

**PLUGIT-HANKKEEN SELVITYKSIÄ JA RAPORTTEJA 14**  
**STUDIES AND REPORTS OF THE PLUGIT PROJECT 14**

**Minna Porali, Annamari Riekkinen, Pentti Pohjolainen,  
Juha Mykkänen, Tanja Toroi, Tarja-Liisa Kärkkäinen, Anne Eerola**

**OHJELMISTOTUOTANNON NYKYTILASELVITYS  
2003 - KOHDERYHMÄNÄ TERVEYDENHUOLLON  
OHJELMISTOYRITYKSET JA ORGANISAATIOT**

**Tekijät:**

Minna Porali  
Pentti Pohjolainen  
Tanja Toroi  
Tarja-Liisa Kärkkäinen  
Anne Eerola  
Tietojenkäsittelytieteen laitos  
Kuopion yliopisto

Annamari Riekkinen  
Juha Mykkänen  
HIS-tutkimusyksikkö, Tietotekniikkakeskus  
Kuopion yliopisto

**Myynti:**

Myyjä  
Osoite  
Puhelinnumero  
www-osoite

ISBN XXX-XXX-XXX-X

PAINOTALON NIMI  
Kuopio 2004

## TIIVISTELMÄ

Valtakunnallisen Tekes-rahoitteen PlugIT-hankkeen yhteydessä tehtiin ohjelmistotuotannon nykytilaa selvittänyt kyselytutkimus. Kyselylomakkeita lähetettiin PlugIT:iin osallistuneille ohjelmistotaloille ja terveydenhuolto-organisaatioille sairaanhoitopiireittäin. Kysymykset ryhmiteltiin aihealueittain. Aihealueet olivat: tietojärjestelmän vaatimusmäärittely, ohjelmistotuotanto, tietojärjestelmän mallinnus, tekniikka ja toteutus, tietojärjestelmän käyttöönotto sekä testaus ja tarkastus. Kyselylomakkeet postitettiin vuoden 2002 lopulla. Tulokset käsiteltiin tilastollisesti vuoden 2003 alussa.

Tutkimuksen tuloksista käy selvästi ilmi, että vastaajat ovat hyvin perehtyneitä vaatimusmäärittelyyn. On myös nähtävillä, että tavoitteena on entisestään parantaa vaatimusmäärittelyä. Vaatimusmäärittelyn suurimpia ongelmia on yhteisen kielen ja käsitteistön löytäminen asiakkaan ja toimittajan välille. Kyselyyn vastanneet toivoivat määrittelyprosessiin yhtenäisiä työtapoja.

Ohjelmistotuotannon eri osa-alueet toteutetaan ja tullaan yhä useammin toteuttamaan projekteina. Mittareiden käyttö projektin seurannassa on harvinaista. Niiden käytön toivotaan lisääntyvän. Vastaajat näkivät ohjelmistotuotannon ongelmana esimerkiksi aikataulujen ennustettavuuden. Ohjelmistotuotannon tekniikan kehittymisen lähtökohtana ovat käytännön ongelmat, joihin tulisi löytää ratkaisu.

Laadullisten vaatimusten määrittelyn, komponenttikeskeisen suunnittelun ja oliokeskeisen analyysin käyttö tietojärjestelmän mallinnuksessa tulee lisääntymään. UML ja suunnittelumallit mallinnuksen apuna syrjäyttävät paperin ja kynän. Vastaajien mukaan suunniteltavasta ratkaisusta tärkeimpiä kuvattavia asioita tällä hetkellä ovat tietokannat, infrastruktuuri ja näytöt. Alisysteemiin jaon, arkkitehtuurin, toimintoketjujen ja komponenttien kuvaamisen toivotaan lisääntyvän.

Vastausten perusteella valmiskomponentteja käytetään joskus tai usein. Tavoitteena on lisätä niiden käyttöä. Integrointitapoja pitäisi kehittää ja integrointiteknologioiden käyttöä lisätä. Integrointiteknologioista käytetyimmät ovat XML-pohjainen integrointi, komponenttipohjainen integrointi sekä Web Service -tekniikka.

Terveydenhuoltoalalla työskentelevien mielestä käyttöönottoon liittyvissä asioissa ei olla riittävän hyvällä tasolla. Ohjelmistoyritysten mielestä ollaan paremmalla tasolla. Molempien mielestä käyttöönottoon liittyviin asioihin pitää kiinnittää enemmän huomiota. Käyttöönottovaiheessa ongelmia aiheuttaa muun muassa useamman toimittajan yhteistyö, epärealistinen aikataulu sekä koulutuksen riittämättömyys. Vastaajien parannusehdotuksia olivat muun muassa käyttäjän mukaanotto jo suunnitteluvaiheessa ja resurssien lisääminen.

Kyselystä kävi ilmi, että testauksessa ollaan siirtymässä kohti testausryhmän suorittamaa testausta. Työparin ohjelmien ja määrittelyjen testaaminen lisääntyy huomattavasti. Testauksen työkalut eivät ole yleisesti käytössä. Testauksen ja tarkastuksen ongelmana koettiin riittämättömät resurssit. Kehittämiskohteena mainittiin muun muassa määritysten testaamisen ja tarkastamisen lisääminen.

Kyselyn tulokset ovat suuntaa-antavia. Tuloksista saatua materiaalia on käytetty PlugIT-hankkeessa tehtyyn työhön.



# ESIPUHE

Ohjelmistotuotannon nykytilaselvitys on tehty PlugIT-hankkeessa vuosina 2002 - 2004. Selvityksen tavoitteena oli kartoittaa terveydenhuollon tietojärjestelmien vaatimusmäärittelyn, ohjelmistotuotannon ja ohjelmistojen käyttöönoton ongelmia sekä keskeisimpiä kehittämiskohteita. Tuloksia on hyödynnetty PlugIT-hankkeessa toiminnan suunnitteluun ja kehittämiseen. Tutkimuksen tulokset ovat antaneet arvokasta tietoa myös PlugIT:n seuraajahankkeiden suunnitteluun.

Ohjelmistotuotannon nykytilaselvitys toteutettiin kyselytutkimuksena. Kyselylomakkeita lähetettiin PlugIT:iin osallistuneille ohjelmistotaloille ja terveydenhuolto-organisaatioille sairaanhoitopiireittäin. Tarja-Liisa Kärkkäinen käsitteli palautetut lomakkeet tilastollisesti ja tuloksia esiteltiin puolivuotisseminaarissa. Koska ohjelmistoteollisuuden nykytilakyselyn tulokset ovat edelleen ajankohtaisia, katsoimme kuitenkin parhaaksi koota ne tähän yhtenäiseen julkaisuun.

PlugIT-hanketta ovat rahoittaneet ja siihen ovat osallistuneet TEKES, Mawell konserni, Medimaker Oy Ltd, Medici Data Oy, Mediweb Oy, Mylab Oy, Tietoenator Oyj, WM-data Novo Oyj, Atkos Oy, BEA Systems Oy, Commit Oy, Enfo Oy, Fujitsu Services Oy, General Electric Healthcare CIS EMEA, Mediconsult Oy, Microsoft Oy, Oracle Finland Oy, Helsingin ja Uudenmaan sairaanhoitopiirin kuntayhtymä, Pirkanmaan sairaanhoitopiirin kuntayhtymä, Pohjois-Savon sairaanhoitopiirin kuntayhtymä, Pohjois-Pohjanmaan sairaanhoitopiirin kuntayhtymä, Satakunnan sairaanhoitopiirin kuntayhtymä, Varsinais-Suomen sairaanhoitopiirin kuntayhtymä, Kuopion kaupungin sosiaali- ja terveyskeskus sekä Siilinjärven ja Maaningan terveydenhuollon kuntayhtymä.

Toivomme, että julkaisu antaa lukijalle tietoa niin ohjelmistoteollisuuden nykytilasta kuin tavoitetilastakin. Kiitämme kaikkia PlugIT-hankkeeseen osallistuneita ja erityisesti Teitä, jotka vastasitte kyselyyn.

Kuopiossa 30. elokuuta 2004

Anne Eerola ja Annamari Riekkinen



# SISÄLLYS

<b>1 JOHDANTO</b> .....	<b>9</b>
1.1 Taustaa ja tavoitteet.....	9
1.2 Rajausta ja toteuttaminen.....	9
1.3 Rakenne.....	9
<b>2 TIETOJÄRJESTELMÄN VAATIMUSMÄÄRITTELY</b> .....	<b>11</b>
2.1 Yleistä tietojärjestelmän vaatimusmäärittelystä.....	11
2.2 Kyselyn tulokset.....	11
2.3 Pohdintaa.....	19
<b>3 OHJELMISTOTUOTANTO</b> .....	<b>20</b>
3.1 Yleistä ohjelmistotuotannosta.....	20
3.2 Kyselyn tulokset.....	20
3.3 Pohdintaa.....	28
<b>4 TIETOJÄRJESTELMÄN MALLINNUS</b> .....	<b>29</b>
4.1 Yleistä tietojärjestelmän mallinnuksesta.....	29
4.2 Kyselyn tulokset.....	29
4.3 Pohdintaa.....	34
<b>5 TEKNIikka JA TOTEUTUS</b> .....	<b>35</b>
5.1 Yleistä tekniikasta ja toteutuksesta.....	35
5.2 Kyselyn tulokset.....	35
5.3 Pohdintaa.....	48
<b>6 TIETOJÄRJESTELMÄN KÄYTTÖÖNOTTO</b> .....	<b>49</b>
6.1 Yleistä tietojärjestelmän käyttöönotosta.....	49
6.2 Kyselyn tulokset.....	49
6.3 Pohdintaa.....	56
<b>7 TESTAUS JA TARKASTUS</b> .....	<b>57</b>
7.1 Yleistä testauksesta ja tarkastuksesta.....	57
7.2 Kyselyn tulokset.....	58
7.3 Pohdintaa.....	67
<b>LÄHTEET</b> .....	<b>68</b>



# 1 JOHDANTO

## 1.1 Taustaa ja tavoitteet

Valtakunnallisen Tekes-rahoitteen tutkimus- ja kehittämishankkeen PlugIT:n puitteissa tehtiin alkuvuodesta 2003 kysely, jonka tarkoituksena oli kartoittaa terveydenhuollon tietojärjestelmien vaatimusmäärittelyn, ohjelmistotuotannon ja ohjelmistojen käyttöönoton ongelmia sekä keskeisimpiä kehittämiskohteita. Kyselyn tuloksia on hyödynnetty PlugIT-hankkeessa toiminnan suunnitteluun ja kehittämiseen. Tässä dokumentissa esitellään kyselyn tulokset.

## 1.2 Rajaus ja toteuttaminen

Kyselyn kohderyhmänä oli kaikki PlugIT:ssa mukana olevat terveydenhuollon organisaatiot ja ohjelmistoyritykset. Terveydenhuollon organisaatioita oli yhteensä 16 kappaletta, joista 12 tietojärjestelmäyksikköä ja 4 käyttäjäorganisaatiota. Ohjelmistoyrityksiä oli mukana 13 kappaletta, joista kolme oli infrastruktuuritoimittajaa. Kysely toteutettiin lomakekyselynä. Kysymykset oli ryhmitelty aihealueittain. Lähetettyjen lomakkeiden ja saatujen vastausten määrät aihealueittain ryhmiteltyinä esitetään taulukossa 1.

**Taulukko 1: Lähetetyt lomakkeet ja saadut vastaukset**

	Lähetetyt lomakkeet (kpl)	Palautetut lomakkeet (kpl)
Tietojärjestelmän vaatimusmäärittely	64	11
Ohjelmistotuotanto	48	7
Tietojärjestelmän mallinnus	48	7
Tekniikka ja toteutus	55	10
Käyttöönotto	68	13
Testaus ja tarkastus	59	9

Jokainen aihealue sisälsi suljettuja ja avoimia kysymyksiä ja väittämiä ohjelmistotuotannossa käytössä olevista menetelmistä, tekniikoista ja välineistä. Vastaajien piti arvioida eri menettelytapojen tai väittämien tämän hetken tilannetta (= nykytila). Lisäksi vastaajien tuli merkitä taso, jossa he toivovat olevansa kolmen vuoden kuluttua (= tavoitetilä).

Käytetty asteikko oli: 0 = Ei koskaan, 1 = Harvoin, 2 = Joskus, 3 = Usein, 4 = Lähes aina, 5 = Aina.

## 1.3 Rakenne

Jokaisen luvun alussa kerrotaan lyhyesti yleistä tietoa aihealueesta. Seuraavaksi esitetään kyselyn tulokset omassa kappaleessaan. Mukaan on otettu kuvia selventämään tuloksia. Kuvissa olevat nuolet osoittavat niitä vaihtoehtoja, joissa vastaajien mielestä on eniten kehitettävää eli nyky- ja tavoitetilän välinen ero on huomattava. Jokaisen luvun lopussa on Pohdintaa-kappale, jossa kootaan aihealueen tulokset yhteen ja mietitään mahdollisia kehityskohtia.

Luvussa 2 esitetään tulokset tietojärjestelmän vaatimusmäärittelyn osalta. Ohjelmistotuotantoon liittyviin kysymyksiin saadut vastaukset esitetään luvussa 3. Luku 4 sisältää tietojärjestelmän mallinnuksen tulokset. Luvussa 5 esitetään tulokset liittyen tekniikkaan ja toteutukseen. Luvussa 6 esitetään tulokset käyttöönoton osalta. Seitsemännessä luvussa tarkastellaan testaus ja tarkastus -osion tuloksia.

Henkilöiden, jotka eivät osallistu ohjelmistojen rakentamiseen kannattaa erityisesti lukea luvut 2 ja 6. Vaatimusmäärittely ja käyttöönotto ovat ohjelmistotuotannon ne alueet joihin tietojärjestelmän käyttäjä useimmiten osallistuu. Myös testaus ja tarkastus -osio (luku 7) soveltuu hyvin heidän luettavakseen.

Yhteenvedon kyselyn tuloksista saa lukemalla pohdintaa-osiot. Jos haluaa saada tuloksista kattavamman kuvan, kannattaa lukea myös varsinaiset tulokset.

# 2 TIETOJÄRJESTELMÄN VAATIMUSMÄÄRITTELY

## 2.1 Yleistä tietojärjestelmän vaatimusmäärittelystä

Ennen uuden järjestelmän toteutusta pitää olla selvillä mitä ollaan lähdössä tekemään. Pitää tietää tuotteen tarkoitus, merkitys ja mitä tuotteen tulee tehdä, jotta se täyttää vaatimukset. Tuotteelle asetetut vaatimukset tulee tiedostaa ennen kuin tuotetta aletaan tarkemmin suunnitella. Robertson määrittelee vaatimuksen seuraavasti: *vaatimus* on jotain, jota tuotteen pitää tehdä tai ominaisuus, joka siinä pitää olla. Vaatimus on olemassa, koska tietyn tyyppinen tuote vaatii tietyt toiminnot tai ominaisuudet tai koska asiakas esittää jonkun vaatimuksen toimitettavalle tuotteelle (Robertson & Robertson, 1999). Vaatimusmäärittelyyn tulee osallistua sekä ohjelmistotuotannon ammattilaisia että ohjelman tulevia käyttäjiä. Ohjelmistoalan ammattilaisten ja käyttäjän tieto-taito täydentävät hyvin toisiaan.

Vaatimusmäärittelyvaiheen tarkoituksena on löytää ja koota tuotteen asiakasvaatimukset yhteen dokumenttiin. Vaatimuksia saadaan esimerkiksi havainnoimalla työtä, jota tuotteen tuleva käyttäjä tekee, haastatteleamalla käyttäjää, aivoriihessä, vanhoihin dokumentteihin tutustumalla ja neuvotteluissa. Vaatimusten kuvaamisessa voidaan käyttää apuna valmista pohjaa, jossa on määritelty kaikki vaatimuksesta kirjattavat tiedot. Hyvin dokumentoidut vaatimukset helpottavat ylläpitovaihetta ja mahdollistavat vaatimusten uudelleen käytön.

Ongelmana vaatimusmäärittelyssä voi olla muun muassa oikeiden vaatimusten löytäminen, yhteisen kielen löytäminen asiakkaan ja tuotteen toimittajan välille ja riskienhallinta. Ohjelmistoprojektien riskilistoilta löytyvät usein virheelliset ja puutteelliset asiakasvaatimukset, tästä syystä asiakasvaatimusten kartoittamisella on suuri merkitys. Myöhemmissä vaiheissa tapahtuva puutteellisen tai virheellisen määrittelyvaiheen paljastuminen tulee erittäin kalliiksi.

## 2.2 Kyselyn tulokset

Tietojärjestelmän vaatimusmäärittelyyn liittyvissä kysymyksissä vastausfrekvenssi vaihteli kolmen ja yhdentoista välillä. Kaikki vastaajat eivät merkinneet tavoitetasoa. Vastauksista kahdeksan oli ohjelmistoyrityksistä. Sairaalan atk-osastolta, erikoissairaanhoidosta ja perusterveydenhuollosta vastauksia oli yksi kustakin yksiköstä. Seuraavana esitetään kyselyn tulokset osa-alueittain.

Käytetty asteikko oli: 0 = Ei koskaan, 1 = Harvoin, 2 = Joskus, 3 = Usein, 4 = Lähes aina, 5 = Aina. Kuvissa olevat nuolet osoittavat niitä vaihtoehtoja, joissa vastaajien mielestä on eniten kehitettävää eli nyky- ja tavoitetilan välinen ero on huomattava.

### **2.2.1 Syyt tietojärjestelmän vaatimusmäärittelyn tekemiseen**

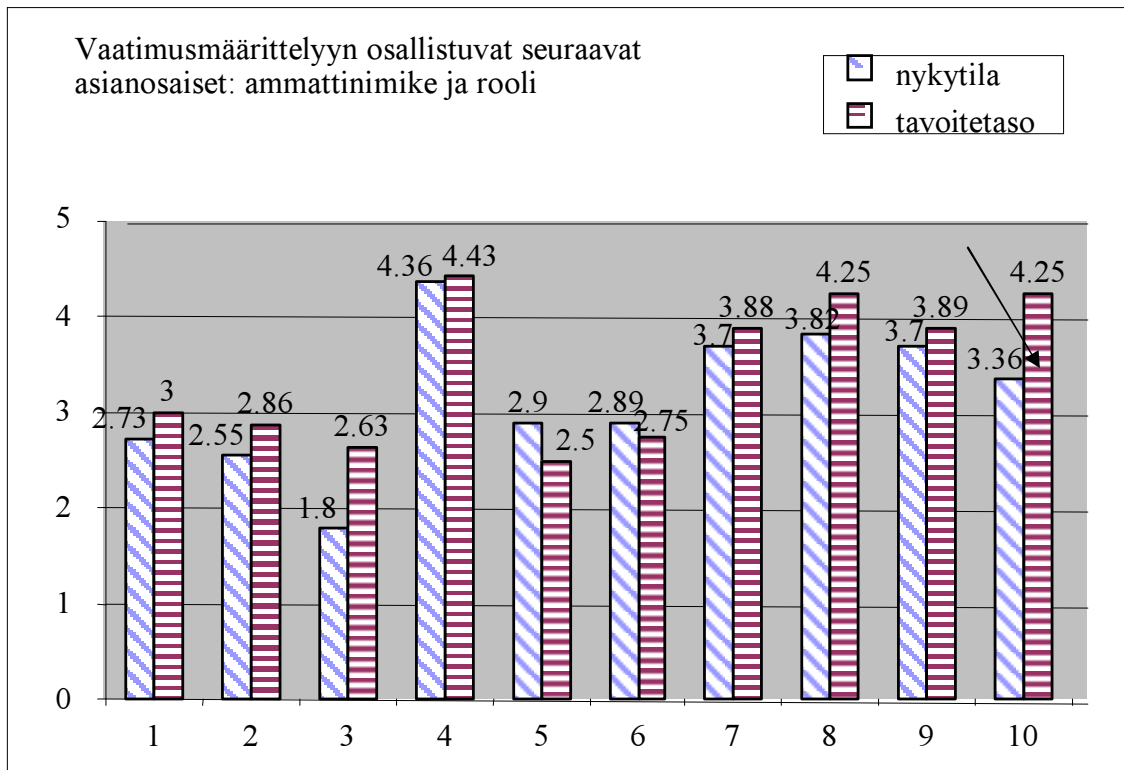
Kyselyn mukaan vaatimusmäärittelyä tehdään useimmiten silloin, kun vaatimuksia ilmaantuu. Vaatimusmäärittelyn tekeminen tarvekartoituksen perusteella on myös yleistä. Sen sijaan, ainoastaan joskus vaatimusmäärittelyä tehdään asiakkaalta tulevan tarjouspyynnön perusteella. Tavoitetilassa korostuu vaatimusmäärittelyn tekeminen enenevässä määrin tarvekartoituksen perusteella sekä asiakkaalta tulevan tarjouspyynnön perusteella. Vaatimusten ilmaantuessa tehtävä vaatimusmäärittely on nykyistä vähäisempää. Kukaan vastanneista ei maininnut mitään muuta syytä vaatimusmäärittelyn tekemiseen.

### **2.2.2 Vaatimusmäärittelyn toteutus**

Kyselyn tulosten perusteella vaatimusmäärittely toteutetaan lähes aina projektina. Ohjelmistojen vaatimukset priorisoidaan useimmiten asiakasryhmässä. Asiakasryhmässä tapahtuva ohjelmistojen vaatimusten priorisointi nähdään myös tavoitetilassa käytetyimpänä keinona. Sen sijaan tavoitteena on, että ohjelmistojen vaatimusten määrittely pelkästään tietojenkäsittelyn ammattilaisten suorittamana sekä yksittäisen asiakkaan kanssa vähenee nykyiseltä tasolta. Työvälineiden käyttö vaatimusmäärittelyssä on kyselyn perusteella harvinaista. Käytössä ei juuri ole itse tehtyjä vaatimusmäärittelyohjelmistoja. Käytössä olevista ostetuista ohjelmista mainittiin Rational Rose ja Caliber. Tavoitteena on hankittujen ohjelmien käytön selvä lisääminen.

