)
 - *vaativuudeltaan erilaisia arviointeja projektin ja riskien mukaan*
 - Arkkitehtuuri sisältää kriittisimmät suunnitteluratkaisut
 - *arviointi erityisen tärkeää*
 - Arkkitehtuurikuvaus on järjestelmän ensimmäinen arvioinnin mahdollistava kuvaus
 - *havaitut puutteet pystytään vielä korjaamaan suhteellisen vähillä kustannuksilla*
 - Arviointi on edullinen tapa testata arkkitehtuuri
 - *prototyypin rakentaminen on kalliimpaa (entä agile, Walking Skeleton?)*
 - *arkkitehtuurin testaus vasta järjestelmätestauksen yhteydessä on kallein tapa*

16.10.2014

581385 Ohjelmistoarkkitehtuurit

7

Arkkitehtuurin arviointi



- Arkkitehtuurin arvioinnin tulokset ovat aina suhteessa arkkitehtuurikuvauksen laatuun: jos kuvaus on epätarkka, myös arvioinnin tulokset ovat epätarkkoja
- Erilaisia arkkitehtuuriarviointeja
 - Arvioidaan heti **ensimmäinen arkkitehtuurin luonnos**
 - Arvioidaan **valmis arkkitehtuurikuvaus**
 - Arvioidaan **valmiin järjestelmän arkkitehtuuri**

16.10.2014

581385 Ohjelmistoarkkitehtuurit

8

Arkkitehtuurin arviointi



- Arvioidaan heti **ensimmäinen arkkitehtuurin luonnos**
 - Luonnokselle vain **pienimuotoinen** arviointi
 - Analysoidaan vaatimukset arkkitehtuurin kannalta: voiko mikään arkkitehtuuri täyttää vaatimukset
 - Tuloksena realistinen, priorisoitu vaatimusjoukko sekä alustava arkkitehtuuriratkaisu niiden täyttämiseksi
 - Alustavassa arkkitehtuurin arvioinnissa tulisi olla läsnä henkilöitä, joilla on valtuus määrätä vaatimuksista

16.10.2014

581385 Ohjelmistoarkkitehtuurit

9

Arkkitehtuurin arviointi



- Arvioidaan **valmis arkkitehtuurikuvaus**
 - Tähän kannattaa käyttää enemmän resursseja
 - Tulisi tapahtua **ennen kuin toteutus aloitetaan**, koska arvioinnin tuloksena arkkitehtuuri saattaa muuttua
 - *Ongelma: toteutus on resursoinnin ja aikataulun takia aloitettava usein jo arkkitehtuurityön aikana*
 - *Ongelma: BDUF*
 - *Osittaisarviointi*

16.10.2014

581385 Ohjelmistoarkkitehtuurit

10

Arkkitehtuurin arviointi



- Arvioidaan **valmiin järjestelmän arkkitehtuuri**
 - Voi olla järkevää, jos halutaan esim. lisätä tietoa järjestelmän laatuattribuuteista (esim. onko järjestelmä siirrettävissä uuteen ympäristöön)
 - Jos ei ole ajan tasalla olevaa arkkitehtuurikuvausta, sellainen voidaan tuottaa (osittain) automaattisesti takaisinmallinnustyökalujen avulla

16.10.2014

581385 Ohjelmistoarkkitehtuurit

11

Arkkitehtuurin arviointi



- **Kysymyksiä:**
 - **Sopiiko suunniteltu arkkitehtuuri järjestelmälle?**
 - *Järjestelmän (laatu)vaatimusten täyttyminen tietyllä arkkitehtuurilla*
 - *Jos ei sovi, niin vastauksen pitäisi kertoa myös miksi*
 - **Mikä vaihtoehtoisista arkkitehtuureista sopii parhaiten järjestelmälle ja miksi?**
 - *Arkkitehtuurille on esitetty useita vaihtoehtoja*
 - *Vastaus kertoo, mikä on paras ja miksi*
 - **Miten hyvä tulee olemaan järjestelmän jokin tietty laadullinen ominaisuus, olettaen järkevä toteutustapa?**
 - *Määrälliset arviot joistain laatuominaisuuksista*
 - *Vastaus voi esimerkiksi kertoa, kuinka kalliiksi järjestelmän ylläpito todennäköisesti tulee*