### **2.2.3 Vaatimusmäärittelyyn osallistujat**

Kuvassa 1 on esitetty eri asianosaisten osallistuminen vaatimusmäärittelyyn. Tietojärjestelmän käyttäjät osallistuvat usein vaatimusmäärittelyyn. Tavoitetilassa heidän osallistumistaan vaatimusmäärittelyyn tulee edelleen lisätä. Ohjelmistosuunnittelija osallistuu lähes aina vaatimusmäärittelyyn ja myös tavoitetilassa hänellä on tärkeä rooli siinä. Sekä asiakkaan että toimittajan projektipäällikön osallistuminen vaatimusmäärittelyyn koetaan tärkeänä asiana. Ohjelmoijan ja ylläpitäjän rooli vaatimusmäärittelyssä näyttää kyselyn perusteella vähenevän nykyiseltä tasolta. Ohjelmoijan kohdalla tämä on ymmärrettävää, mutta ylläpitäjän kohdalla asia herättää kummastusta. Ylläpitäjällä on kuitenkin arvokasta tietoa ohjelmiston kehityshistoriasta, joten ainakin ohjelmistouudistuksissa hänen tiedoillaan on varmasti käyttöä. Yhden vastaajan mielestä informaation tulisi tavoitetilassa aina osallistua vaatimusmäärittelyyn.

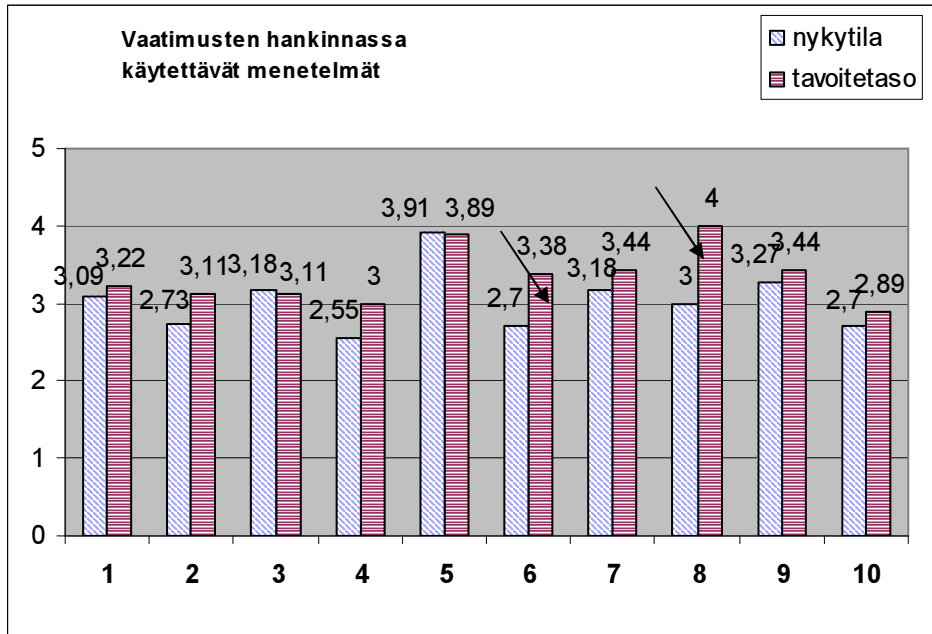


Kuva 1. Vaativuusmäärittelyyn osallistuvat asianosaiset

- |                            |                                  |
|----------------------------|----------------------------------|
| 1. Lääkäri                 | 6. Ylläpitäjä                    |
| 2. Sairaanhoidaja          | 7. Asiakkaan projektipäällikkö   |
| 3. Osastosihteeri          | 8. Toimittajan projektipäällikkö |
| 4. Ohjelmistosuunnittelija | 9. Pääkäyttäjä                   |
| 5. Ohjelmoija              | 10. Käyttäjät                    |

## 2.2.4 Vaativuusmäärittelyssä käytettävät menetelmät

Vaihtelu vaativuusmäärittelyssä käytettävien menetelmien välillä oli vähäistä. Jokaista vaihtoehtona annettua menetelmää käytetään vähintään usein. Nykytilassa ykkössija oli neuvotteluilla. Myös tavoitetilassa neuvotteluilla on tärkeä asema. Neuvotteluja suosittuimmaksi menetelmäksi näyttää vastausten perusteella nousevan aivoriihen käyttö vaativuusmäärittelyssä. Tavoitteena on myös protokollan käytön lisääminen. Kuvassa 2 on esitetty graafisesti eri menetelmien käyttö. Kukaan vastaajista ei ilmoittanut käyttävänsä mitään muuta menetelmää vaativuusmäärittelyssä.

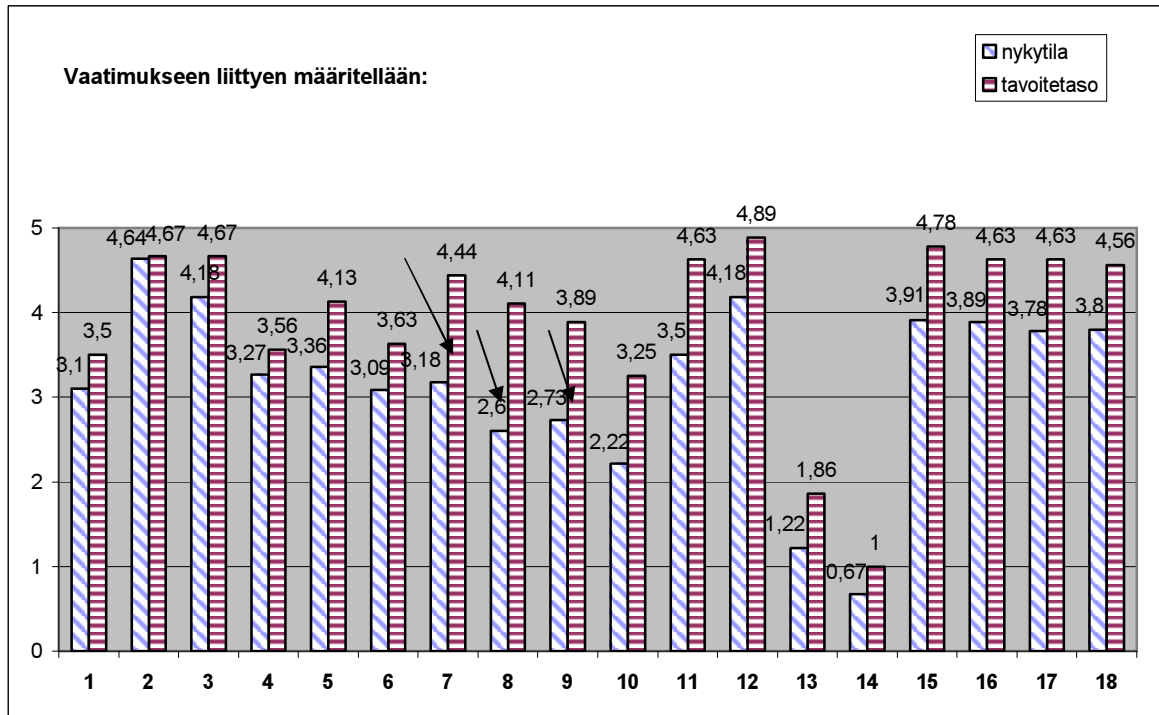


Kuva 2. Vaativuusten hankinnassa käytettävät menetelmät

1. Haastattelut
2. Dokumentteihin tutustuminen
3. Kyselyjen tekeminen
4. Havainnointi
5. Neuvottelut
6. Prototyyppien tekeminen
7. Oman organisaation asiantuntijat
8. Aivoriihi
9. Asiakaspalautteet
10. Muut vastaavanlaiset tuotteet

### 2.2.5 Vaativuksesta määriteltävät asiat

Vaativuksesta määriteltäviä asioita on lukuisia. Kuvassa 3 on esitetty vastaajien näkemykset asioista ja ominaisuuksista, joita vaativukseen liittyen määritellään. Kuvassa kiinnittää huomion se, että jokaiseen määriteltävään asiaan toivotaan tavoitetilassa kiinnitettävän enemmän huomioita. "Joskus"-tasolta ollaan siirtymässä selvästi "lähes aina" -tasolle ja jopa "aina"-tasolle asti. Tällä hetkellä useimmiten määriteltävät ominaisuudet ovat: tarvittavat toiminnot, tallennettavat tiedot ja tietoturva. Vastaajien mukaan useimmiten määriteltävät ominaisuudet tavoitetilassa ovat: tietoturva, laillisuus, tarvittavat toiminnot ja tallennettavat tiedot. Myös vaativuksen laillisuus ja projektin, systeemin sekä järjestelmäympäristön asettamat rajoitteet koettiin tärkeiksi. Vaativuusten määrittelyssä keskeisimmät kehittämistarpeet liittyvät operoitavuuden, suorituskyvyn ja ylläpidettävyyden kuvaamiseen (ks. kuvan 3 nuolet). Vähiten tärkeinä määriteltävinä asioina nähtiin kulttuuriset ja poliittiset tekijät, mikä on ohjelmistoteollisuuden kansainvälistymisen kannalta yllättävää. Kukaan vastaajista ei maininnut, että vaativukseen liittyen määritellään jotain muuta.

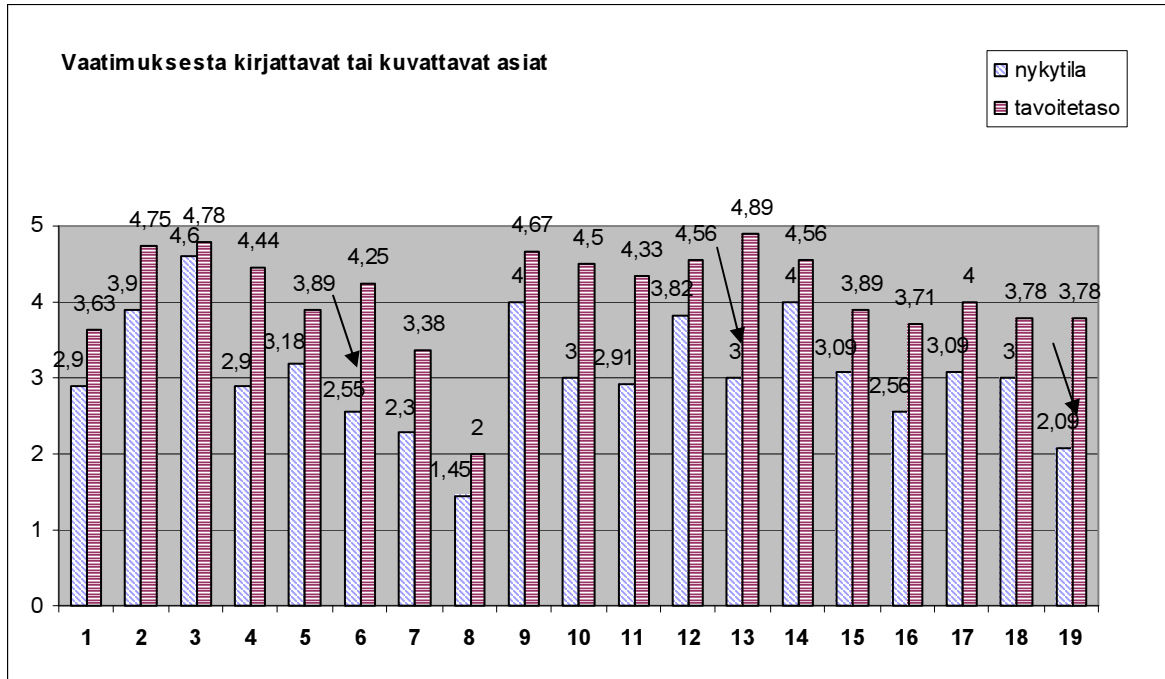


Kuva 3. Vaatimuksesta määriteltävät asiat

- |                            |  |
|----------------------------|--|
| 1. Vaikutusalue (scope)    | 11. Turvallisuus                               |
| 2. Tarvittavat toiminnot   | 12. Tietoturva                                 |
| 3. Tallennettavat tiedot   | 13. Kulttuuriset tekijät                       |
| 4. Tilapäiset tiedot       | 14. Poliittiset tekijät                        |
| 5. Käytettävyys            | 15. Laillisuus                                 |
| 6. Look and feel (tuntuma) | 16. Projektin asettamat rajoitteet             |
| 7. Suorituskyky            | 17. Systemin asettamat rajoitteet              |
| 8. Operoitavuus            | 18. Järjestelmäympäristön asettamat rajoitteet |
| 9. Ylläpidettävyys         |  |
| 10. Siirrettävyys          |  |

## 2.2.6 Vaatimuksesta kirjattavat tai kuvattavat asiat

Vaatimuksista kirjattavia tai kuvattavia asioita on myös lukuisia. Kuvassa 4 on esitetty kyselyssä mukana olleiden näkemyksiä eri asioiden kirjaamis- ja kuvaamistiheydestä. On selvästi nähtävissä, että tavoitetilassa tavoitteena on lähes jokaisen asian kohdalla päästä vähintään "lähes aina" -tasolle. Tällä hetkellä useimmiten kirjattavat asiat ovat: sanallinen kuvaus, tärkeys ja kustannusarvio. Tavoitetilassa tärkeimpiä asioita kyselyn mukaan ovat: toteuttamiseen liittyvät riskit, sanallinen kuvaus ja tavoite sekä tarkoitus. Tärkeinä asioina nähdään myös vaatimuksen perustelu, vaatimuksen yhteydet muihin vaatimuksiin sekä vaatimuksen muutoshistoria. Vaatimusten kuvaamisessa eniten kehittämistä näyttää olevan toteuttamiseen liittyvien riskien, maksajan (omistajan) sekä vaatimuksen toteutuksen mukanaan tuomien uusien ongelmien kirjaamisessa ja kuvaamisessa (ks. kuvan 4 nuolet). Vähiten kirjattu asia on vaatimuksen muut asianosaiset. Kukaan vastaajista ei nimennyt muita vaatimuksesta kuvattavia asioita.

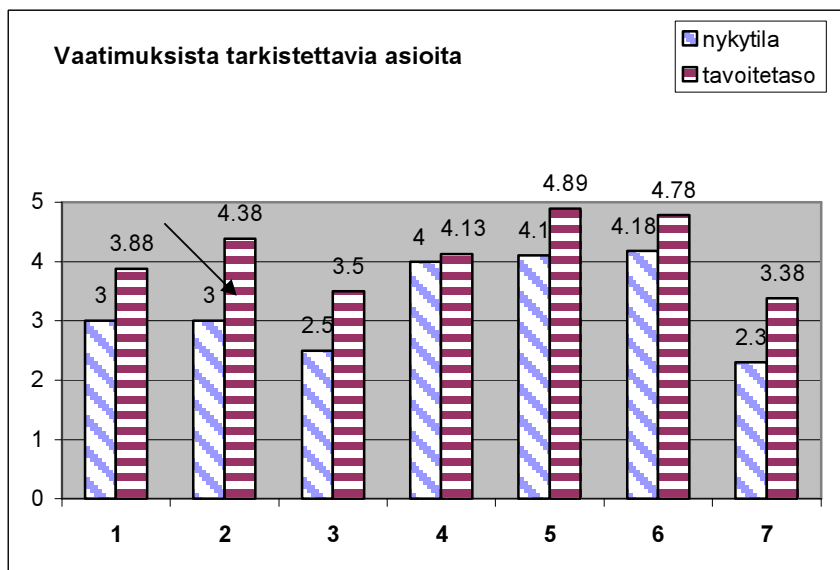


Kuva 4. Vaatimuksesta kirjattavat tai kuvattavat asiat

- |   |   |
|---|---|
| 1. Yksilöivä tunnus, esimerkiksi numero | 12. Aikatauluvaatimus   |
| 2. Tavoite, tarkoitus                   | 13. Toteuttamiseen liittyvät riskit                             |
| 3. Sanallinen kuvaus                    | 14. Kustannusarvio  |
| 4. Vaatimuksen perustelu                | 15. Avoimet kysymykset  |
| 5. Asiakas                              | 16. Vaikutukset konversioon                                     |
| 6. Maksaja (omistaja)                   | 17. Valmiiden osien<br>käyttömahdollisuus                       |
| 7. Loppukäyttäjät                       | 18. Odottamaan jätetyt vaatimukset                              |
| 8. Muut asianosaiset                    | 19. Vaatimuksen toteutuksen<br>mukanaan tuomat (uudet) ongelmat |
| 9. Tärkeys, prioriteetti                |   |
| 10. Yhteydet muihin vaatimuksiin        |   |
| 11. Muutoshistoria                      |   |

### 2.2.7 Vaatimuksista tarkistettavia asioita

Vaatimuksista useimmiten tarkistettavat asiat tänä päivänä kyselyn mukaan ovat: taloudellinen toteutettavuus, tekninen toteutettavuus ja tärkeys. Vastausten perusteella tekninen ja taloudellinen toteutettavuus ovat myös tavoitetilassa tärkeimmät vaatimuksista tarkistettavat asiat. Kolmannella sijalla tavoitetilassa on oikeellisuus, jonka tarkastamisessa on vastaajien mukaan myös eniten parannettavaa. Kuvassa 5 on selvästi nähtävissä, että tavoitetilassa jokaista asiaa tulisi tarkistaa nykyistä useammin. Vastaajista kukaan ei maininnut muita vaatimuksista tarkastettavia asioita.



Kuva 5. Vaatimuksista tarkistettavia asioita

1. Ristiriidattomuus
2. Oikeellisuus
3. Täydellisyys
4. Tärkeys
5. Tekninen toteutettavuus
6. Taloudellinen toteutettavuus
7. Inhimilliset näkökulmat

## 2.2.8 Vaatimustenhallinta

Vaatimusmäärittelyvaiheen jälkeen vaatimuksia muutetaan tai niitä voi tulla lisää. Kyselyn mukaan on yleistä, että määrittelyvaiheen jälkeen 10 - 20 % vaatimuksista muuttuu. Joskus jopa puolet vaatimuksista voi muuttua. Muutokset 70 %:iin vaatimuksista tai siitä ylöspäin ovat yksittäistapauksia. Tavoitteena on vähentää vaatimuksiin tapahtuvia muutoksia 10 %:n tasolle. Vaatimusten muuttaminen on kyselyn mukaan useimmiten hallittu tapahtuma ja vastaajien mukaan sen tulisi tavoitetilassa olla aina hallittua. Nykyään vaatimuksia hallitaan kirjaamalla vaatimukset erilliseen dokumenttiin sekä kiinnittämällä huomiota jäljitettävyyteen. Vastausten perusteella jäljitettävyyteen tullaan tavoitetilassa panostamaan nykyistä enemmän.

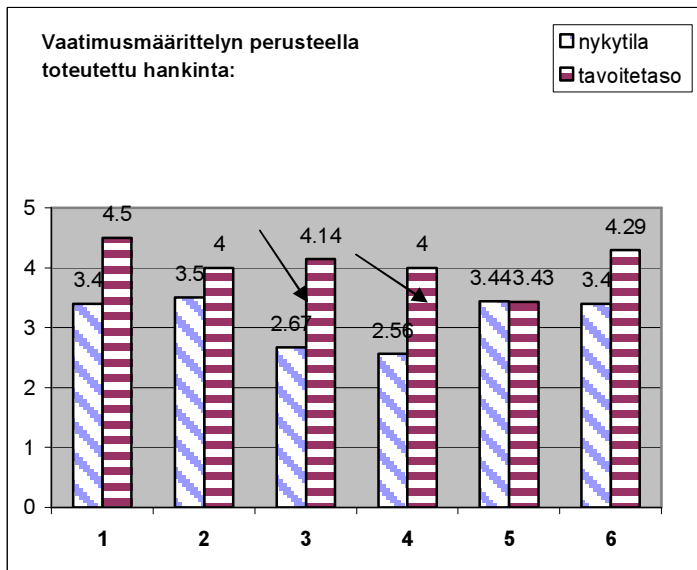
## 2.2.9 Riskianalyysi

Vastaajien mukaan riskejä arvioidaan usein. Tavoitetilassa riskien arviointia suoritetaan lähes aina. Torjuntasuunnitelman suunnittelu, riskien toteutumisen todennäköisyyden arviointi ja vaatimusmäärittelyn luotettavuuden vahvistaminen protoja toteuttamalla eivät tänä päivänä ole yleisesti käytössä. Sekä torjuntasuunnitelman suunnittelua että riskien toteutumisen todennäköisyyden arviointia tulisi tehdä nykyistä useammin. Vaatimusmäärittelyn luotettavuuden vahvistaminen protoja toteuttamalla toivotaan olevan usein käytetty keino riskienhallinnassa.

## 2.2.10 Vaatimusmäärittelyn perusteella toteutettu hankinta

Kyselyn mukaan tänä päivänä vaatimusmäärittelyn perusteella toteutettu hankinta perustuu useimpien kustannusten ja hyötyjen arviointiin. Kun hankinta perustuu vaatimusmäärittelyyn, ovat kustannukset usein mitattavia. Tavoitetilassa toivotaan, että vaatimusmäärittelyn perusteella toteutettu hankinta perustuu lähes aina hyötyjen arviointiin. Vaatimuksista katsotaan lähes aina syntyvän sopimus. Tavoitetilassa korostuu nykyistä enemmän investointilaskelmiin perustuva hankinta. Hyötyjen tulisi myös olla mitattavia, jos hankinta toteutetaan vaatimusmäärittelyn perusteella. (Kuva 6)

Hankinnan hyötyjä ja kustannuksia seurataan useimmiten hankintaprosessin aikana ja prosessin päättyessä. Kyselyn mukaan tavoitetilassa seuranta edelleen yleistyy. Hankintaprosessin päättymisen jälkeen tapahtuu seuranta, jonka toivotaan myös yleistyvän.



Kuva 6. Vaatimusmäärittelyn perusteella toteutettu hankinta

1. Perustuu hyötyjen arviointiin
2. Perustuu kustannusten arviointiin
3. Perustuu investointilaskelmiin
4. Hyödyt ovat mitattavia
5. Kustannukset ovat mitattavia
6. Vaatimuksista syntyy sopimus

## 2.3 Pohdintaa

Kyselystä käy selvästi ilmi, että vastaajat ovat hyvin perehtyneitä vaatimusmäärittelyyn. On myös nähtävillä, että tavoitteena on entisestään parantaa vaatimusmäärittelyä. Varsinkin vaatimukseen liittyvien asioiden määrittelyyn ja vaatimuksista kirjattaviin asioihin kiinnitetään tavoitetilassa entistä enemmän huomiota. Uuden teknologian tuomia mahdollisuuksia tulee myös hyödyntää entistä enemmän.

Vaatimusten hankinnassa käytettävistä menetelmistä muiden vastaavanlaisten tuotteiden osuus on hyvä, mutta toisaalta mietityttää eikö kannata vielä enemmän hyödyntää jonkun jo kerran tekemää työtä? Tätä samaa mietti myös eräs vastaajista. Voidaan säästää paljon aikaa ottamalla selvää, onko joku aikaisemmin jo määrittellyt vastaavanlaisia vaatimuksia.

Vaatimustenhallinnalla on tärkeä merkitys koko ohjelmiston kehittämisen aikana. Vaatimukset muuttuvat projektin aikana ja niitä tulisi voida muuttaa hallitusti. Kyselyn perusteella vastaajat näkevät myös tämän tärkeänä asiana. Vaatimusten jäljitettävyyden ei tällä hetkellä ole kovin yleistä, mutta näyttää siltä, että siihen aletaan panostaa enemmän. Jäljitettävyyden avulla voidaan todeta se, että tuote vastaa asiakkaan vaatimuksia (ks. Myöhänen, 2002). Yksi vastaajista mainitsi asiakkaan muutuvat tarpeet yhtenä vaatimusmäärittelyn suurimmista ongelmista.

Kyselyn mukaan, riskienhallintaan tullaan tavoitetilassa kiinnittämään huomiota nykyistä enemmän, mikä varmasti on hyvä suuntaus.

Käyttäjät otetaan tänä päivänä kiitettävästi mukaan vaatimusmäärittelyyn, tavoitetilassa heidän roolinsa näyttää muuttuvan yhä tärkeämmäksi. Vaikka asiakas on mukana vaatimusmäärittelyssä, vaikuttaa siltä, ainakin kyselyn mukaan, että yhteisen kielen löytäminen asiakkaan ja toimittajan välille ja se, ettei asiakas tunne tuotetta eikä vaatimusmäärittelyn tarpeellisuutta ovat vaatimusmäärittelyn suurimpia ongelmia. Ongelmana nähdään myös eri sairaaloiden erilaiset toimintatavat. Jokainen pitää tiukasti kiinni omasta toimintatavastaan.

Terveydenhuoltoalan työntekijät eivät välttämättä ole tottuneet käyttämään tietokonetta ja eivät ehkä ymmärrä teknologian tuomia ratkaisuja tai eivät halua ymmärtää. Varsinkin, kun on kyseessä esimerkiksi vuodeosastolla käytössä oleva hoidon toteutumisen seurantajärjestelmä. Ohjelman käytön katsotaan yleensä olevan liian hankalaa ja aikaa vievää. Tässä tullaankin siihen asiaan, jolla loppukäyttäjä voitaisiin saada ymmärtämään vaatimusmäärittelyn tarpeellisuus. Osallistumalla vaatimusmäärittelyyn loppukäyttäjä pystyy vaikuttamaan tuotteeseen.

Kyselyyn vastanneet toivoivat määrittelyprosessiin ja dokumentointiin helppoa ja yksinkertaista työkalua, jota kaikki osapuolet käyttäisivät.

# 3 OHJELMISTOTUOTANTO

## 3.1 Yleistä ohjelmistotuotannosta

Ohjelmistotuotantoon kuuluu ohjelmistojen tuotantoprosessit, niiden hallitseminen ja niissä käytettävät menetelmät. Ohjelmistotuotannon pyrkimyksenä on sovitun aikataulun puitteissa saada aikaan asiakkaan toiveiden mukainen luotettavasti toimiva järjestelmä. Yrityksillä on yleensä oma laatujärjestelmä, joka määrittelee yrityksen toimintatavat. Ohjelmiston kehitystyön jakaminen eri vaiheisiin tapahtuu käyttämällä jotain vaihejakomallia. Vaihejakomalleja ovat esimerkiksi vesiputousmalli, spiraalimalli, v-malli, protoilumallit ja inkrementaalinen malli, jolla tarkoitetaan ohjelmiston vähittäistä lisäämistä.

Ohjelmistotuotannon tyypillisiä projekteja ovat: määrittely-, suunnittelu-, ohjelmointi-, testaus-, käyttöönotto- ja ylläpitoprojekti. Projektista on hyvä laatia projektisuunnitelma. Suunnitelmassa kuvataan muun muassa projektin aikataulu, projektiorganisaatio, toteutusvälineet ja projektiin liittyvät riskit. Projektisuunnitelma on tärkeä väline projektin ohjauksessa ja aikataulun ja resurssien käytön seurannassa. Projektin seuranta voidaan toteuttaa ennalta määrättyllä tavalla. Tapoja ovat esimerkiksi raportointi, ajankäytön seuranta, projektipalaverit ja katselmukset.

Projektiin käytettävän työmäärän arviointi on yleensä vaikeaa. Usein työmäärää arvioidaan vertaamalla projektia aikaisempiin samantapaisiin projekteihin. Projektin paloittelu mahdollisimman pieniin osakokonaisuuksiin helpottaa arviointia. Työmäärien arvioinnissa voidaan käyttää menetelmiä, jotka perustuvat historiatietojen systemaattiseen käyttöön.

Mittaamisen avulla saadaan tietoa tuotteen ja prosessin toimivuudesta sekä tavoitteiden saavuttamisesta ja tuotteen ja prosessin kehittämiskohteista. Mittaamisen perustana ovat mittaustavoitteet ja sopivien mittareiden löytäminen. Mittareita ovat esimerkiksi asiat, jotka koetaan hankaliksi ja asiat, joista kerätään mittausta mahdollistavaa dataa. Sopivia mittareita ovat muun muassa projektien aika- ja kustannusarvioiden toteutuminen, työn tuottavuus sekä ohjelmiston koko. Sopivien mittareiden löytäminen voi olla ongelmallista. Suurena ongelmana voidaan myös nähdä ihmisten asenteet. Mittaamisen tarkoituksena on kuitenkin prosessin parantaminen, ei suinkaan prosessiin osallistuvien henkilöiden työtehon vertailu.

Tärkeä osa ohjelmistotuotantoa on dokumenttien tuottaminen. Dokumenttien kirjoittaminen nähdään usein välttämättömänä pahana ja niiden tuottaminen jää vähälle huomiolle. Projektin koosta ja monimutkaisuudesta riippuu, mitä dokumentteja projektista pitää tehdä. Jokaisesta tuotteesta tulee kuitenkin olla tietty perusdokumentaatio. Dokumentointia voidaan helpottaa käyttämällä dokumentointimalleja ja tarkastusmenettelyä. Ohjelmistotyön tuottavuutta voidaan parantaa muun muassa uudelleenkäyttöä lisäämällä.

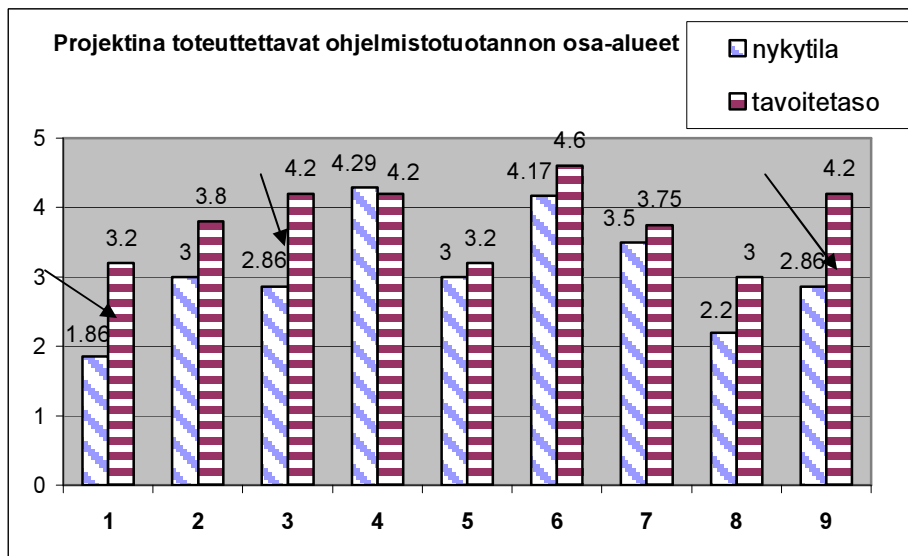
## 3.2 Kyselyn tulokset

Ohjelmistotuotantoon liittyvissä kysymyksissä vastausfrekvenssi vaihteli kahden ja seitsemän välillä. Kaikki vastaajat eivät merkinneet tavoitetasoa. Vastauksista kuusi oli ohjelmistoyrityksistä. Yksi vastaajista ei ilmoittanut yksikkönsä tyyppiä. Seuraavana esitetään kyselyn tulokset osa-alueittain.

Käytetty asteikko oli: 0 = Ei koskaan, 1 = Harvoin, 2 = Joskus, 3 = Usein, 4 = Lähes aina, 5 = Aina. Kuvissa olevat nuolet osoittavat niitä vaihtoehtoja, joissa vastaajien mielestä on eniten kehitettävää eli nyky- ja tavoitetilän välinen ero on huomattava.

### 3.2.1 Projektina toteutettavat ohjelmistotuotannon osa-alueet

Kyselyn mukaan, ohjelmistotuotannon osa-alueista toteutus ja käyttöönotto toteutetaan "lähes aina" projektina. Vastaajat toivovat, että tavoitetilassa myös määrittely, suunnittelu, koulutus ja ylläpito toteutetaan useimmiten projekteina. Eniten kehittämistarpeita on esitutkimuksen, suunnittelun ja ylläpidon toteuttamisessa (ks. kuvan 7 nuolet).

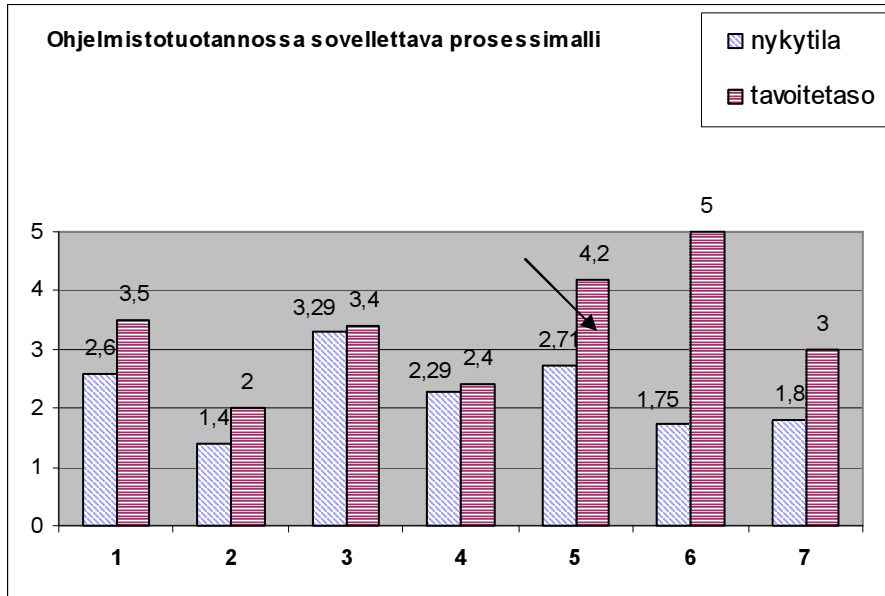


Kuva 7. Projektina toteutettavat ohjelmistotuotannon osa-alueet

- |                |                           |
|----------------|---------------------------|
| 1. Esitutkimus | 6. Käyttöönotto           |
| 2. Määrittely  | 7. Järjestelmän koulutus  |
| 3. Suunnittelu | 8. Toiminnan kehittäminen |
| 4. Toteutus    | 9. Ohjelmiston ylläpito   |
| 5. Testaus     |                           |

### 3.2.2 Ohjelmistotuotannossa sovellettava prosessimalli

Ohjelmistotuotannossa sovellettavista prosessimalleista kyselyn mukaan suosituin on inkrementaalinen malli. Seuraavana on komponenttipohjainen malli. Vastausten perusteella tavoitetilassa käytetyin malli on komponenttipohjainen malli. Kahden vastaajan mielestä RUP:a tulisi käyttää aina (Kuva 8). Kukaan vastaajista ei maininnut, että ohjelmistotuotannossa käytetään jotain muuta mallia.

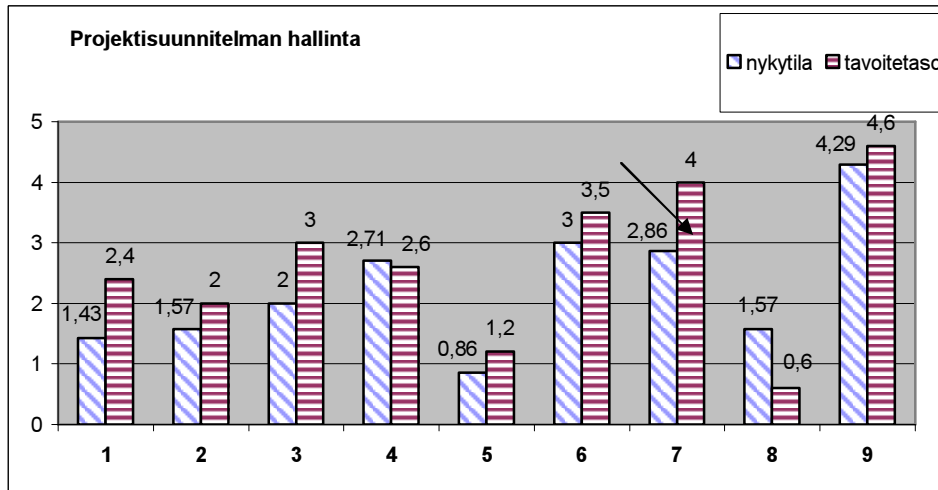


Kuva 8. Ohjelmistotuotannossa sovellettava prosessimalli

1. Vesiputousmalli
2. Spiraalimalli
3. Inkrementaarinen malli
4. Protoilu
5. Komponenttipohjainen malli
6. Rational Unified Process (RUP)
7. V-malli

### 3.2.3 Projektisuunnitelman hallinta

Kuvassa 9 on kuvattu projektisuunnitelman hallinta. Kyselyssä erottuu selvästi, että projektisuunnitelman versiointia pidetään tärkeänä asiana sekä nyt että tavoitetilassa. Tavoitetilassa yleistyy projektin jokaisen vaiheen jälkeen tapahtuva seuraavan vaiheen suunnitelman tarkennus. Projektisuunnitelmaa muutetaan ohjelmistotuotannon eri vaiheissa, harvinaista on, ettei sitä muuteta lainkaan. Projektisuunnitelman muutos on harvoin hallitsematonta ja tavoitteena on, että muutos olisi entistä harvemmin hallitsematonta. Kuutoskohdassa vastaajia pyydettiin tarkentamaan, missä vaiheessa projektisuunnitelmaa muutetaan. Yksi vastaajista mainitsi, että projektisuunnitelmaa muutetaan tarpeen mukaan ja toisen vastaajan mukaan sitä muutetaan kesken toteutuksen.



Kuva 9. Projektisuunnitelman hallinta

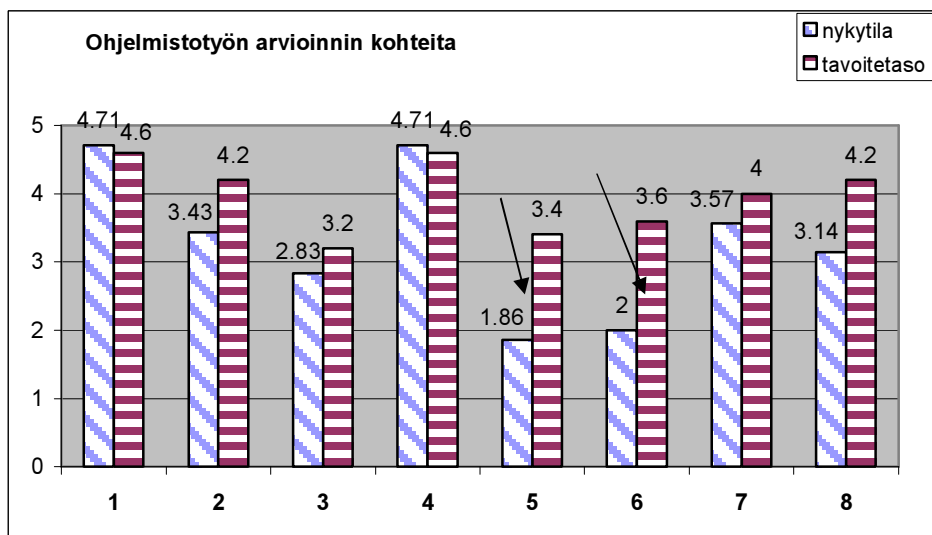
1. Projektisuunnitelmaa muutetaan ennen tarvekartoitusta
2. Projektisuunnitelmaa muutetaan ennen tarjouspyyntöjä
3. Projektisuunnitelmaa muutetaan ennen vaatimusmäärittelyjä
4. Projektisuunnitelmaa muutetaan ennen toteutuksen aloittamista
5. Projektisuunnitelmaa ei muuteta ollenkaan
6. Projektisuunnitelmaa muutetaan muussa vaiheessa, missä?
7. Projektin jokaisen vaiheen jälkeen tarkennetaan seuraavan vaiheen suunnitelma
8. Suunnitelman muutos on hallitsematonta
9. Projektisuunnitelmat versioidaan

### 3.2.4 Dokumentointi

Yleisimmin dokumentaatiossa noudatetaan itse määriteltyä dokumentaation vähimmäistasoa. Dokumenteille tehdään joskus tarkastus. Tavoitteena on lisätä tarkastusten määrää. Dokumentointi voidaan myös hoitaa niin, että jokainen dokumentoi parhaaksi katsomallaan tavalla. Tästä tavasta halutaan kuitenkin luopua. Kyselytutkimusten tulosten mukaan dokumentointia tehdään harvoin standardin mukaan. Kukaan vastaajista ei maininnut, minkä standardin mukaan he dokumentoivat.

### 3.2.5 Ohjelmistotyön arvioinnin kohteet

Kuvassa kymmenen on esitetty eri arviointikohteiden yleisyys. Eniten arvioidaan ohjelmiston kehittämisen vaatimaa työmäärää. Seuraavaksi yleisin arvioinnin kohde on ohjelmistojen laatu. Tavoitetilassa ohjelmiston kehityksen vaatiman työmäärän arviointi säilyttää asemansa. Lähes yhtä yleistä on investointien kannattavuuden ja ohjelmiston käytettävyyden arviointi. Suhteellisesti eniten kasvaa ohjelmistotyön tuottavuuden ja tehokkuuden arviointi (ks. kuvan 10 nuolet).



Kuva 10. Ohjelmistotyön arvioinnin kohteet

1. Työmäärä
2. Investointien kannattavuus
3. Ohjelmiston koko
4. Ohjelmiston vaatima työmäärä
5. Ohjelmistotyön tehokkuus
6. Ohjelmistotyön tuottavuus
7. Ohjelmistojen laatu
8. Ohjelmiston käytettävyys

### 3.2.6 Projektin tai ohjelmistotyön arviointimenetelmät

Projektia tai ohjelmistotyötä voidaan arvioida eri menetelmiä käyttämällä. Kyselyn tulosten perusteella yleisimmin projektia tai ohjelmistotyötä arvioidaan palavereissa ja projektiryhmässä. Lähes yhtä käytetty oli arviointi ennalta määrätyissä tarkistuspisteissä. Mittareiden käyttö projektin tai ohjelmistotyön arvioinnissa ei ole kovinkaan yleistä. Tavoitetilassa mittareiden käyttö kuitenkin lisääntyy "harvoin"-tasolta "joskus"-tasolle.

### 3.2.7 Mittareiden käyttö ohjelmistotuotannossa

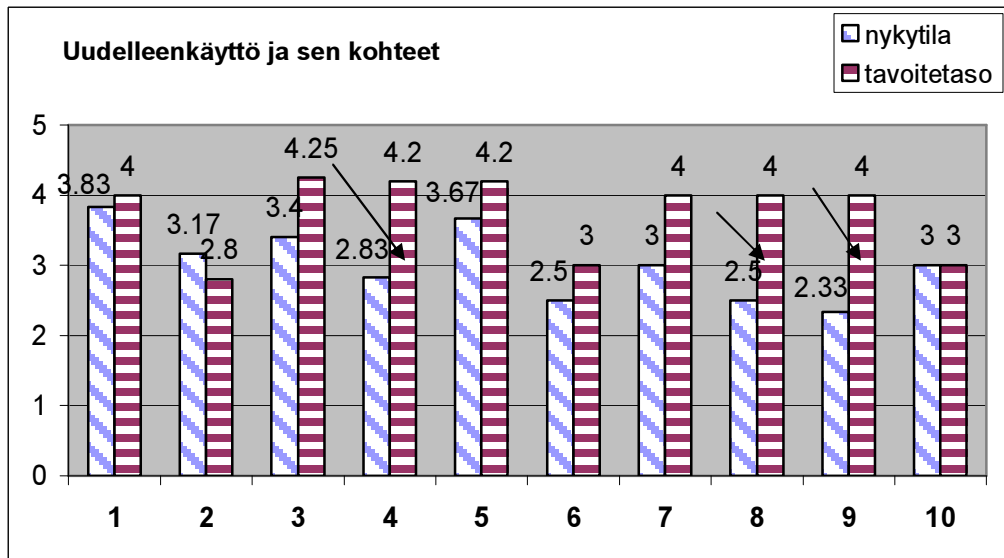
Kyselyn perusteella mittareiden käyttö ei ole kovin yleistä. Mittareiden eri käyttökohteet olivat samat kuin arvioinnin kohteet (ks. kappale 3.2.5). Eniten mittareita käytetään ohjelmiston kehittämisen vaatiman työmäärän seurantaan. Tavoitetilassa molempien mitattavien kohteiden tasot pysyvät jotta-kuinkin samoina. Suhteellisesti tasoaan kasvattaa eniten mittareiden käyttö investointien kannattavuuden seurantaan. Mittareiden käyttö ohjelmiston koon mittaamiseen ei ole yleistä eikä näytä paljontaan yleistyvän. Muuten mittareiden käyttö tulee yleistymään vähintään "usein"-tasolle.