16.10.2014

581385 Ohjelmistoarkkitehtuurit

12

Arkkitehtuurin arviointi



- Arvioinnin perusongelma:
 - [tulevan järjestelmän tiettyyn laatuominaisuuteen vaikuttavien kohtien \(suunnittelupäätösten\) poimiminen arkkitehtuurin kuvauksesta](#)
- Asiantuntija-arvio on yleisin tapa tehdä arviointi:
 - tulos riippuu täysin arvioijan kokemuksesta ja taidoista
 - Siksi on kehitetty menetelmiä, jotka ainakin periaatteessa ovat yksilön mielipidettä objektiivisempia

16.10.2014

581385 Ohjelmistoarkkitehtuurit

13

Arkkitehtuurin arviointi



- Eri tyyppistä arviointia
 - Arkkitehtuuriin tai arkkitehtuurin suunnitteluprosessiin liittyvät [tarkistuslistat ja kysymyslomakkeet](#)
 - *esim. "onko käyttöliittymä erotettu sovelluslogiikasta?"*
 - *yleisiä tai sovellusaluekohtaisia*
 - *voidaan yhdistää skenaariopohjaisiin menetelmiin*
 - Mallipohjainen arviointi
 - *Edellyttää täsmällisiä yksityiskohtaisia malleja (tarkasteltavan ominaisuuden suhteen)*
 - *Sopii huonosti laajojen kokonaisuuksien arviointiin*
 - [Staattisiin ominaisuuksiin](#) perustuva mallipohjainen arviointi
 - *Sopii erityisesti sisäisten yksittäisten rakenteellisten piirteiden arviointiin: syntaktinen oikeellisuus, tyylin noudattaminen, lukkiutumattomuus, kuormitus, jne (tila-automaatit, jonoverkot, riippuvuusmatriisit, ...)*
 - [Dynaamisten ominaisuuksien](#) simulointipohjainen arviointi
 - *Sopii yksittäisten käyttäytymispiirteiden arviointiin*

16.10.2014

581385 Ohjelmistoarkkitehtuurit

14

Arkkitehtuurin arviointi



- Skenaarioihin perustuva arviointi
 - *Esitetään **konkreettisia** esimerkkitilanteita, joissa tietyt laatuominaisuudet tulevat esiin*
 - *Tutkitaan, miten arkkitehtuuri sopii kyseiseen skenaarioon*
 - *Etuna skenaarioiden suhteellisen helppo löytäminen, konkreettisuus ja ymmärrettävyys*

16.10.2014

581385 Ohjelmistoarkkitehtuurit

15

Arkkitehtuurin arviointi



- Skenaarion luonne riippuu sitä, mitä laatuominaisuutta halutaan tarkastella
 - *Jos tarkastellaan muutettavuutta, skenaario käsittelee jotakin muutostarvetta järjestelmän evoluutiossa*
 - *Jos tarkastellaan suorituskykyä, skenaario käsittelee jotakin suorituskykykriittistä tilannetta järjestelmän käytössä (tämä on lähellä käytötapausten käsitettä)*
 - *Jos tarkastellaan turvallisuutta, skenaario käsittelee jotakin uhkatilannetta järjestelmän käytössä*
 - *Jos tarkastellaan uudelleenkäytettävyyttä, skenaario käsittelee jonkin uuden sovelluksen tekemistä uudelleen käyttämällä järjestelmää (= tuoterunko)*
 - ...