### 3.2.8 Versionhallinta

Avoimella kysymyksellä kartoitettiin versiohallinnan toteutustapoja. Kyselyyn vastanneista viisi vastasi kysymykseen. Vastaaajista kolme kertoi, että ohjelmien versiohallinta hoidetaan heillä versiohallintatyökalun avulla. Yksi vastasi versiohallinnan hoidettavan PVCS:n avulla ja yksi vastasi ohjelmakoodi CVS:llä ja komponenttikokoonpanot manuaalisesti.

### 3.2.9 Uudelleenkäyttö ja sen kohteet

Kuvassa 11 on esitetty kohteet, joita uudelleenkäytetään ohjelmistotuotannossa. Tällä hetkellä eniten uudelleenkäytetään kokemusta. Seuraavina ovat oman organisaation komponenttien uudelleenkäyttö ja yksikössä tehtyjen aliohjelmien uudelleenkäyttö. Molempien edellä mainittujen kohteiden uudelleenkäyttö on yleistä myös tavoitetilassa. Aliohjelmakirjastojen käyttö on nykyistä yleisempää. Suhteellisesti eniten suosiotaan kasvattaa testitapausten (ks. Jäntti, 2003) ja suunnittelumallien (ks. Immonen, 2002) uudelleenkäyttö. Kohdassa 10 yksi vastaajista mainitsi ohjelmistotyön vaihetuotteena syntyvät toimitusdokumentit.



Kuva 11. Uudelleenkäyttö ja sen kohteet

- |   |  |
|---|--|
| 1. Henkilöt pysyvät saman aihepiirin projekteissa: kokemuksen uudelleenkäyttö | 6. Muualta hankitut komponentit                  |
| 2. Copy paste   | 7. Dokumenttipohjat                              |
| 3. Yksikössä tehdyt aliohjelmat   | 8. Suunnittelumallit                             |
| 4. Aliohjelmakirjastojen käyttö   | 9. Testitapaukset                                |
| 5. Oman organisaation komponentit   | 10. Muut ohjelmistotyössä syntyvät vaihetuotteet |

### 3.2.10 Standardien käyttö

Viiden vastaajan mukaan heidän yksiköissään sovelletaan ohjelmistotuotannon laatustandardeja. Käytössä olevat standardit ovat: ISO 9001, Pro 2, RUP ja oma laatustandardi, joka perustuu ISO 9001:een. Kaksi vastaajaa ilmoitti, että laatustandardeja ei sovelleta. Kahden vastaajan mukaan heidän yksiköissään sovelletaan terveydenhuollon standardeja. Käytössä olevat standardit ovat: Juhta ja yleiset käytössä olevat. Yksi vastaajista ilmoittaa, että standardeja ei käytetä. Neljä vastaajaa ei vastannut kysymykseen.

### 3.2.11 Ohjelmistotuotannon kehittämishankkeiden lähtökohdat

Vastaajille oli annettu viisi valmista vaihtoehtoa sekä mahdollisuus kirjoittaa oma vaihtoehtonsa. Kyselyn mukaan ohjelmistotuotannon kehittämisen lähtökohtana on lähes aina yksikön toimintastrategia. Tärkeänä nähdään myös tietotekniikkastrategia. Myös tavoitetilassa nämä pysyvät tärkeimpinä lähtökohtina. Tällä hetkellä vähiten käytetyin lähtökohta on esitutkimus. Yhden vastaajan mielestä kehittämishankkeiden pitää aina pohjautua asiakkaan tarpeisiin.

### 3.2.12 Projektin työntekijöiden osallistuminen projektin eri vaiheisiin

Kysymyslomakkeella valmiina olleet projektin työntekijät olivat: päätöksentekijä, pääkäyttäjä, käyttäjä, systeemin suunnittelija, ohjelmoija, ylläpitäjä ja konsultti. Vastaajien itse lisäämiä olivat: tuotepäällikkö, testaaaja ja projektipäällikkö. Sulkuihin on merkitty lomakkeiden määrä, jossa vaihtoehto oli valittu.

Vaatusmäärittelyyn osallistuvat:

- päätöksentekijä (3)
- pääkäyttäjä (4)
- käyttäjä (3)
- systeemin suunnittelija (3)
- ohjelmoija (1)
- tuotepäällikkö (1)

Määrittelyyn (analyysiin) osallistuvat:

- päätöksentekijä (2)
- pääkäyttäjä (4)
- käyttäjä (1)
- systeemin suunnittelija (4)
- tuotepäällikkö (1)

Toteutukseen osallistuvat:

- pääkäyttäjä (1)
- systeemin suunnittelija (3)
- ohjelmoija (6)
- ylläpitäjä (1)

Sopimusneuvotteluihin osallistuvat:

- päätöksentekijä (3)
- pääkäyttäjä (2)
- ylläpitäjä (1)
- konsultti (1)

Suunnitteluun osallistuvat:

- pääkäyttäjä (1)
- systeemin suunnittelija (5)
- ohjelmoija (5)
- ylläpitäjä (1)

Moduulien testaukseen osallistuvat:

- pääkäyttäjä (2)
- käyttäjä (2)
- systeemin suunnittelija (3)
- ohjelmoija (6)
- ylläpitäjä (1)

Moduulien integrointiin osallistuvat:

- pääkäyttäjä (1)
- systeemin suunnittelija (4)
- ohjelmoija (6)
- ylläpitäjä (1)

Järjestelmätestaukseen osallistuvat:

- pääkäyttäjä (5)
- käyttäjä (3)
- systeemin suunnittelija (4)
- ohjelmoija (3)
- testaaja (1)

Hyväksymistestaukseen osallistuvat:

- päätöksentekijä (2)
- pääkäyttäjä (5)
- käyttäjä (3)
- systeemin suunnittelija (2)
- ohjelmoija (1)
- testaaja (1)
- projektipäällikkö (1)

Uusien ominaisuuksien toteutukseen osallistuvat:

- käyttäjä (1)
- systeemin suunnittelija (1)
- ohjelmoija (4)
- ylläpitäjä (3)

Virheiden korjaukseen osallistuvat:

- käyttäjä (1)
- systeemin suunnittelija (1)
- ohjelmoija (3)
- ylläpitäjä (3)

Integrointitestaukseen osallistuvat:

- pääkäyttäjä (3)
- käyttäjä (2)
- systeemin suunnittelija (4)
- ohjelmoija (6)
- ylläpitäjä (1)
- testaaja (1)

Käyttöönottoon osallistuvat:

- päätöksentekijä (2)
- pääkäyttäjä (4)
- käyttäjä (4)
- systeemin suunnittelija (2)
- ohjelmoija (1)
- ylläpitäjä (1)
- konsultti (1)

Ylläpitoon osallistuvat:

- käyttäjä (1)
- ohjelmoija (1)
- ylläpitäjä (2)

Muutosten toteutukseen osallistuvat:

- käyttäjä (1)
- systeemin suunnittelija (1)
- ohjelmoija (4)
- ylläpitäjä (3)

Regressiotestaukseen osallistuvat:

- pääkäyttäjä (1)
- käyttäjä (1)
- systeemin suunnittelija (1)
- ohjelmoija (2)
- ylläpitäjä (2)
- projektipäällikkö (1)

### 3.3 Pohdintaa

Kyselyn perusteella on selvästi nähtävissä, että ohjelmistotuotannon eri osa-alueet toteutetaan ja tullaan vastedes yhä enenevässä määrin toteuttamaan projekteina. Ohjelmiston kehittäminen voidaan useimmiten jakaa peräkkäisiin tai rinnakkaisiin osaprojekteihin, joten projektityöskentely on luonteva menetelmä.

Projektisuunnitelma on tärkeä projektin ohjauksen ja seurannan väline. Kyselyn mukaan projektisuunnitelma laaditaan lähes aina ja suuntaus on sen laatimiseen aina. Projektisuunnitelmaan tehdään usein muutoksia, mutta muutokset ovat yleensä hallittuja. Projektisuunnitelmat myös versioidaan lähes aina.

Projektin edistymistä seurataan kyselyn mukaan lähes aina. Projektin edistymistä arvioidaan useimmiten projektiryhmässä ennalta määrättyissä tarkistuspisteissä. Mittareiden käyttö projektin seurannassa on harvinaista. Niiden käytön toivotaan hieman lisääntyvän. Vastausten perusteella mittareiden käyttö ohjelmistotuotannossa on kaiken kaikkiaan aika harvinaista. Mittareita käytetään lähes aina ohjelmiston kehittämisen vaatiman työmäärän seurantaan. Muuten mittareiden käyttö jää "harvoin"-tasolle. Tavoitetilassa mittareita toivotaan kuitenkin käytettävän usein.

Uudelleenkäyttö on tällä hetkellä melko yleistä, mutta myös tällä osa-alueella tavoitellaan parempaa tasoa. Kaiken kaikkiaan on nähtävissä, että yritykset pyrkivät kehittämään toimintaansa, vaikka monessa suhteessa jo nyt ollaan hyvällä tasolla.

Ohjelmistotuotannon suurimpina ongelmina yksi vastaajista näki aikataulujen ennustettavuuden ja muutosvaatimusten hallinnan projektin aikana. Ohjelmistotuotannolla on lyhyt historia verrattuna muuhun teolliseen tuotantoon, joten osittain syy ohjelmistotuotannon ongelmiin löytyy alan kehittymättömyydestä. Ohjelmistotuotteet ovat usein laajoja ja monimutkaisia, tästä syystä ohjelmistojen toteuttamistyöhön käytettävää aikaa on vaikea arvioida. Ohjelmistotuotannon ongelmia ratkomaan on kehitetty muun muassa uusia työkaluja ja uusia työmenetelmiä.

Ohjelmistojen toteuttaminen ei ole rutiininomaista. Ohjelmistotuotannon tekniikan kehittymisen lähtökohtana ovat käytännön ongelmat, joihin tulisi löytää ratkaisu. Yksi vastaajista kuvasi ohjelmistotuotantoa haastavaksi ja sitä se kyllä on.

# 4 TIETOJÄRJESTELMÄN MALLINNUS

## 4.1 Yleistä tietojärjestelmän mallinnuksesta

Tietojärjestelmän toteuttamisen aikana rakennettavasta järjestelmästä laaditaan useita erilaisia kuvauksia, toisin sanoen järjestelmää mallinnetaan eri tavoin. Mallinnuksen tarkoituksena on laatia ohjeita ja piirustuksia tuotteen rakentamista varten. Ohjelmistoa ei voida tehdä, kuten ei esimerkiksi taaloakaan, ilman selvää suunnitelmaa ja piirustuksia. Mallinnusta tehdään ohjelmistotuotannon eri vaiheissa siten, että aluksi tehdään karkean tason kuvauksia, joista edetään kohti yksityiskohtaisempia kuvauksia. Mallinnuksen tarkoituksena on myös välittää tietoa niiden ihmisten välillä, jotka osallistuvat järjestelmän kehittämiseen.

Mallien avulla järjestelmää voidaan esitellä erilaisista näkökulmista. *Ulkoisen näkökulman* avulla mallinnetaan järjestelmän asiayhteyttä tai ympäristöä. Näin saadaan tietoa sekä järjestelmän toiminnasta että teknisestä ympäristöstä, jossa järjestelmää tullaan käyttämään. Samalla saadaan selville muun muassa muut järjestelmät, jotka ovat vuorovaikutuksessa kehitettävän järjestelmän kanssa. *Käyttäytymisnäkökulma* auttaa mallintamaan järjestelmän käyttäytymistä eli järjestelmän avulla suoritettavia toimintoja. *Rakenteellinen näkökulma* auttaa järjestelmän arkkitehtuurin tai järjestelmän tuottaman tiedon rakenteen mallintamisessa. (Ks. Kruchten, 1995)

Mallinnusta helpottamaan on kehitetty useita erilaisia kuvaustekniikoita, esimerkiksi luokkakaa- viot, prosessikaaviot ja tilakaaviot. Kuvaustekniikoita voidaan soveltaa menetelmien avulla. Menetelmiä ovat esimerkiksi seinätaulumenetelmä, Case-välineet ja UML. UML (= Unified Modeling Language) on mallinnuskieli, joka on tarkoitettu ohjelmiston mallintamiseen, määrittelyyn, rakentamiseen ja dokumentointiin.

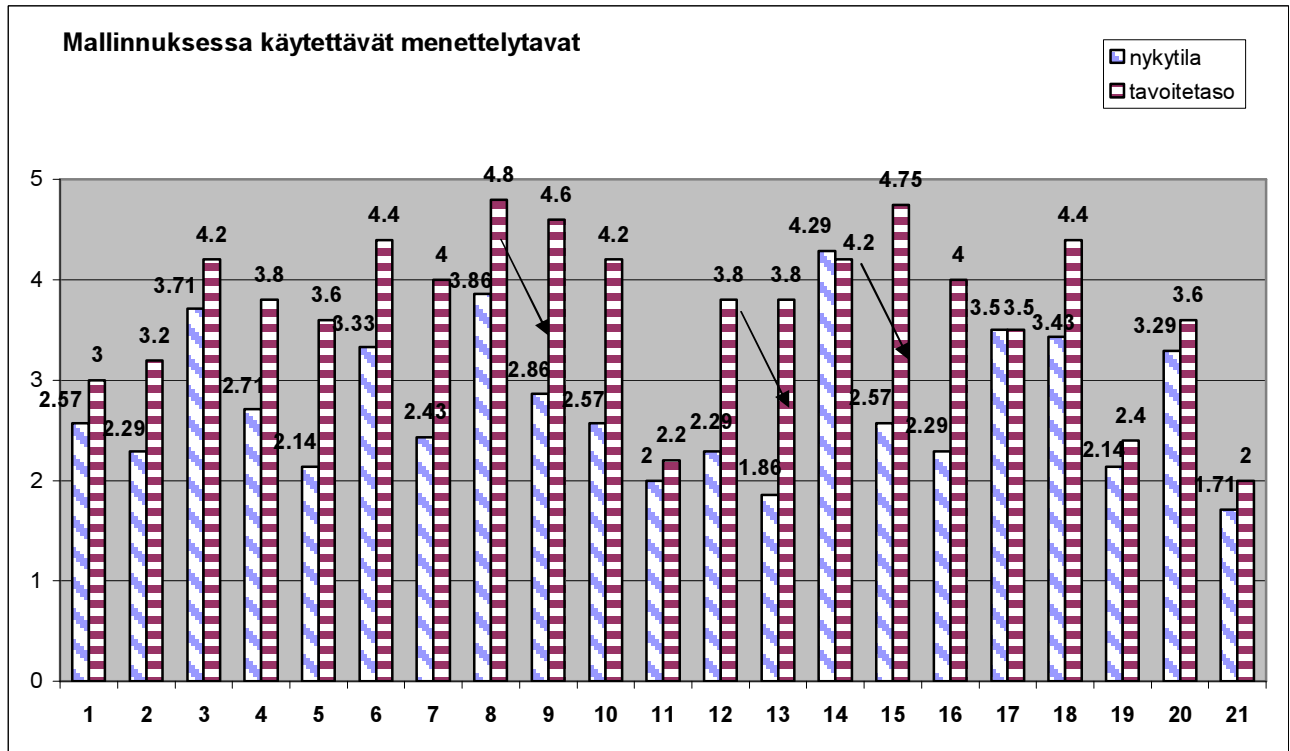
## 4.2 Kyselyn tulokset

Mallinnukseen liittyvissä kysymyksissä vastausfrekvenssi vaihteli kahden ja seitsemän välillä. Kaikki vastaajat eivät merkinneet tavoitetasoa. Vastauksista neljä oli ohjelmistoyrityksistä. Yksi vastaajista oli sairaalan ATK-osastolta. Kaksi vastaajaa ei ilmoittanut yksikkönsä tyyppiä. Seuraavana esitetään kyselyn tulokset osa-alueittain.

Käytetty asteikko oli: 0 = Ei koskaan, 1 = Harvoin, 2 = Joskus, 3 = Usein, 4 = Lähes aina, 5 = Aina. Kuvissa olevat nuolet osoittavat niitä vaihtoehtoja, joissa vastaajien mielestä on eniten kehitettävää eli nyky- ja tavoitetilan välinen ero on huomattava.

### 4.2.1 Mallinnuksessa käytössä olevat menettelytavat

Tällä hetkellä mallinnuksessa eniten käytössä olevat menettelytavat ovat: näyttöjen suunnittelu, toiminnallisten vaatimusten määrittely ja nykyjärjestelmän tutkiminen. Tavoitetilassa käytetyimpiä ovat: toiminnallisten vaatimusten määrittely, komponenttikeskeinen suunnittelu ja laadullisten vaatimusten määrittely. Suhteellisesti eniten kasvaa komponenttikeskeinen suunnittelu, oliokeskeinen analyysi ja laadullisten vaatimusten määrittely (ks. kuvan 12 nuolet). Kyselyn perusteella vähiten käytettyjä ovat ja tulevat olemaan pika-ohjelmointi ja seinätaulutekniikka.

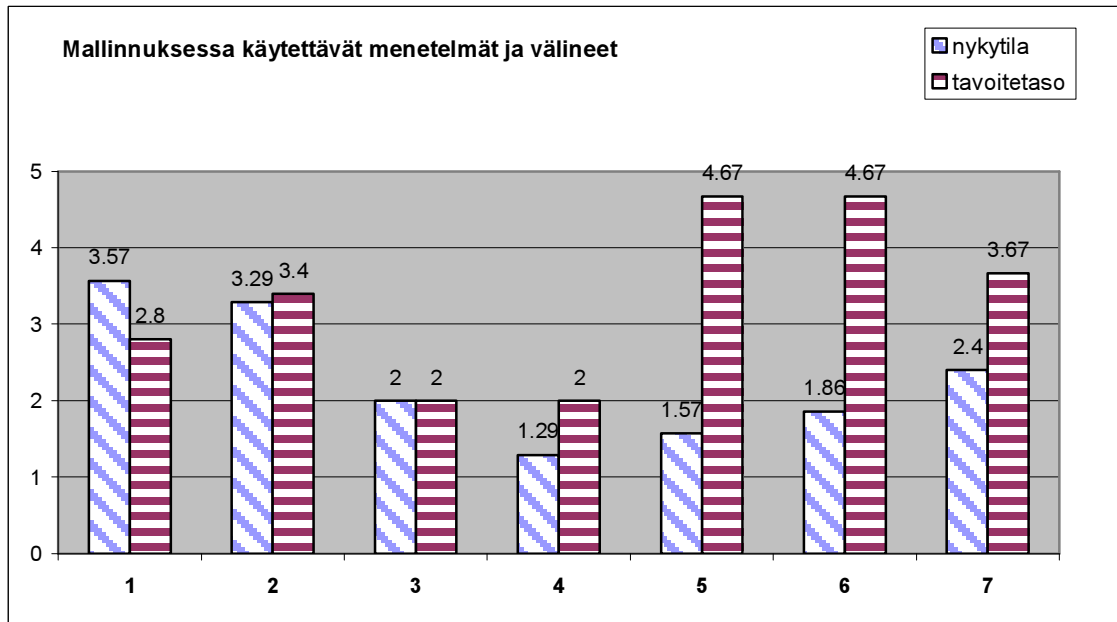


Kuva 12. Mallinnuksessa käytettävät menettelytavat

- |  |                                       |
|--|---------------------------------------|
| 1. Valmisohjelmistoihin tutustuminen                                     | 11. Seinätaulutekniikka               |
| 2. Samanlaista toimintaa harjoittavien yritysten toimintaan tutustuminen | 12. Käsitemanalyysi                   |
| 3. Nykyjärjestelmän tutkiminen   | 13. Oliokeskeinen analyysi            |
| 4. Toimintojen mallintaminen   | 14. Näyttöjen suunnittelu             |
| 5. Prosessien mallintaminen  | 15. Komponenttikeskeinen suunnittelu  |
| 6. Toimintojen kehittäminen  | 16. Toimialakomponenttien suunnittelu |
| 7. Prosessien kehittäminen   | 17. Tietopohjainen suunnittelu        |
| 8. Toiminnallisten vaatimusten määrittely                                | 18. Toimintopohjainen suunnittelu     |
| 9. Laadullisten vaatimusten määrittely                                   | 19. Prototyypit                       |
| 10. Rakenteinen analyysi ja suunnittelu                                  | 20. Ideointi                          |
|  | 21. Pika-ohjelmointi                  |

#### 4.2.2 Mallinnuksessa käytettävät menetelmät ja välineet

Tällä hetkellä käytetyimmät välineet ovat paperi ja kynä sekä piirtoväline. Tavoitetilassa suosiotaan lisäävät eniten UML ja suunnittelumallit, jotka myös ovat käytetyimmät välineet. Tällä hetkellä molempia käytetään joskus. Myös sovelluskehukset koettiin kiinnostaviksi. Case-välineiden käyttö pysyy vähäisenä (Kuva 13). Yksi vastaajista ilmoittaa, että käytössä on lähes aina yrityksen itse kehittämä menetelmä. Tavoitteena on käyttää tätä menetelmää aina. Yksi vastaajista ilmoittaa, että mallinnukseen käytetään lähes aina omaa ohjelmistoa. Tavoitteena on käyttää tätä ohjelmaa aina.

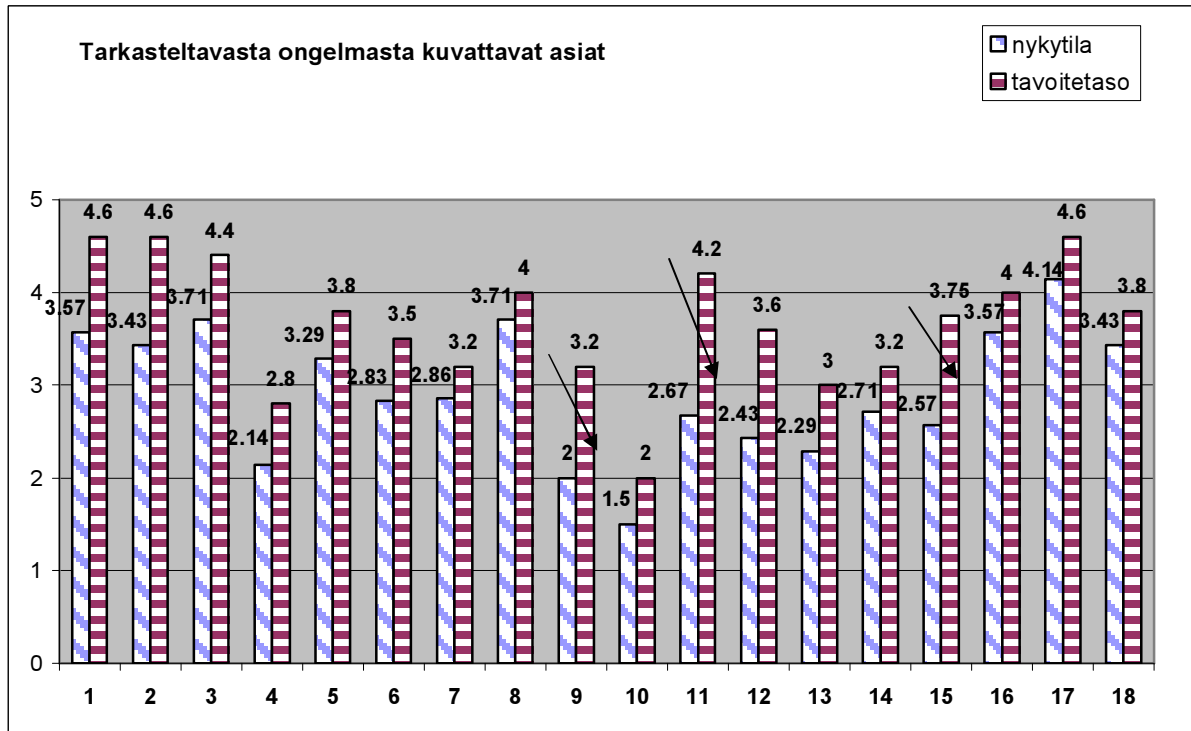


Kuva 13. Mallinnuksessa käytettävät menetelmät ja välineet

1. Paperi ja kynä
2. Piirtoväline
3. Seinätaulutekniikka
4. Case-väline
5. UML
6. Suunnittelumallit
7. Sovelluskehukset

#### 4.2.3 Tarkasteltavasta ongelmasta kuvattavat asiat

Yleisimmin tarkasteltavasta ongelmasta kuvataan: tietokannat, mahdollinen ratkaisu ja yhteydet naapurisysteemeihin. Tavoitetilassa useimmiten kuvattavat asiat ovat: järjestelmän tarkoitus, ratkaisuidea ja tietokannat. Suhteellisesti eniten tasoaan kasvattavat avaintoimintojen kuvaus, ihmisten välisen yhteistyön kuvaus ja olemassa olevien komponenttien kuvaus (ks. kuvan 14 nuolet). Vähiten tärkeänä asiana pidettiin organisaatiohierarkian kuvaamista. Kukaan vastaajista ei maininnut muuta ongelman mallintamiseen käytettävää kuvausta.



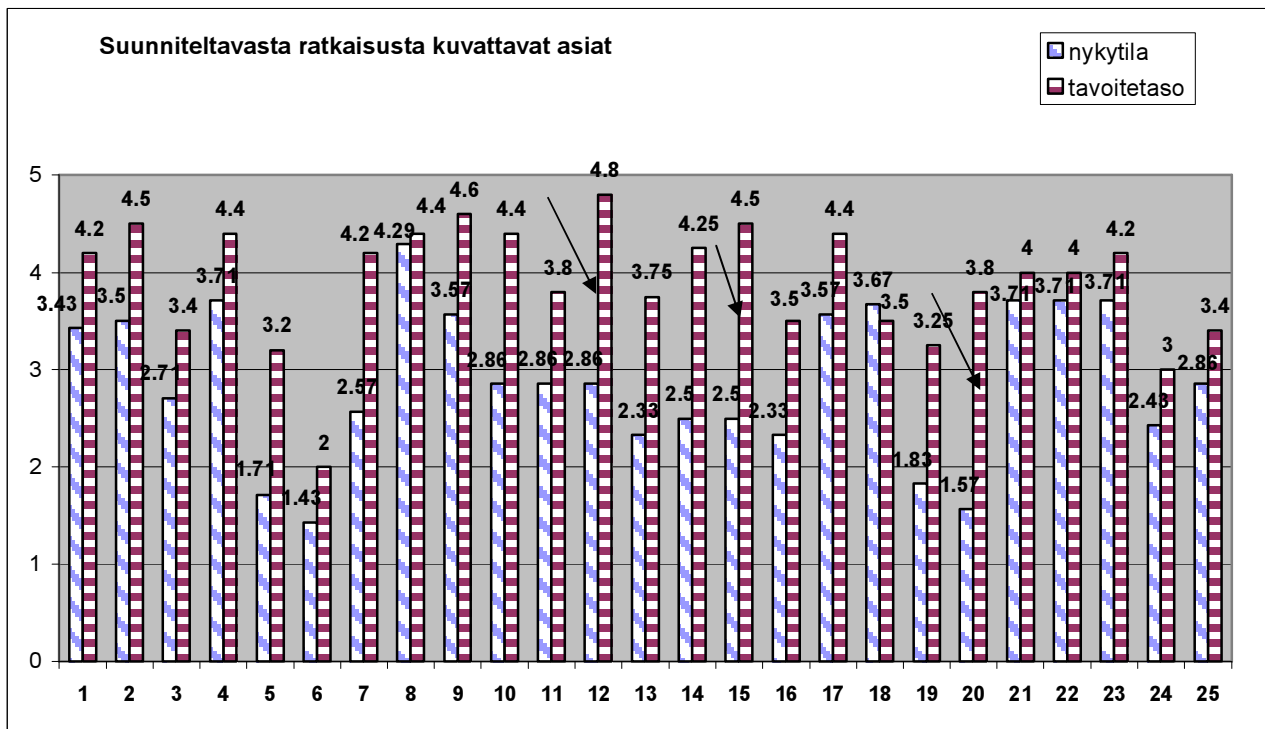
Kuva 14. Tarkasteltavasta ongelmasta kuvattavat asiat

1. Järjestelmän tarkoitus
2. Ratkaisuidea
3. Mahdollinen ratkaisu
4. Hyväksymisen mittarit
5. Kohdealueen rajaus
6. Jako alisysteemeihin
7. Naapurisysteemit
8. Yhteydet naapurisysteemeihin
9. Ihmisten välinen yhteistyö
10. Organisaatiohierarkia
11. Avaintoiminnot, toimintoketjut
12. Käsitteet ja niiden väliset suhteet
13. Olemassa olevat ohjelmistot
14. Käytössä olevat ohjelmistot
15. Olemassa olevat komponentit
16. Ohjelmien väliset kutsusuhteet
17. Tietokannat
18. Järjestelmät, laitteet, tietoliikenne

#### 4.2.4 Suunniteltavasta ratkaisusta kuvattavat asiat

Suunniteltavasta ratkaisusta kuvataan yleisimmin näytöt. Seuraavaksi eniten kuvattavia asioita ovat: yhteydet naapurisysteemeihin, tietokantaratkaisut, tietokannan tietosisältö ja infrastruktuuri. Yleisimmin kuvattavana asiana tavoitetilassa on toimintoketjujen (avaintoimintojen kuvaaminen). Seuraavina tulevat arkkitehtuurin kuvaus, jako alisysteemeihin ja komponenttien rajapinnat.

Suhteellisesti eniten tasoaan kasvattavat tilakaaviot. Myös komponenttien rajapintojen kuvaus ja toimintoketjujen kuvaus kasvattavat suosiotaan merkittävästi (ks. kuvan 15 nuolet). Organisaatiohierarkian kuvaus ei ole kovin yleistä. Yhden vastaajan mukaan mallinnuksessa käytettävät valmiit sisällölliset standardit ovat: omat templatet ja UML. Molemmissa nykytilana "lähes aina", tavoitetiloja ei vastaaja määritellyt.



Kuva 15. Suunniteltavasta ratkaisusta kuvattavat asiat

- |   |   |
|---|---|
| 1. Kohdealueen rajaus                   | 14. Komponenttien väliset suhteet             |
| 2. Jako alisysteemeihin                 | 15. Komponenttien rajapinnat                  |
| 3. Naapurisysteemit                     | 16. Käytettävät valmiskomponentit             |
| 4. Yhteydet naapurisysteemeihin         | 17. Ohjelmistot                               |
| 5. Ihmisten välinen yhteistyö           | 18. Ohjelmien väliset kutsusuhteet            |
| 6. Organisaatiohierarkia                | 19. Yhteistyökaaviot (esimerkiksi skenaariot) |
| 7. Ihminen – tietokone vuorovaikutus    | 20. Tilakaaviot                               |
| 8. Näytöt                               | 21. Tietokantaratkaisut                       |
| 9. Arkkitehtuuri                        | 22. Tietokannan tietosisältö                  |
| 10. Käsitteet ja niiden väliset suhteet | 23. Infrastruktuuri                           |
| 11. Toimintojen osiin jako              | 24. Middleware                                |
| 12. Toimintoketjut (avaintoiminnot)     | 25. Tietoliikenne                             |
| 13. Toimialakomponentit                 |   |

#### 4.2.5 Mallinnuksessa tuotettavat kaaviot ja kuvaukset

Mallinnuksessa tuotettavista kuvauksista yleisin on arkkitehtuurikuvaus, joka tehdään lähes aina. Tavoitetilassa se tulisi vastaajien mielestä tehdä aina. Usein tehtäviä kaavioita ovat: komponenttikaavio, luokkakaavio, käyttötapaus skenaariot ja käyttötapauskaavio. Tavoitetilassa näiden merkitys vielä korostuu. Suhteellisesti eniten tasoaan kasvattavat aktiviteettikaavio ja pakettikaavio; joskus-tasolta lähes aina -tasolle asti. Vähiten käytetty kaavio on päätöstaulu. Tälläkin osa-alueella oli kaikkien vaihtoehtojen kohdalla havaittavissa selvää tasonnousua.

### 4.3 Pohdintaa

Kyselyn perusteella on nähtävissä, että toiminnallisten vaatimusten määrittely pysyy tärkeimpänä mallinnuksessa käytettynä menettelytapana. Sen sijaan laadullisten vaatimusten määrittely, komponenttikeskeinen suunnittelu ja oliokeskeinen analyysi, yleistyvät huomattavasti nykytasoonsa nähden. Kaiken kaikkiaan, lähes kaikki menettelytavat ovat tavoitetilassa tämänhetkistä käytetympiä. Ainoastaan seinätaulutekniikka, prototyypit ja pika-ohjelmointi jäävät myös tavoitetilassa harvointasolle. Mallinnuksessa käytetyissä menetelmissä ja välineissä on havaittavissa selvä suuntaus pois paperista ja kynästä kohti UML:ää ja suunnittelumalleja.

Suunniteltavasta ratkaisusta tärkeimpinä pidetyt kuvattavat asiat ovat tavoitetilassa eri asioita kuin tällä hetkellä tärkeimpinä pidetyt asiat. Tällä hetkellä tärkeinä asioina pidetään tietokantoja, infrastruktuuria ja näyttöjä. Tavoitetilassa tärkeimmiksi asioiksi nousevat alisysteemeihin jako, arkkitehtuuri, toimintoketjut ja komponentit. Huomattava on kuitenkin se seikka, että organisaatiohierarkiaa lukuun ottamatta, kaikki vaihtoehtoina annetut asiat ovat tavoitetilassa vähintään usein kuvattuja.

Mallinnuksen ongelmia vastaajien mielestä ovat mallinnuksen käytäntöön peilaamisen vaikeus, UML:n sisältämien mallien puutteellisuus kunnan mallin tekemiseen (esimerkiksi käyttöliittymä) ja mallinnustyökalujen riittämätön reaali maailman mallinnuksen tuki.

Mallinnuksen kehittämiskohteena voisi olla UML:n, suunnittelumallien ja sovelluskehysten käytön lisääminen. Mallinnus pitää selvästi nähdä suunnittelun apukeinona: mitä tulee mallintaa ja miten.

# 5 TEKNIikka JA TOTEUTUS

## 5.1 Yleistä tekniikasta ja toteutuksesta

Ohjelmistojen toteutus on monimutkaista. Viime vuosikymmeninä ohjelmistotuotantoon on kehitetty lukuisia uusia tekniikoita. Tekniikka tuntuu kehittyvän niin nopeasti, ettei kehityksessä pysytäkään mukana. Ennen kuin olet oppinut käyttämään uutta tekniikkaa, tulee markkinoille jo uusi tekniikka. Komponenttipohjainen tekniikka on lupaava tapa, jonka avulla voidaan hallita tietojärjestelmien monimutkaisuutta ja kustannuksia. Sen uskotaan kehittyvän vielä seuraavat kymmenen vuotta.

Terveydenhuoltoalalla järjestelmäympäristöt ovat usein erittäin heterogeenisiä. Samoissa yksiköissä voi olla järjestelmiä, joista kukin vaatii erilaisen infrastruktuurin. Näiden järjestelmien yhteistoiminnan parantaminen on kallista ja vaikeaa. Komponenttipohjaiset järjestelmät mahdollistavat olemassa olevien järjestelmien hyödyntämisen. Vanhoja järjestelmiä voidaan käyttää uuden järjestelmän pohjana. Näin käytössä oleva järjestelmä voidaan uusiksi vähitellen, osa kerrallaan.

Jotta terveydenhuollon tietojärjestelmiä voidaan kehittää, pitää sovellusten olla yhteentoimivia sekä keskenään että ulkoisten yhteistyökumppanien järjestelmien kanssa. Sovellusten yhteistoiminnallisuus voidaan toteuttaa eri tasoilla ja tavoilla. Järjestelmien, joilla on eri toimittajat, yhteentoimivuuden perustana ovat standardit ja yhteiset sopimukset. Terveydenhuollon sovelluksissa käytettyjä standardeja ovat muun muassa HL7, HISA, CORBAmed, UML, XML ja HTML.

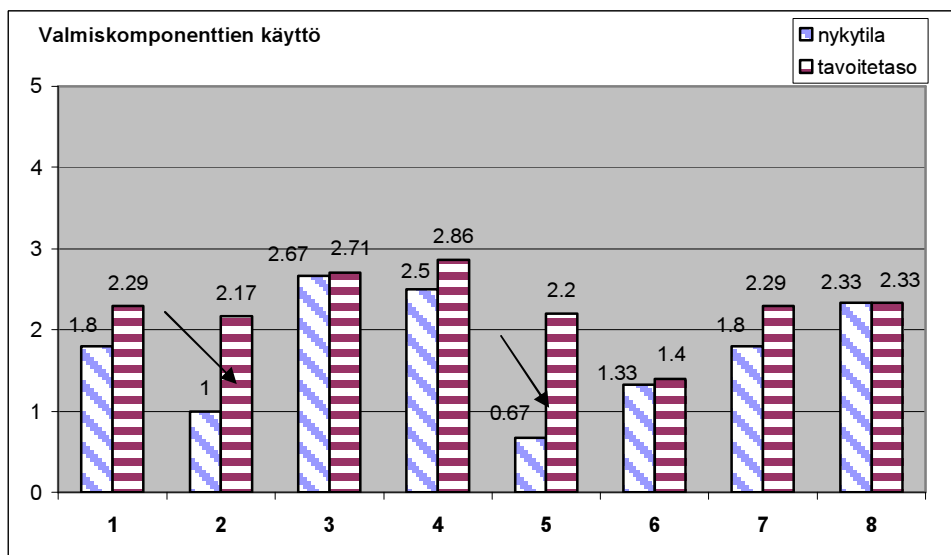
## 5.2 Kyselyn tulokset

Tekniikkaan ja toteutukseen liittyvissä kysymyksissä vastausfrekvenssi vaihteli viiden ja kymmenen välillä. Kaikki vastaajat eivät merkinneet tavoitetasoa. Vastauksista kuusi oli ohjelmistoyrityksistä ja kaksi sairaalan ATK-osastolta. Kaksi vastaajaa ei ilmoittanut yksikkönsä tyyppiä. Seuraavana esitetään kyselyn tulokset osa-alueittain.

Käytetty asteikko oli: 0 = Ei koskaan, 1 = Harvoin, 2 = Joskus, 3 = Usein, 4 = Lähes aina, 5 = Aina. Kuvissa olevat nuolet osoittavat niitä vaihtoehtoja, joissa vastaajien mielestä on eniten kehitettävää eli nyky- ja tavoitetilan välinen ero on huomattava.

### 5.2.1 Valmiskomponenttien käyttö

Kyselyn perusteella käytetyimpiä komponentteja tällä hetkellä ovat itse tehdyt luokkakirjastot. Niiden jälkeen tulevat itse tehdyt komponentit ja koodin uudelleenkäyttö kopioimalla. Nämä kolme ovat myös tavoitetilassa käytetyimmät valmiskomponentit. Selvästi eniten suosiotaan lisää ulkoa käytetyt sovelluspalvelut. Myös ostettujen luokkakirjastojen käyttö lisääntyy tavoitetilassa (Kuva 16). Muina komponentteina yksi käyttäjä mainitsi Open Source -komponentit.



Kuva 16. Valmiskomponenttien käyttö

- |                                |                                       |
|--------------------------------|---------------------------------------|
| 1. Ostetut komponentit         | 5. Ulkoa käytetyt sovelluspalvelut    |
| 2. Ostetut luokkakirjastot     | 6. Ostetut sovelluskehyykset          |
| 3. Itse tehdyt luokkakirjastot | 7. Omat sovelluskehyykset             |
| 4. Itse tehdyt komponentit     | 8. Koodin uudelleenkäyttö kopioimalla |

Valmiskomponenttien suurimmat puutteet ja hyödyntämisen esteet vastaajien mielestä ovat:

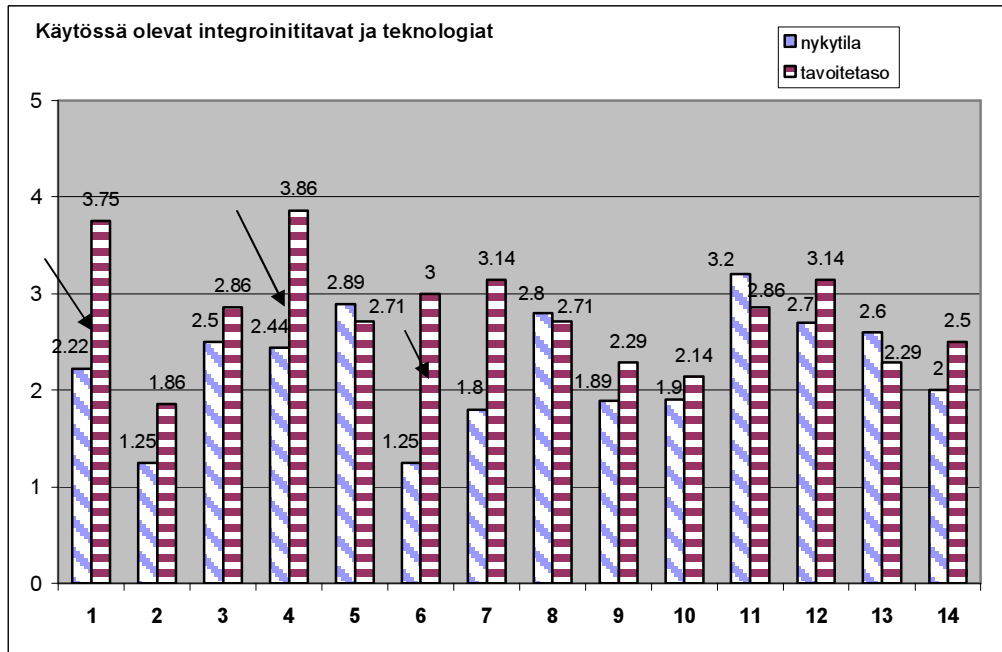
- komponenttien huono laatu
- huono tuki- ja päivityspalvelu
- sopivien komponenttien löytäminen
- standardien puute
- valmiskomponentteja ei ole
- oma yritys tuottaa, ei tarvitse ostaa, apu lähellä ongelmatilanteissa
- ulkomaiset toimittajat
- tietoturva

Valmiskomponenttien käyttöä voitaisiin vastaajien mielestä lisätä, jos

- komponenttien valmistus ja myynti olisi vakiintunutta liiketoimintaa
- terveydenhuollon ohjelmistot ja tiedot olisivat standardoituja

### 5.2.2 Käytössä olevat integrointitavat ja -teknologiat

Useimmiten käytössä oleva integrointitapa on määräajoin eräsiirtona tapahtuvat tiedonsiirrot. Usein käytettyjä tapoja ovat myös tiedon välitys tietalueen tai tietokannan kautta ja samaan koneeseen asennettujen sovellusten välinen integrointi. Tavoitetilassa teknologioista käytetyin on XML-pohjainen integrointi. Myös komponenttipohjainen integrointi on vastausten perusteella suosittua. Näiden kahden taso nousee merkittävästi tavoitetilassa. Suhteellisesti eniten tasoaan kasvattaa web service -tekniikka (Kuva 17). Muista integrointitavoista ja -teknologioista yksi vastaajista mainitsee J2EE integrointialustan.