16.10.2014

581385 Ohjelmistoarkkitehtuurit

16

Arkkitehtuurin arviointi



- Skenaariopohjaisia arviointimenetelmiä
 - SAAM (Software Architecture Analysis Method)
 - Varhaisimpia skenaariopohjaisia menetelmiä
 - Kehitetty SEI:ssä (Software Engineering Institute, Carnegie-Mellon University)
 - Tarkoitettu lähinnä muunneltavuuteen liittyvien laatuominaisuuksien arviointiin, mutta sillä voidaan arvioida myös toiminnallisuutta
 - ATAM (Architecture Tradeoff Analysis Method)
 - SAAM-menetelmästä edelleen SEI:ssä johdettu yleisempi menetelmä myös muihin laatuominaisuuksiin
 - MPM (Maintenance Prediction Method)
 - Jan Boschin kehittämä ylläpidettävyyteen keskittyvä arviointimenetelmä

16.10.2014

581385 Ohjelmistoarkkitehtuurit

17

ATAM



- Architectural trade-off analysis
 - Rick Kazman, Mark Klein, Paul Clements: **ATAM: Method for Architecture Evaluation**
(<http://www.sei.cmu.edu/reports/00tr004.pdf>)
- Ensisijaisesti riskien kartoitukseen kehityksen alkuvaiheessa
- Keskittyy laadullisiin ominaisuuksiin
- Määrittelee prosessin ja käytännöt
- Prosessi luonteeltaan formaalin katselmuksen kaltainen
 - Lähtökohtana (liike)toimintatavoitteet ja järjestelmän arkkitehtuuri
 - Suorittajana arviointiryhmä, jossa eri sidosryhmien edustaja

16.10.2014

581385 Ohjelmistoarkkitehtuurit

18

ATM-prosessi



- **Vaihe 1:**

- **Esittelysiosiossa** käydään läpi ATAM-menetelmä, järjestelmän arkkitehtuuri sekä liiketoimintatavoitteet
- **Analyysiosiossa**
 - etsitään laatuominaisuuksiin vaikuttavat arkkitehtuuriratkaisut
 - analysoidaan järjestelmän laatuvaatimukset
 - kuvataan laatuvaatimuksiin liittyvät skenaariot
 - identifioidaan skenaarioiden avulla arkkitehtuurin kriittiset kohdat

- **Vaihe 2:**

- **Testausosiossa** täydennetään skenaarioita järjestelmän käyttäjien näkökulmasta ja analysoidaan arkkitehtuuri uutta skenaariojoukkoa vasten
- **Raportointiosiossa** esitetään arvioinnin tulokset

16.10.2014

581385 Ohjelmistoarkkitehtuurit

19

ATAM-prosessi



- Esittely- ja analyysiosiot muodostavat menetelmän ensimmäisen vaiheen, johon osallistuvat arviointiryhmän lisäksi tuoteprojektin vastuuhenkilöt
- Testaus- ja raportointiosio muodostavat arvioinnin toisen vaiheen, johon osallistuu ensimmäisen vaiheen osallistujien lisäksi muiden sidosryhmien edustajia (esim. asiakkaan)
- Koko prosessi kestää noin kolme päivää
 - Kahdessa jaksossa (1+2), välissä parin viikon tauko
- Toisen vaiheen aluksi käytetään päivä ensimmäisen vaiheen tulosten kertaamiseen uusille osallistujille

16.10.2014

581385 Ohjelmistoarkkitehtuurit

20

ATAM-prosessi



- Esittelyosio:
 - (1) ATAM-menetelmän esittely (arvioinnin johtaja selostaa)
 - *Arviointiprosessi, roolitus, käytettävät tekniikat ja tulosten muoto*
 - *Tärkeää: kaikille realistiset odotukset prosessista*
 - (2) Järjestelmän olennaiset vaatimukset, liiketoiminta-tavoitteet ja toimintaympäristö (projektipäällikkö)
 - (3) Arkkitehtuuri, sen tekninen toimintaympäristö ja rajapinnat muihin järjestelmiin (pääarkkitehti)
 - *Käytetään näkymiä*
 - *Erityisesti arkkitehtuurin kannalta keskeiset näkökulmat*
 - *Tulisi olla esiteltävissä tunnissa*