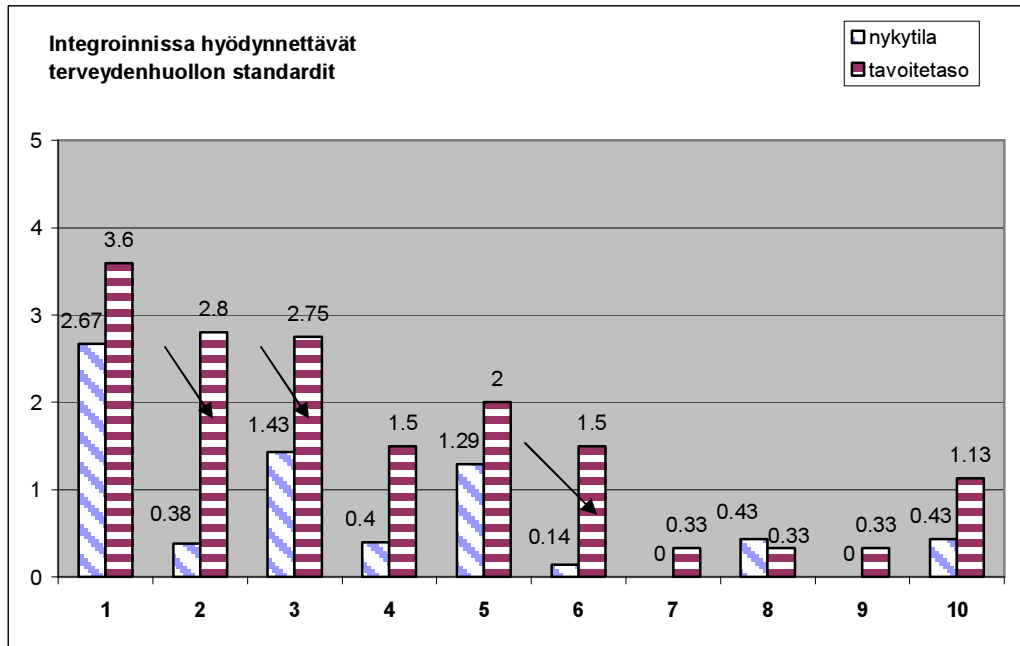


Kuva 17. Käytössä olevat integrointitavat ja -teknologiat

1. Komponenttipohjainen integrointi (useamman sovelluksen käyttämä komponentti, esimerkiksi potilaan tunnistus)
2. Väyläpohjainen integrointi (esimerkiksi CORBA broker)
3. Viestipohjainen integrointi (esimerkiksi HL7 -sanomarakenteita)
4. XML-pohjainen integrointi
5. Tiedon välitys yhteisen tietoalueen tai tietokannan kautta
6. Web service -tekniikka
7. Tietoverkon yli tapahtuva integrointi ja palvelujen käyttö
8. Samaan koneeseen asennettujen sovellusten välinen integrointi
9. Toisen sovelluksen käyttöliittymän kutsuminen
10. Käyttöliittymän tarjoaminen toiselle sovellukselle
11. Määräajoin eräsiirtona tapahtuvat tiedonsiirrot
12. Ulkoiseen standardimäärittelyyn (esimerkiksi HL7) perustuva integroinnin toteutus
13. Kahden väliseen sopimiseen perustuva integroinnin toteutus
14. Hallinnollisten taustarekisterien ja koodistojen käyttö ulkoisesta ohjelmistosta ohjelmointirajapinnan kautta

### 5.2.3 Integroinnissa hyödynnettävät terveydenhuollon standardit

Selvityksen mukaan käytetyin terveydenhuollon standardeista, nyt ja tavoitetilassa, on HL7 versio 2. CDA:ta (Clinical Document Architecture) ja DICOM:ia käytetään harvoin. Muiden standardien käyttö on satunnaista. HL7:n versio 3:n käyttö kasvaa merkittävästi. CDA:n käyttö tavoitetilassa on nykyistä yleisempää. Huomattavasti tasoaan kasvattaa CCOW. (Kuva 18)

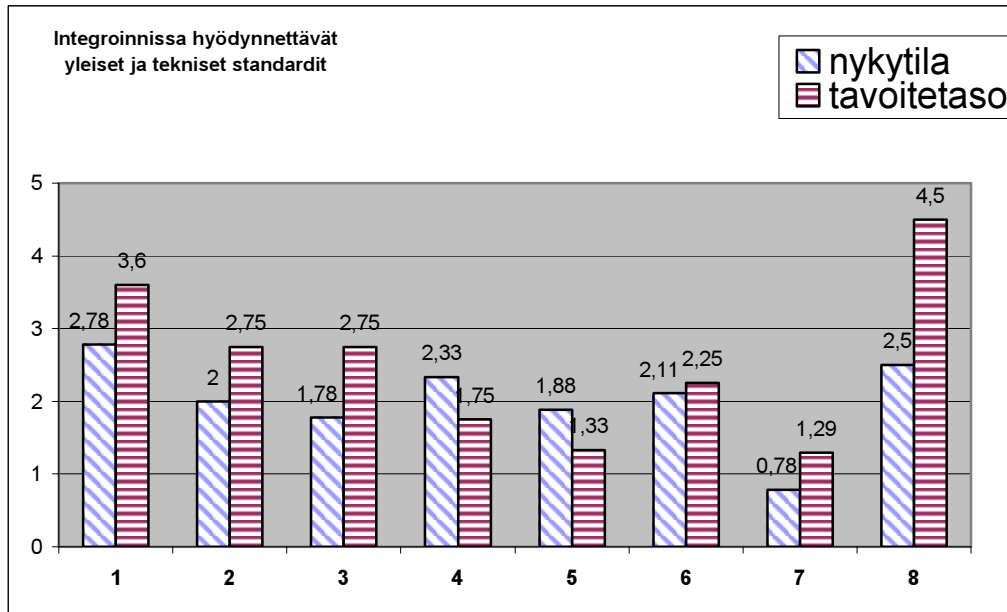


Kuva 18. Integroinnissa hyödynnettävät terveydenhuollon standardit

- |  |                                |
|--|--------------------------------|
| 1. HL7 versio 2                            | 6. CCOW                        |
| 2. HL7 versio 3                            | 7. Arden syntax                |
| 3. CDA (Clinical Document Architecture)    | 8. OMG Healthcare-standardit   |
| 4. IHE (Integrating Healthcare Enterprise) | 9. HISA                        |
| 5. DICOM                                   | 10. Muut CEN TC 251 standardit |

#### 5.2.4 Integroinnissa hyödynnettävät yleiset ja tekniset standardit

Käytetyin standardi on XML. Seuraavana tulevat muut standardit, joita ovat, vastaajien mukaan: EJB, Web services, SOAP, UDDI, WSDL ja J2EE. Muita vaihtoehtoina annettuja standardeja käytetään joskus tai harvemmin. Tavoitetilassa eniten käytettyjä ovat muut standardit, myös XML näyttää säilyttävän hyvin asemansa. XML Schema ja XML DTD ovat nykyistä käytetympiä. Muiden vaihtoehtoina annettujen standardien käyttö pysyy satunnaisena. (Kuva 19)

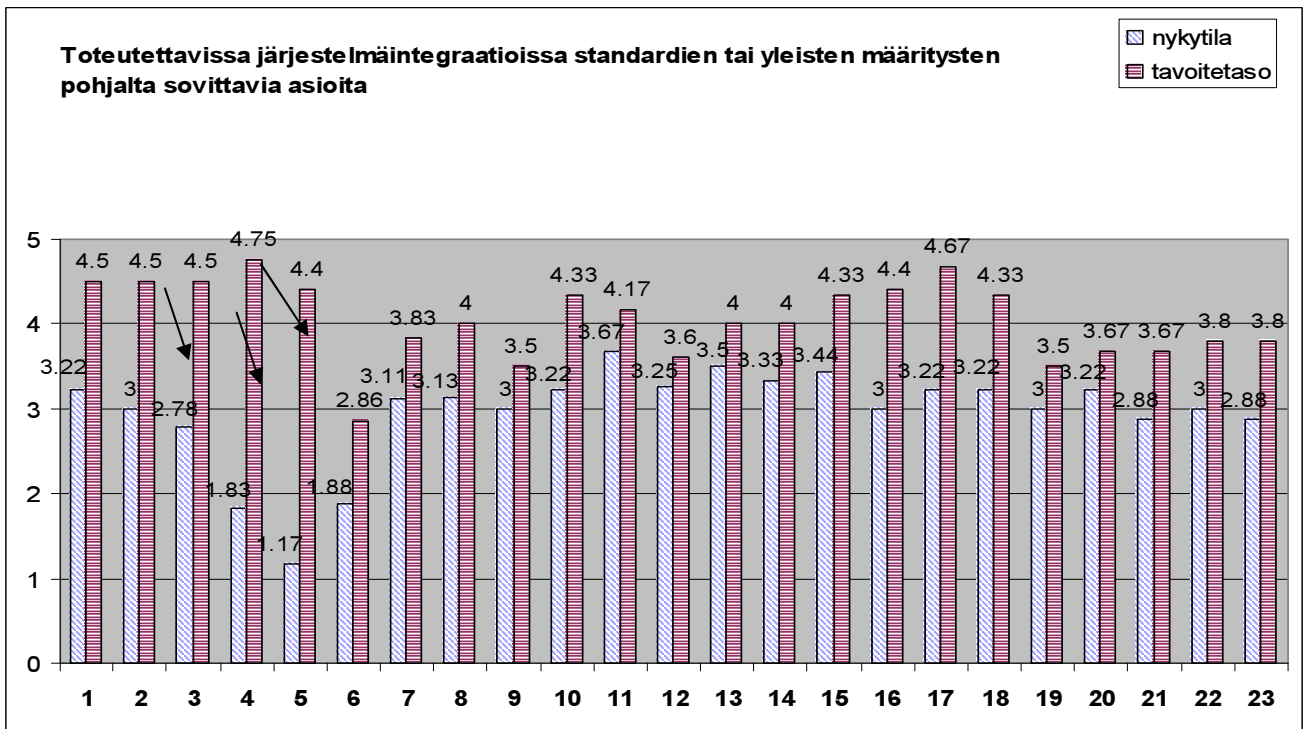


Kuva 19. Integroinnissa hyödynnettävät yleiset ja tekniset standardit

- |               |                    |
|---------------|--------------------|
| 1. XML        | 5. LDAP            |
| 2. XML Schema | 6. PKI-standardit  |
| 3. XML DTD    | 7. OMG (CORBA) IDL |
| 4. HTML       | 8. Muut standardit |

### 5.2.5 Toteutettavissa järjestelmäintegraatioissa standardien tai yleisten määritysten pohjalta sovittavat asiat

Selvityksen mukaan useimmiten sovittavia asioita ovat: tietoelementtien muoto ja pituus, käytettävät viestit tai operaatiot ja jokaisen ohjelmakutsun vastaus tai kuittaus. Tavoitetilassa useimmiten sovittavina asioina nähdään: määritellyt rajapinnat elektronisen potilaskertomuksen hallintaan ja virhetilanteiden käsittely. Seuraavina ovat: käyttäjän tunnistaminen, käyttöoikeuksien hallinta ja sisään kirjautumisen hallinta, tavoitetasona kaikilla "lähes aina". Suhteellisesti eniten tasoaan nostaa asiakkaan suostumuksen hallinta. Myös sopiminen määritellyistä rajapinnoista elektronisen potilaskertomuksen hallintaan yleistyy huomattavasti. Selvä tason lisäys on myös sisään kirjautumisen hallinnalla. Ainoa sovittava asia, joka jää usein-tason alle sekä nyt että tavoitetilassa on asiakasorganisaation organisaatioyksiköt ja organisaatorakenne. (Kuva 20)



Kuva 20. Toteutettavissa järjestelmäintegraatioissa standardien tai yleisten määritysten pohjalta sovittavia asioita

- |  |  |
|--|--|
| 1. Käyttäjän tunnistaminen   | 13. Käytettävät viestit tai operaatiot                         |
| 2. Käyttöoikeuksien hallinta   | 14. Viestien tai operaatioiden kutsumisessa käytetty tekniikka |
| 3. Sisään kirjautumisen hallinta                                     | 15. Tuleeko jokaisesta ohjelmakutsusta vastaus tai kuittaus    |
| 4. Määritellyt rajapinnat elektronisen potilaskertomuksen hallintaan | 16. Järjestelmien toiminnalliset vastuut                       |
| 5. Asiakkaan suostumuksen hallinta                                   | 17. Virhetilanteiden käsittely                                 |
| 6. Asiakasorganisaation organisaatioyksiköt ja organisaatorakenne    | 18. Tietoliikenne järjestelmien välillä                        |
| 7. Tarkka tietosisältö   | 19. Integroinnissa käytetyt ulkoiset tuotteet                  |
| 8. Tietosisällön eri osissa käytetyt koodistot                       | 20. Ratkaisun ylläpitovastuu                                   |
| 9. Tiedon esityksessä käytetty tekniikka                             | 21. Ratkaisun konfigurointi                                    |
| 10. Tietoelementtien merkitys  | 22. Verkkoliikenteen salausratkaisut                           |
| 11. Tietoelementtien muoto ja pituus                                 | 23. Tarvittava muu tekninen infrastruktuuri                    |
| 12. Tyhjiin tietojoukkojen tai -elementtien käsittely                |  |

### 5.2.6 Toteutettavissa järjestelmäintegraatioissa kahdenvälisesti toimittajien kesken sovittavat asiat (vaihtoehdot samat kuin edellisessä osiossa)

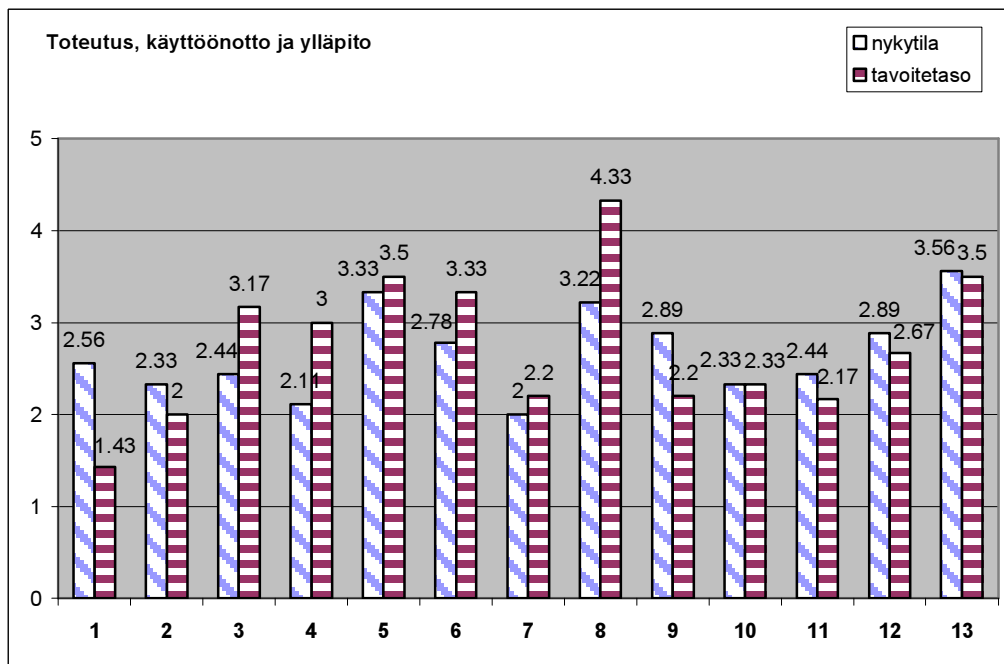
Kahdenvälisesti toimittajien kesken sovitaan useimmiten viestin tai operaatioiden kutsumisessa käytetystä tekniikasta, tietosisällön eri osissa käytetyistä koodistoista ja tarkasta tietosisällöstä. Tavoite-tilassa useimmiten sovitaan virhetilanteiden käsittelystä. Lähes aina sovitaan myös järjestelmien toiminnallisista vastuista, integroinnissa käytetyistä ulkoisista tuotteista, ratkaisun ylläpitovastuista ja ratkaisun konfiguroinnista. Suhteellisesti eniten taseaan kasvattaa määritellyt rajapinnat elektronisen potilaskertomuksen hallintaan. Taseaan selvästi kasvattavat myös virhetilanteiden käsittely ja asiakkaan suostumuksen hallinta.

## 5.2.7 Toteutettavissa järjestelmäintegraatioissa itsenäisesti tai asiakkaan vaatimusten mukaan soveltavia asioita (vaihtoehdot samat kuin standardien tai yleisten määrittelyjen pohjalta soveltavissa asioissa)

Itsenäisesti tai asiakkaan vaatimusten mukaan yleisimmin soveltavia asioita ovat: ratkaisun ylläpito-vastuu, tietoliikenne järjestelmien välillä sekä ratkaisun konfigurointi. Tarkasta tietosisällöstä sopiminen yleistyy tavoitetilassa. Tietoliikenne järjestelmien välillä pysyttelee myös tavoitetilassa tärkeänä soveltavana asiana. Tavoitetason "lähes aina" saavuttivat seuraavat asiat: käyttäjän tunnistaminen, käyttöoikeuksien hallinta, sisään kirjautumisen hallinta, tietosisällön eri osissa käytetyt koodit, virhetilanteiden käsittely ja ratkaisun ylläpito-vastuu. Suhteellisesti eniten tasoaan kasvattaa määritellyt rajapinnat elektronisen potilaskertomuksen hallintaan. Selvästi tasoaan kasvattivat myös asiakkaan suostumuksen hallinta ja tarkka tietosisältö.

## 5.2.8 Toteutus, käyttöönotto ja ylläpito

Sekä nykytilassa että tavoitetilassa toimittaja vastaa ohjelmiston uusien versioiden asentamisesta. Ohjelmisto(n osa) hyödyntää usein asiakas-palvelin ympäristöä. Usein ohjelmisto asennetaan asiakkaalle sellaisenaan, suuntaus on asentaa ohjelmisto asiakkaalle sellaisenaan lähes aina. Tavoitetilassa vältetään saman toiminnallisuuden tai tiedon toteutusta erikseen useaan eri ohjelmistoon. (Kuva 21)



Kuva 21. Toteutus, käyttöönotto ja ylläpito

1. Sama toiminnallisuus tai tieto on toteutettu erikseen useaan eri ohjelmistoon
2. Ohjelmisto nojautuu suoraan muilta ohjelmistoilta saatavaan tietoon tai toiminnallisuuteen
3. Ohjelmisto hyödyntää usean sovelluksen käyttämää integraatioalustaa
4. Ohjelmisto(n osa) hyödyntää keskuskoneympäristöä
5. Ohjelmisto(n osa) hyödyntää asiakas-palvelin ympäristöä
6. Ohjelmisto(n osa) hyödyntää web-ympäristöä
7. Ohjelmisto toimii itsenäisesti työasemalla

8. Ohjelmisto asennetaan sellaisenaan asiakkaalle
9. Ohjelmiston asennus vaatii asiakkaan suorittamaa konfigurointityötä
10. Ohjelmiston asennus vaatii toimittajan suorittamaa konfigurointityötä
11. Ohjelmiston käyttöönotto vaatii toimittajan ja kumppanin tekemää integrointia
12. Ohjelmistoon tehdään asiakkaan toivomia lisäpiirteitä käyttöönoton jälkeen
13. Toimittaja vastaa ohjelmiston uusien versioiden asentamisesta

Vastaajien mielestä (n=5) tärkeimmät yleiset koodistot sovelluksissa ovat:

- |  |                          |
|--|--------------------------|
| - YSA  | - käyttäjä               |
| - SOSA   | - käyttäjän organisaatio |
| - FINMESH  | - käyttäjän oikeudet     |
| - ne, jotka liittyvät sähköiseen potilaskertomukseen | - ICD 9, 10              |
| - potilas  | - diagnoosi              |
| - organisaatio                                       | - toimenpiteet           |
| - toimenpide   | - erikoisalajat          |
| - tutkimus   | - kunnat                 |

Vastaajien (n=6) mielestä tärkeimmät yhteiskäyttöiset rekisterit sovelluksissa ovat:

- |                       |                |
|-----------------------|----------------|
| - käyttäjärekisteri   | - käyttäjät    |
| - potilasrekisteri    | - diagnoosi    |
| - LDAP                | - potilas      |
| - käyttäjän tunnistus | - organisaatio |
| - lääkärit            |                |

Vastaajien (n=3) mielestä ohjelmistopalveluita, joita sovellusten tekijöille pitäisi tarjota ohjelmointirajapintoina ovat:

- potilaan tunnistus
- käyttäjän hallinta
- käyttäjän tunnistus
- yhteiskäyttöisten rekisterien ja koodistojen käyttö
- J2EE

Vastaajien (n=8) mielestä integroinnin suurimpia ongelmia ovat:

- |  |  |
|--|--|
| - asiakkaan ympäristöt aina erilaisia        | - kaupalliset syyt                             |
| - integroitavien järjestelmien erilaisuus    | - yhteistyön puute                             |
| - hankaluudet neuvotella toteutustapa        | - testauksen raskaus                           |
| - työn määrä suhteessa saatavaan korvaukseen | - vähäinen osaaminen                           |
| - sanomien määrittely                        | - työpöytäintegraatio-standardien kypsyttömyys |
| - standardit / standardoinnin puute          |  |
| - lainsäädäntö                               |  |

Vastaajien (n=4) mielestä integroinnin kehittämiskohteita ovat:

- työpöytäintegraation standardien ja toimintamallien vakiinnuttaminen
- sähköinen potilaskertomus
- web service tekniikka
- palvelupohjaisuus sanoman välityksen sijaan

Toteutuksen suurimpina ongelmina mainittiin (n=2):

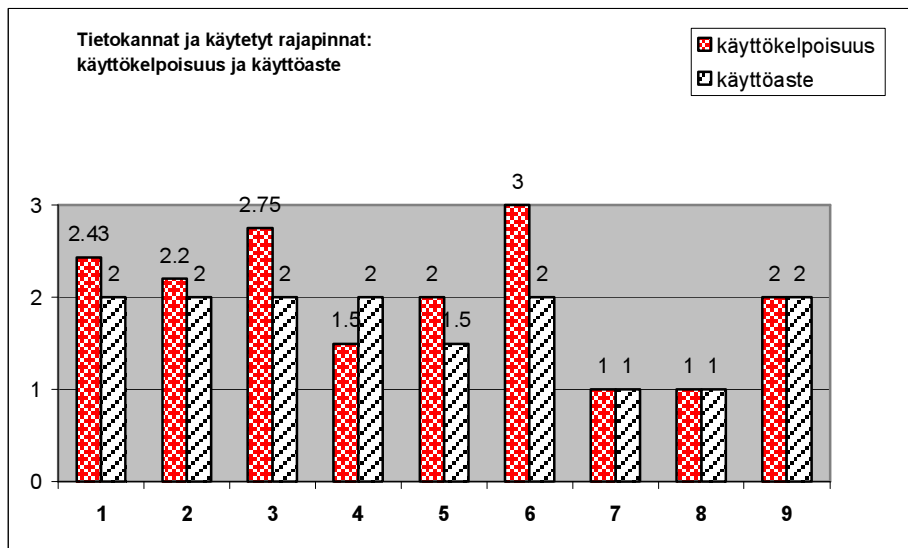
- liian tiukat aikataulut
- resurssipula

### 5.2.9 Käytössä olevat välineet ja menetelmät

Vastaajat saivat itse nimetä käytössään olevat välineet ja menetelmät. Tämän jälkeen heidän tuli arvioida välineiden ja menetelmien käyttökelpoisuutta ja käyttöastetta. Seuraavassa on esitetty käytössä olevat välineet ja menetelmät ryhmiteltynä käyttötarkoituksen mukaisesti. (Välineen ja menetelmän perässä oleva numero kertoo kuinka moni on sen maininnut.)

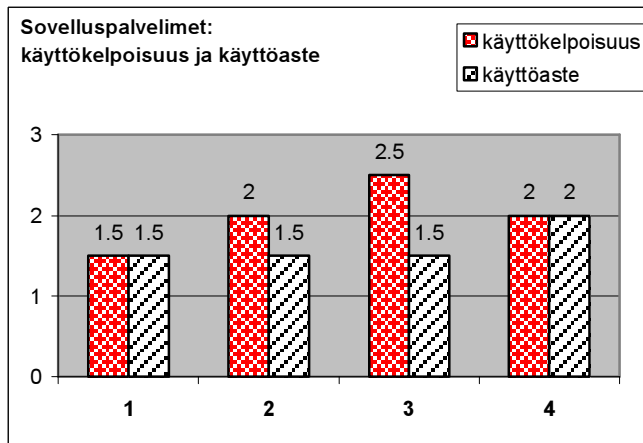
Käyttökelpoisuuden asteikko:	Käyttöasteen asteikko:
0=kelvoton	0=satunnaista
1=menettelee	1=Joskus
2=hyvä	2=usein
3=erinomainen	3= aina

#### 5.2.9.1 TIETOKANNAT JA KÄYTETYT RAJAPINNAT



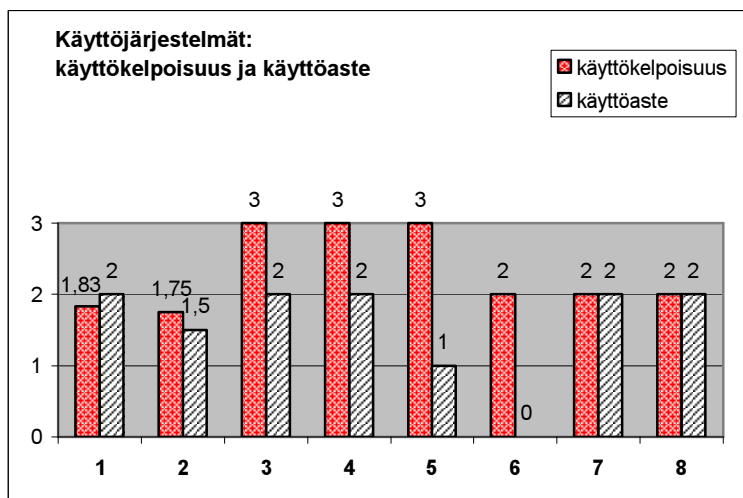
- |  |                |
|--|----------------|
| 1. Oracle, Oracle OCI, 8/9 & 8.1.7 (7) | 6. JDBC (2)    |
| 2. OpenM, Mumps & Cache (5)            | 7. Paradox (1) |
| 3. MS SQL Server (4)                   | 8. Access (1)  |
| 4. InterBase (2)                       | 9. FF (1)      |
| 5. My SQL (2)                          |                |

## 5.2.9.2 SOVELLUSPALVELIMET



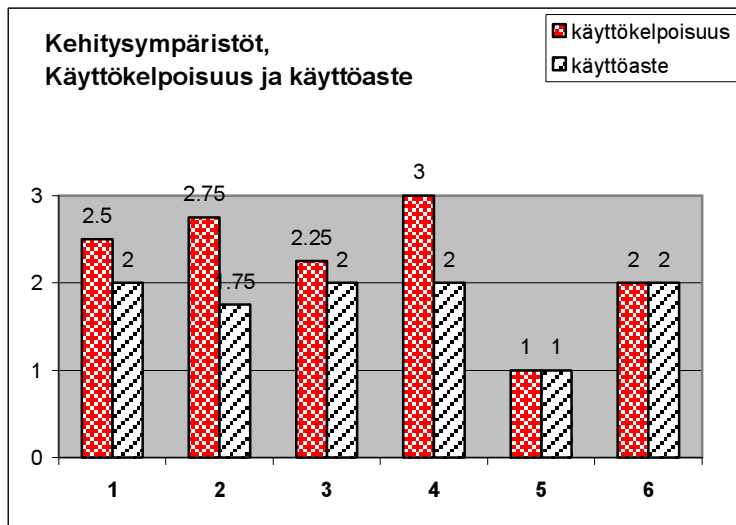
1. JBoSS (2)
2. Oracle IAS (2)
3. BEA Weblogic server (2)
4. WebSphere (2)

## 5.2.9.3 KÄYTTÖJÄRJESTELMÄT



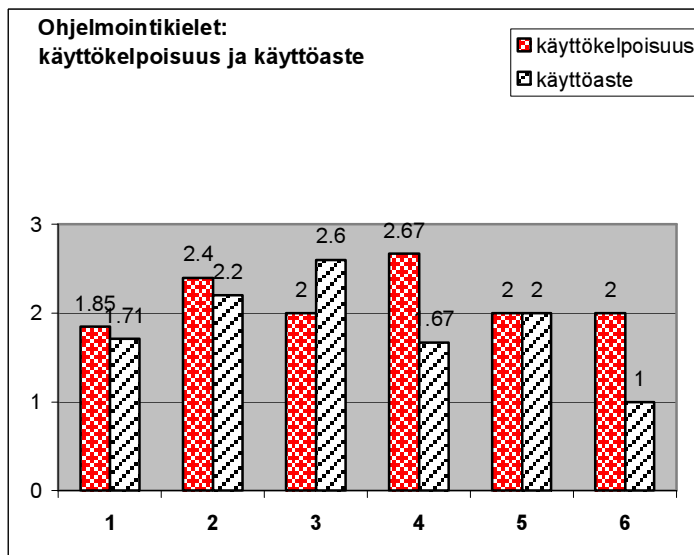
- |                                  |                |
|----------------------------------|----------------|
| 1. Windows NT, XP, 2000, NT4 (8) | 5. Aix (1)     |
| 2. Linux (5)                     | 6. OS/2 (1)    |
| 3. Unix (2)                      | 7. HP UX (1)   |
| 4. Open VMS (1)                  | 8. Solaris (1) |

#### 5.2.9.4 KEHITYSYMPÄRISTÖT



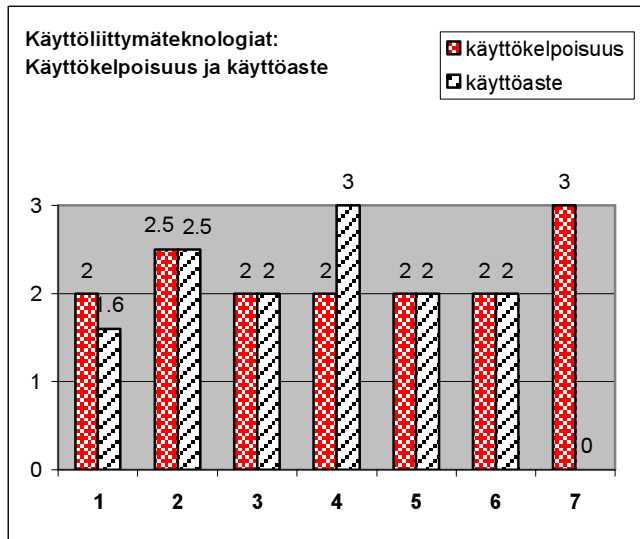
1. MS Visual Studio (sis. VB.NET) (6)
2. Delphi (4)
3. JBuilder (4)
4. JDeveloper (2)
5. Borland c++ (1)
6. Visual Age for Java (1)

#### 5.2.9.5 OHJELMOINTIKIELET



1. C/C++ (7)
2. Java (5)
3. Visual Basic (5)
4. Object Pascal (4)
5. M-kieli (2)
6. C# (1)

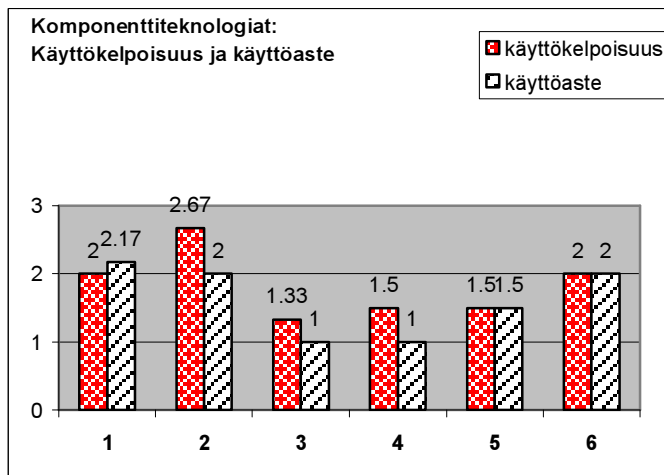
## 5.2.9.6 KÄYTTÖLIITTYMÄTEKNOLOGIAT



1. JSP (5)
2. HTML (4)
3. ASP (3)
4. Windows (2)

5. Swing (Java) (1)
6. MSF (1)
7. CSP (1)

## 5.2.9.7 KOMPONENTTITEKNOLOGIAT



1. DLL (6)
2. EJB (3)
3. CORBA (3)
4. JavaBeans (2)
5. COM (2)
6. .NET (1)

#### 5.2.9.8 MUUT VÄLIOHJELMISTOT

Muina väliohjelmistoina mainittiin TUXEDO (2) ja Terminal Service, jonka oli maininnut yksi vastaaja. Käyttökelpoisuus oli molemmilla "erinomainen". TUXEDO:n käyttöaste oli "usein" ja Terminal Service:n "aina".

#### 5.2.9.9 MALLINNUS- JA SUUNNITTELUYÖKALUT

Mallinnus- ja suunnittelutyökaluista kolme vastaajaa oli maininnut Rational Rosen. MS Visionia käytti kaksi vastaajaa. System Architect ja ABC Flowcharter mainittiin yhden kerran. Rational Rosen käyttökelpoisuus oli "hyvä" ja käyttöaste "usein". MS Visionin käyttökelpoisuus oli "menettelee" ja käyttöaste "joskus". System Arcitectin käyttökelpoisuus oli "hyvä". ABC Flowcharterin käyttökelpoisuus oli "menettelee". Molempien käyttöaste oli "joskus".

#### 5.2.9.10 RAPORTOINTIVÄLINEET

Raportointivälineinä mainittiin QuickReport, Cognos Impromptu ja Crystal Report. Näistä jokainen sai yhden maininnan. QuickReport:n käyttökelpoisuus oli "hyvä" ja käyttöaste "joskus". Crystal Report:n käyttökelpoisuus oli "erinomainen" ja käyttöaste "usein". Cognos Impromptu:lle ei ollut annettu arvoja.

#### 5.2.9.11 MUUTOSPYYNTÖJEN HALLINTA

Muutospyyntöjen hallintavälineenä yksi vastaaja mainitsi Requestin. Käyttökelpoisuutena oli "hyvä" ja käyttöasteena "aina".

#### 5.2.9.12 VERSION JA KOKOONPANON HALLINTA

Version ja kokoonpanon hallintavälineinä mainittiin Source Safe (2), PVCS ja CVS. PVCS ja CVS oli mainittu kerran. Source Safe:n käyttökelpoisuus 2,5 ja käyttöaste "aina". CVS:n käyttökelpoisuus "hyvä" ja käyttöaste "usein". PVCS:n kohdalla ei ollut annettu arvoja.

#### 5.2.9.13 KOODIN ANALYSOINTI

Koodin analysointivälineinä mainittiin Bound Checker (käyttökelpoisuus "erinomainen" ja käyttöaste "joskus"), Aivosto VB Analyzer (käyttökelpoisuus "hyvä", käyttöastetta ei ollut määritelty), sourceforge (käyttökelpoisuus "menettelee", käyttöastetta ei ollut määritelty), Optimizer (käyttökelpoisuus "hyvä" ja käyttöaste "usein") ja Jtest (käyttökelpoisuus "hyvä" ja käyttöaste "joskus"). Kukin väline mainittiin yhden kerran.

#### 5.2.9.14 SUORITUSKYVYN OPTIMOINTI

Suorituskyvyn optimointiin käytettävänä välineinä mainittiin SQL Plus trace, Webstress ja Optimizer. Kukin väline mainittiin yhden kerran. SQL Plus trace:n ja Optimizer:n käyttökelpoisuus oli "hyvä" ja käyttöaste "usein". Webstress:n käyttökelpoisuus oli "menettelee" ja käyttöaste "usein".

#### 5.2.9.15 KEHITYSMETODOLOGIAT

Kehitysmetodologioista RUP mainittiin yhden kerran. Käyttökelpoisuutena oli "erinomainen" ja käyttöasteena "usein".

#### 5.2.9.16 SUUNNITTELMALLIT

Suunnittelumalleista mainittiin mallit yleensä ja J2EE Design patterns. Molemmat mainittiin yhden kerran. Molemmilla oli käyttökelpoisuutena "erinomainen" ja käyttöasteena "usein".

#### 5.2.9.17 RYHMÄTYÖN HALLINTAOHJELMISTOT

Ryhmätyön hallintaohjelmistoista mainittiin Pro Intra (2), Lotus Notes ja MS Source Safe. Lotus Notes ja MS Source Safe oli mainittu kerran. Pro Intra:n käyttökelpoisuus oli "erinomainen" ja käyttöaste "aina". Lotus Notes:n käyttökelpoisuus oli "menettelee" ja käyttöaste "usein". MS Source Safe:n käyttökelpoisuus oli "hyvä" ja käyttöaste "usein".

#### 5.2.9.18 PROJEKTIN HALLINTAOHJELMISTOT

Projektin hallintaohjelmistoista mainittiin MS Project (4) ja MS Source Safe, yksi maininta. Molempien käyttökelpoisuus oli "hyvä" ja käyttöaste "usein".

Vastaajien mukaan yksikköjen ohjelmistojen ensisijaisia kohdealustoja ovat:

- sovelluspalvelin ja sen päälle rakennetut lisäratkaisut (portaalit / integraatioserveri)
- Windows
- Linux
- PDA

### 5.3 Pohdintaa

Kyselyn perusteella valmiskomponentteja käytetään joskus tai usein. Tavoitteena on lisätä valmiskomponenttien käyttöä. Huomattavaa on kuitenkin se, ettei tavoitetilassa tavoitella edes "lähes aina"-tasoa.

Integrointitavoissa ja -teknologioissa on selvästi tavoitteena lisätä eri tapojen ja teknologioiden käyttöä. Tällä hetkellä useassa ollaan "joskus"-tasolla tai sen alla, mutta pyrkimyksenä on päästä "usein"-tasolle. Kyselyn perusteella tavoitetilassa selvästi käytetyimpiä ovat XML-pohjainen integrointi ja komponenttipohjainen integrointi, joiden käyttö tavoittelee jo lähes aina -tasoa. Huiman nousun tekee Web Service -tekniikka, joka kipuaa "harvoin"-tasolta "usein"-tasolle saakka.

Integroinnissa hyödynnettävien standardien käyttö ei kyselyn mukaan ole kovinkaan yleistä. Tavoitetilassakaan standardien käyttö ei näyttäisi mitenkään merkittävästi lisääntyvän. Poikkeuksena on terveydenhuollon standardeista HL7. Hämmästyttä herättää se, että integroinnin yhtenä suurimmista ongelmista vastaajat kokevat juuri standardien puutteen tai niiden puutteellisen määrittelyn.

# 6 TIETOJÄRJESTELMÄN KÄYTTÖÖNOTTO

## 6.1 Yleistä tietojärjestelmän käyttöönotosta

Käyttöönotto on hallittu vaihe, jossa asiakas ottaa käyttöön valmiin järjestelmän. Myös käyttöönotto toteutetaan useimmiten projektina ja se on kohdeorganisaation kannalta ainutkertainen tapahtuma. Isossa organisaatiossa uuden tietojärjestelmän käyttöönotto voi olla vuosia kestävä hanke. Iso hanke on syytä pilkkoa pienemmiksi osaprojekteiksi, jotta hankkeen läpivienti tapahtuu mahdollisimman hallitusti.

Järjestelmän käyttöönottoprojektiin osallistuvat: kohdeorganisaation edustajat, järjestelmän toimittaja sekä mahdollisesti organisaation oma tietotekniikka-yksikkö. Jokaisella toimijalla on käyttöönotossa omat vastualueensa. Vastuut vaihtelevat projektikohtaisesti. Kohdeorganisaation vastuulla on esimerkiksi toimituksen hyväksyminen, toimittajan vastuulla on ohjelmistoasennukset ja tuotteen ylläpito, tietotekniikka-yksikkö vastaa perustietotekniikan suunnittelusta ja järjestelmänhallinnasta.

Käyttöönottoon vaikuttavat muun muassa jo käytössä oleva teknologia ja tietojärjestelmät. Terveystieteiden organisaation tietojärjestelmä muodostuu usein useista tietojärjestelmistä, jotka on toteutettu eri aikaan ja erilaisilla teknologioilla. Uuden järjestelmän käyttöönottovaiheessa tämä vaatii paljon työtä järjestelmien välisten liittymien toiminnan varmistamiseksi.

Jotta uudesta järjestelmästä saadaan kaikki hyöty irti, pitää sitä myös osata käyttää. Koulutus on tärkeä osa käyttöönottoa. Koulutuksesta sovitaan toimittajan kanssa ja sen toteutus vaihtelee projektikohtaisesti. Olennainen osa koulutusta on kattava ja selvä järjestelmän käyttöohje, jonka käyttämiin tarvittaessa myös opastetaan. Koulutus voi myös olla käyttöönoton hankalin vaihe. Järjestelmän käyttäjät voivat pitää koulutusta riittämättömänä.

Yksi käyttöönoton ongelma on järjestelmän testaaminen etukäteen siinä laajuudessa kuin se otetaan käyttöön, mistä seuraa se, ettei voida tietää käyttöönoton laajamittaisia vaikutuksia. Lisäksi ongelmana voi olla liian tiukka aikataulu, puutteellinen koulutus sekä muutosvastarinta.

## 6.2 Kyselyn tulokset

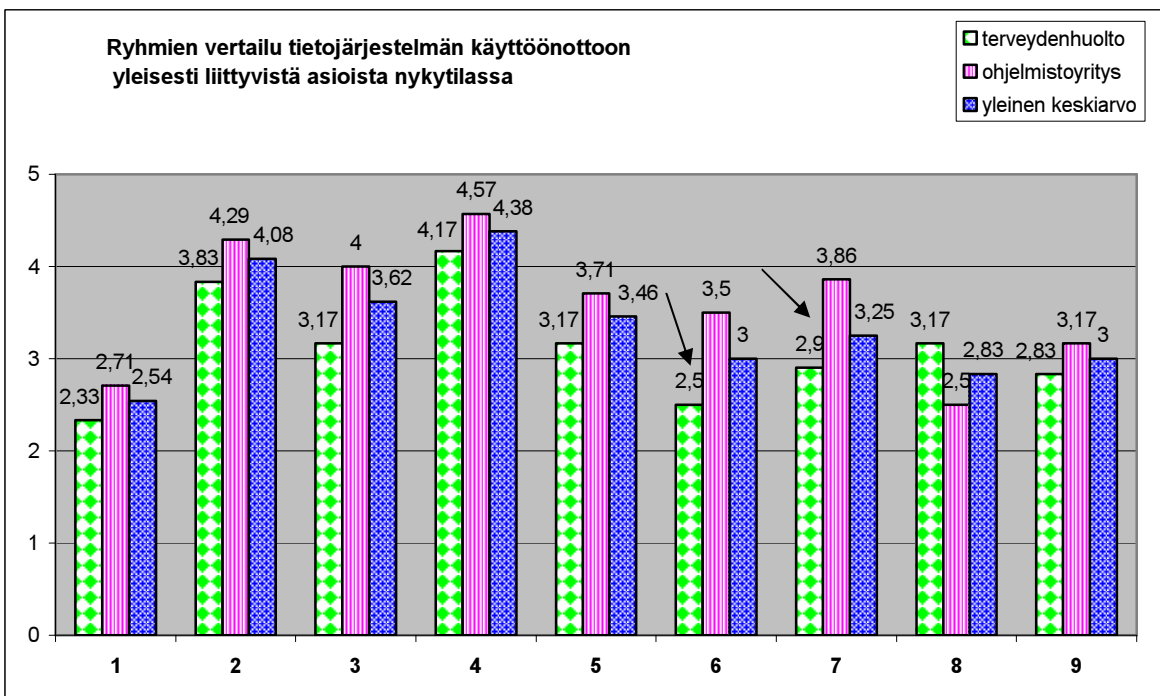
Tietojärjestelmän käyttöönottoon liittyvissä kysymyksissä vastausfrekvenssi vaihteli yhdeksän ja kolmentoista välillä. Kaikki vastaajat eivät merkinneet tavoitetasoa. Vastauksista seitsemän oli ohjelmistoyrityksistä. Sairaalan atk-osastolta, erikoissairaanhoidosta ja perusterveydenhuollosta vastauksia oli kaksi kustakin yksiköstä. Terveyskeskuksen atk-osastolta oli yksi vastaus. Seuraavana esitetään kyselyn tulokset osa-alueittain.