16.10.2014

581385 Ohjelmistoarkkitehtuurit

21

ATAM-prosessi



- Analyysiosa:
- (1) Arkkitehtuuriratkaisujen ja niiden laatuvaikutusten tunnistus
 - arkkitehtuurityylit ja patternit listataan

16.10.2014

581385 Ohjelmistoarkkitehtuurit

22

ATAM-prosessi



- Analyysiosa:
- (2) Skenaarioiden määrittely
 - Eri tyyppisiä skenaarioita riippuen siitä, mitä laatuominaisuutta tarkastellaan
 - Käyttötapaukset
 - Kuinka käyttäjät käyttävät
 - Kasvuskenaariot
 - Kuvaavat suunniteltuja tai ennakoitavia muutoksia ohjelmistossa
 - Rajoja kartoittavat skenaariot
 - Etsivät järjestelmän rajoja: esim, kuormitusmuutoksia, alustan vaihto, palvelimen rikkoutuminen
 - Kaikkiin skenaarioihin voidaan liittää suoritusaika-/määrätavoitteita

16.10.2014

581385 Ohjelmistoarkkitehtuurit

23

ATAM-prosessi

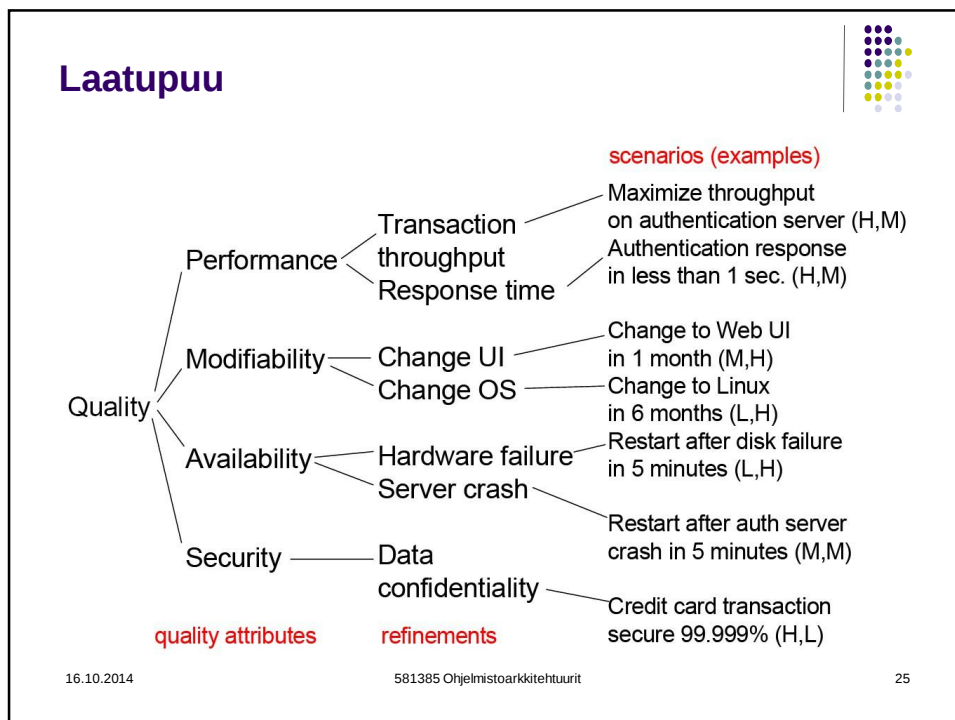


- **Laatupuun** (utility tree) laatiminen jäsentämään skenaarioita (ks. Seuraava kuva)
 - Puussa kuvataan **laatuattribuutit** ja niiden jako alikohtiin
- Lehdiksi **skenaarioita**, jotka testaavat kyseistä ominaisuutta
 - Skenaariot painotetaan **tärkeydellä ja vaikeudella**
 - Asteikkona esim (H=high, M=medium, L=low)
 - (painotus kyettävä perustelemaan)
 - Painopari esim (H,L)
- Laatupuun on järjestelmäkohtainen, vaikka samantyyppisillä järjestelmillä on yleensä samantyyppiset laatupuut
 - Eri järjestelmissä voidaan tarkastella eri osajoukkoja eri painoituksin (esim. ylläpidettävyyys vs. siirrettävyys)
 - painot puun haaroille

16.10.2014

581385 Ohjelmistoarkkitehtuurit

24



ATAM

- (3) Painotetun laatupuun & skenaarioiden liittäminen arkkitehtuuriratkaisuihin
 - Skenaarioihin liitetään niitä tukevat arkkitehtuuriratkaisut
 - *Tarkastelu aloitetaan tärkeimmistä ja vaikeimmista (H,H)-luokka, vähempiarvoiset saattavat jäädä vähäiselle käsittelylle*
 - *Arkkitehtuurityylit ja muut skenaarioon vaikuttavat ratkaisut*
 - Identifioidaan **riskit, turvalliset ratkaisut, herkkyyshkohdat** ja **tasapainottelukohdat** puun avulla
 - *Tunnistuksen perustana laatuattribuuttikohtaiset kysymykset*

16.10.2014 581385 Ohjelmistoarkkitehtuurit 26

ATAM



- *Riski (risk)*
 - Suunnittelupäätös tai arkkitehtuurin piirre, joka voi johtaa jonkin laatuominaisuuden huononemiseen
 - Kuvataan kertomalla kyseinen päätös/ominaisuus, siitä mahdollisesti aiheutuva laatuongelma ja ongelman syy
- *Turvallinen ratkaisu (non-risk)*
 - Järjestelmän laatuominaisuuksia parantava ja arkkitehtuurissa voimassa oleviin olettamuksiin perustuva päätös (jos arkkitehtuurioletukset muuttuvat, ratkaisu ei välttämättä pysy turvallisena)
 - Kuvataan kertomalla taustaoletukset, suunnittelupäätös, siitä seuraavat laatuodotukset sekä syyt näille seurauksille

16.10.2014

581385 Ohjelmistoarkkitehtuurit

27

ATAM



- *Herkkyyskohta (sensitivity point)*
 - Suunnittelupäätös tai arkkitehtuurin piirre, joka on kriittinen jonkin laatuominaisuuden kannalta
 - Jos tätä ominaisuutta muutetaan tai päätöksestä luovutaan, jokin laatuominaisuus on vaarassa huonontua
 - Miksi järjestelmä saavuttaa tietyn laatuominaisuuden?
 - Kuvataan kertomalla kyseinen suunnittelupäätös ja siitä riippuva laatuominaisuus

16.10.2014

581385 Ohjelmistoarkkitehtuurit

28

ATAM



- *Tasapainottelukohta (trade off point)*
 - Herkkyyiskohta, joka vaikuttaa useampaan kuin yhteen laatuominaisuuteen (esim. edullisesti johonkin ja epäedullisesti toiseen)
 - Esim. jonkin patternin soveltaminen voi lisätä muunneltavuutta, mutta heikentää suorituskkyä
 - Kuvataan selostamalla kyseinen suunnittelupäätös ja sen vaikutus laatuominaisuuksiin

16.10.2014

581385 Ohjelmistoarkkitehtuurit

29

ATAM



- Testausosio:
 - (1) Muut kiinnostusryhmät kuin arkkitehdit tai arviointiryhmä (esim. testaajat, ylläpitäjät, asiakkaat, johto) ideoivat skenaarioita omista lähtökohdistaan
 - *Esim. käyttötapauksia, odotettavissa olevia muutoksia, ...*
 - *Skenaariot priorisoidaan esim. äänestämällä*
 - *Tavoitteena on herättää keskustelua eri kiinnostusryhmien välille ja saavuttaa yhteinen näkemys laatuominaisuuksista*

16.10.2014

581385 Ohjelmistoarkkitehtuurit

30

ATAM



- Testausosio jatkuu:
 - (2) Skenaarioita verrataan jo löydettyihin skenaarioihin
 - *Jos nämä täsmäävät, kaikki on hyvin*
 - *Jos skenaariota ei löydy puusta, mutta sille on puussa sopiva laatuhaara, siitä tulee uusi lehti puuhun (tai useita, jos se liittyy useisiin laatuäkökohtiin)*
 - *Jos skenaariolle ei ole sopivaa haaraa puussa, se joko ei liity lainkaan laatuun tai se liittyy uuteen laatuominaisuuteen*
 - *Jälkimmäisessä tapauksessa puuhun lisätään uusi haara, johon skenaario liitetään lehdeksi (tällainen skenaario paljastaa yleensä järjestelmästä uusia riskejä)*

16.10.2014

581385 Ohjelmistoarkkitehtuurit

31

ATAM



- Raportointi:
 - ATAM-prosessin tulokset esitetään arvioinnin lopuksi koko arviointiin osallistuneelle ryhmälle ja laaditaan arviointiraportti
 - Raportti voidaan organisoida esim. skenaarioittain
 - *skenaariota tukevat arkkitehtuuriratkaisut*
 - *skenaarioon liittyvät riskikohdat, turvalliset ratkaisut, herkkyysskohdat ja tasapainokohdat (listataan erikseen)*
 - *laatuominaisuudet, joita skenaario testaa*
 - *kuvaus olosuhteista, joissa skenaario tapahtuu*

16.10.2014

581385 Ohjelmistoarkkitehtuurit

32

ATAM



- Raportointi:

- Saadaan tietoa, joka auttaa ymmärtämään arkkitehtuuria ja hallitsemaan sen riskejä: miten arkkitehtuuri liittyy laatuun
- Tuloksena voi joskus tulla myös parannusehdotuksia arkkitehtuuriin (ei kuitenkaan ole ATAM:in varsinainen tavoite)
- Arvioinnin tuloksiin kootaan myös nk. **riskiteemoja**
 - *riskien ryhmiä, joiden taustalla on sama yleinen syy*
 - *esim. tiedon häviäminen laitteisto- tai ohjelmistohäiriöissä → riskiteema "riittämätön tiedon varmistus"*
 - *Tarkoituksena on liittää riskit liiketoimintatavoitteisiin ja tehdä ne ymmärrettävämmiksi johdolle*

16.10.2014

581385 Ohjelmistoarkkitehtuurit

33

Arvioinnin ongelmia



- Arvioinnin raskaus pienille järjestelmille & organisaatioille
 - Voidaan tehdä osittainen tai kevennetty katselmointi
 - Tarkastuslista keskeisimmistä arkkitehtuurilta vaadittavista ominaisuuksista arvioinnin minimivaatimus
- Arvioinnin ajoitus
 - Aikaisessa vaiheessa arviointi voi tuntua tarpeettomalta, sillä järjestelmästä tiedetään suhteellisen vähän
 - Toteutusvaiheen alkaessa arviointi voidaan nähdä uhkana järjestelmän toteutusaikataululle (löydettyjen vikojen korjaaminen!)
 - Toteutuksen jälkeinen arviointi puolestaan voidaan nähdä turhana, koska järjestelmä on jo toteutettu ja toimii
 - Näistä syistä arviointi pitäisi kiinnittää johonkin ohjelmistoprosessin etappiin

16.10.2014

581385 Ohjelmistoarkkitehtuurit

34

Arvioinnin ongelmia



- Usein kuullun huomion mukaan arkkitehdit tietävät arkkitehtuurin ongelmista jo etukäteen, ja siksi arviointi voi vaikuttaa turhalta
- Tällöin korostuu arvioinnin rooli tiedon välityksen työkaluna
 - Arviointia voi esim. hyödyntää opetustilaisuutena kutsumalla arviointiin mukaan ohjelmistosuunnittelijat, jotka lopulta toteuttavat arkkitehtuurin mukaisen järjestelmän