Käytetty asteikko oli: 0 = Ei koskaan, 1 = Harvoin, 2 = Joskus, 3 = Usein, 4 = Lähes aina, 5 = Aina. Kuvissa olevilla nuolilla osoitetaan ne kohdat, joissa ryhmien väliset erot ovat suurimmat.

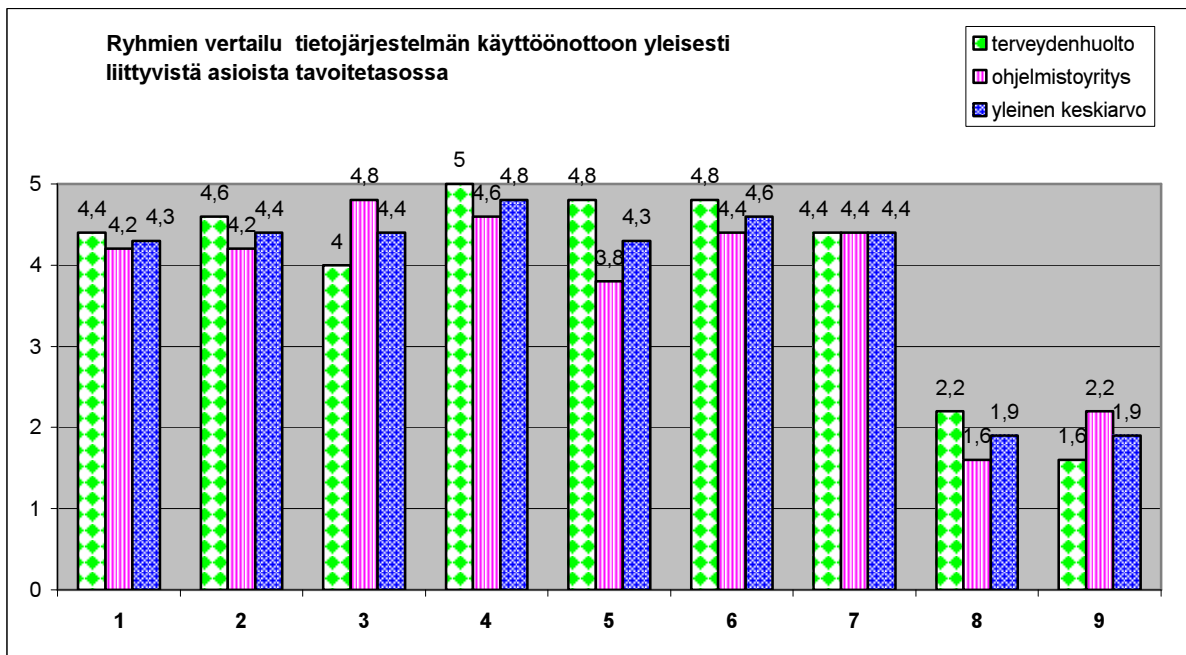
## 6.2.1 Tietojärjestelmän käyttöönottoon yleisesti liittyviä asioita

Kuvissa 22 ja 23 on vertailtu terveydenhuoltoalalla ja ohjelmistoyrityksissä työskentelevien henkilöiden näkemyksiä tietojärjestelmän käyttöönottoon yleisesti liittyvistä asioista sekä nykytilassa että tavoitetasossa. Ohjelmistoyrityksissä työskentelevien mielestä käyttöönotossa ollaan kautta linjan vähän paremmalla tasolla kuin terveydenhuoltoalalla työskentelevien mielestä. Kuvia vertaamalla huomataan, että sekä terveydenhuoltoalalla että ohjelmistoyrityksissä työskentelevien mielestä käyttöönoton kaiken kaikkiaan tulisi olla huomattavasti nykyistä sujuvampaa.

Suurin mielipide-ero ryhmien välillä on käyttöönottoon varattavien henkilöresurssien määrässä ja tietojen siirron onnistumisessa vanhasta järjestelmästä uuteen. Tavoitetasossa tämä ero tasoittuu. Molempien ryhmien mielestä tavoitetilassa käyttöönottovaiheessa järjestelmään tehtävät muutokset vähenevät nykyiseltä tasolta. Myös käyttöönoton aiheuttamat käyttökatkokset vähenevät tavoitetilassa.



Kuva 22. Ryhmien vertailu tietojärjestelmän käyttöönottoon yleisesti liittyvistä asioista, nykytila

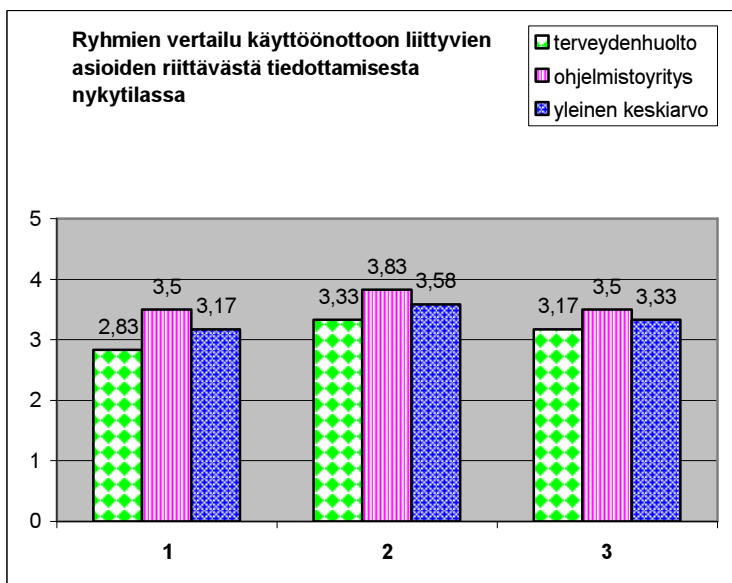


Kuva 23. Ryhmien vertailu tietojärjestelmän käyttöönottoon yleisesti liittyvistä asioista, tavoitetaso

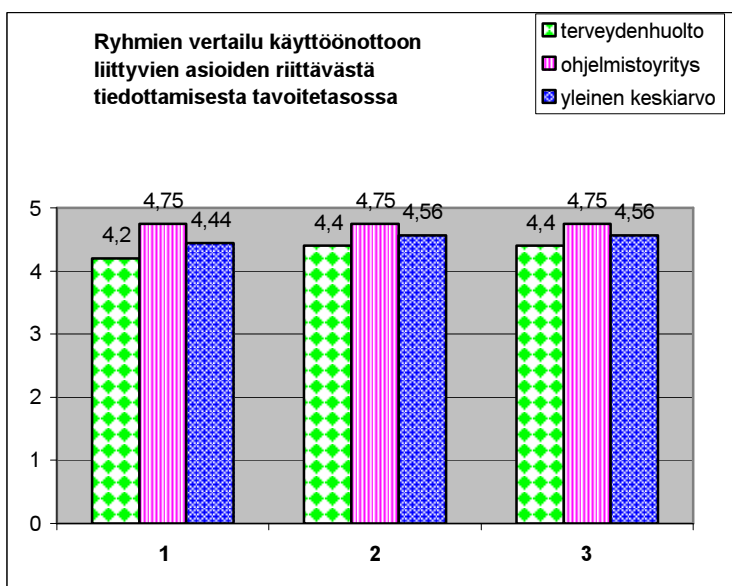
1. Käyttöönotto on kaiken kaikkiaan sujuvaa
2. Käyttöönotto toteutetaan projektina
3. Ennen käyttöönottoa järjestelmää koekäytetään
4. Käyttöönotosta laaditaan suunnitelma
5. Käyttäjät ovat mukana päätöksenteossa
6. Käyttöönottoon varataan sopivasti henkilöresursseja,
7. Tietojen siirto vanhasta järjestelmästä uuteen onnistuu
8. Käyttöönottovaiheessa järjestelmään tehdään muutoksia
9. Käyttöönotto aiheuttaa käyttökatkoksia

## 6.2.2 Käyttöönottoon liittyvistä asioista tiedottaminen

Kuvissa 24 ja 25 on vertailtu terveydenhuoltoalalla ja ohjelmistoyrityksissä työskentelevien henkilöiden näkemyksiä käyttöönottoon liittyvien asioiden tiedottamisesta sekä nykytilassa että tavoitetasossa. Molempien ryhmien mielestä käyttöönottoon liittyvistä asioista tiedottamisessa on selvästi parantamisen varaa. Ryhmien välillä on havaittavissa pieniä eroja sekä nykytilassa että tavoitetasossa. Suurin ero ryhmien välillä on käyttöönottoon liittyvistä asioista tiedottamisessa ennen käyttöönottoa.



Kuva 24. Ryhmien vertailu käyttöönottoon liittyvien asioiden riittävästä tiedottamisesta, nykytila



Kuva 25. Ryhmien vertailu käyttöönottoon liittyvien asioiden riittävästä tiedottamisesta, tavoitetaso

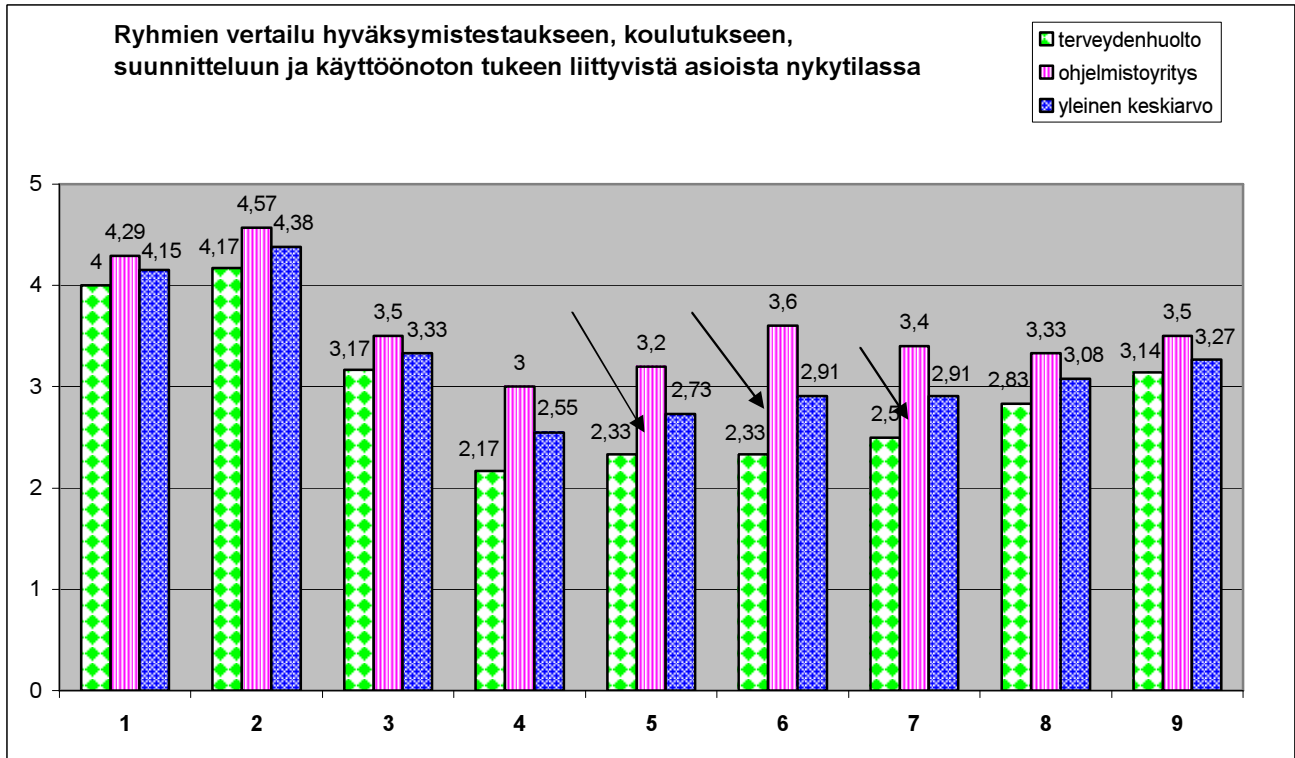
1. Käyttöönottoon liittyvistä asioista tiedotetaan riittävästi ennen käyttöönottoa
2. Käyttöönottoon liittyvistä asioista tiedotetaan riittävästi käyttöönoton aikana
3. Käyttöönottoon liittyvistä asioista tiedotetaan riittävästi käyttöönoton jälkeen

### 6.2.3 Hyväksymistestaukseen, koulutukseen, suunnitteluun ja käyttöönoton tukeen liittyviä asioita

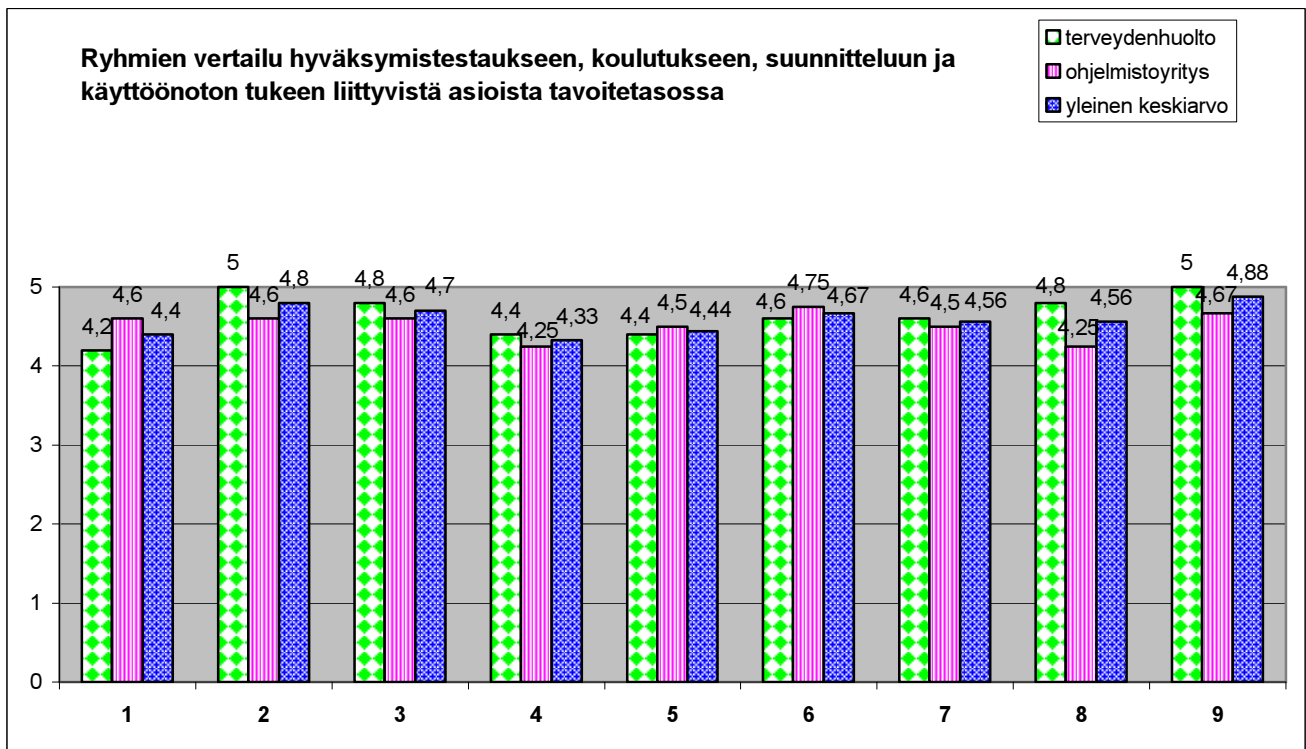
Kuvissa 26 ja 27 on vertailtu terveydenhuoltoalalla ja ohjelmistoyrityksissä työskentelevien henkilöiden näkemyksiä hyväksymistestaukseen, koulutukseen, suunnitteluun sekä käyttöönoton tukeen liittyvistä asioista sekä nykytilassa että tavoitetilassa. Terveydenhuoltoalalla työskentelevät arvioi-

vat nykytilan huonommaksi kuin ohjelmistoyrityksissä työskentelevät. Kuvia vertaamalla huomataan, että molempien ryhmien mielestä tavoitetilassa pitää kiinnittää enemmän huomiota asioihin, jotka liittyvät hyväksymistestaukseen, koulutukseen, työmäärän ja ajankäytön suunnitteluun sekä käyttöönoton tukeen. Tavoitetasona on "melkein aina" tai peräti "aina".

Tavoitetilassa ryhmien väliset erot eivät ole suuria. Eniten parannettavaa molemmilla ryhmillä on siirtymävaiheeseen suunnitellun työmäärän toteutumisessa.



Kuva 26. Ryhmien vertailu hyväksymistestaukseen, koulutukseen, suunnitteluun ja käyttöönoton tukeen liittyvistä asioista, nykytila



Kuva 27. Ryhmien vertailu hyväksymistestaukseen, koulutukseen, suunnitteluun ja käyttöönoton tukeen liittyvistä asioista, tavoitetaso

1. Järjestelmälle tehdään hyväksymistestaus
2. Hyväksymistestaus perustuu järjestelmälle asetettuihin vaatimuksiin
3. Koulutusta järjestetään riittävästi kaikille järjestelmän käyttäjille
4. Siirtymävaiheen toteutunut työmäärä vastaa suunniteltua
5. Siirtymävaiheeseen käytetty aika vastaa suunniteltua
6. Käyttöönotontuki on riittävää
7. Käyttöönottoon kuluva kalenteriaika on sopiva
8. Ohjelmiston valmistaja tukee käyttöönottoa riittävästi
9. Toimittajan käyttöönoton jälkeinen tuki on riittävää

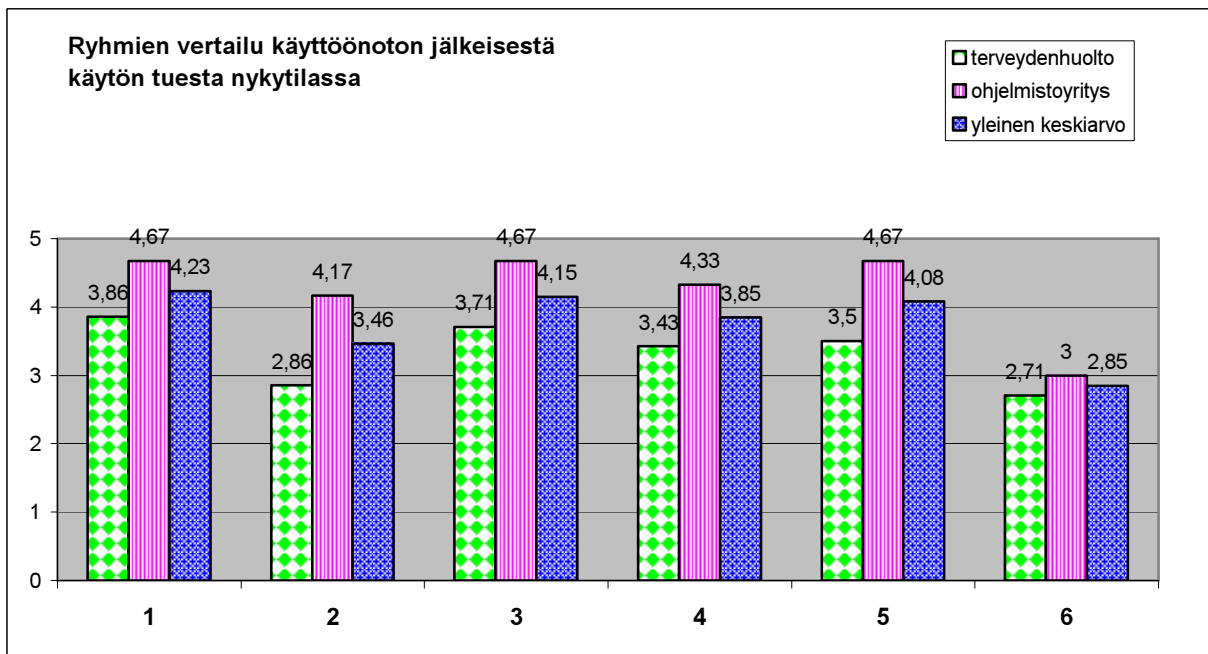
#### 6.2.4 Käyttöönottovaiheen ongelmatilanteiden hoito

Sekä terveydenhuoltoalalla että ohjelmistoyrityksissä työskentelevien mielestä käyttöönottovaiheen ongelmatilanteissa oikea ihminen on helppo tavoittaa ja ratkaisu ongelmaan löytyy riittävän nopeasti. Tavoitteena kuitenkin on, että nykyiseltä "usein"-tasolta päästään "lähes aina" -tasolle asti.

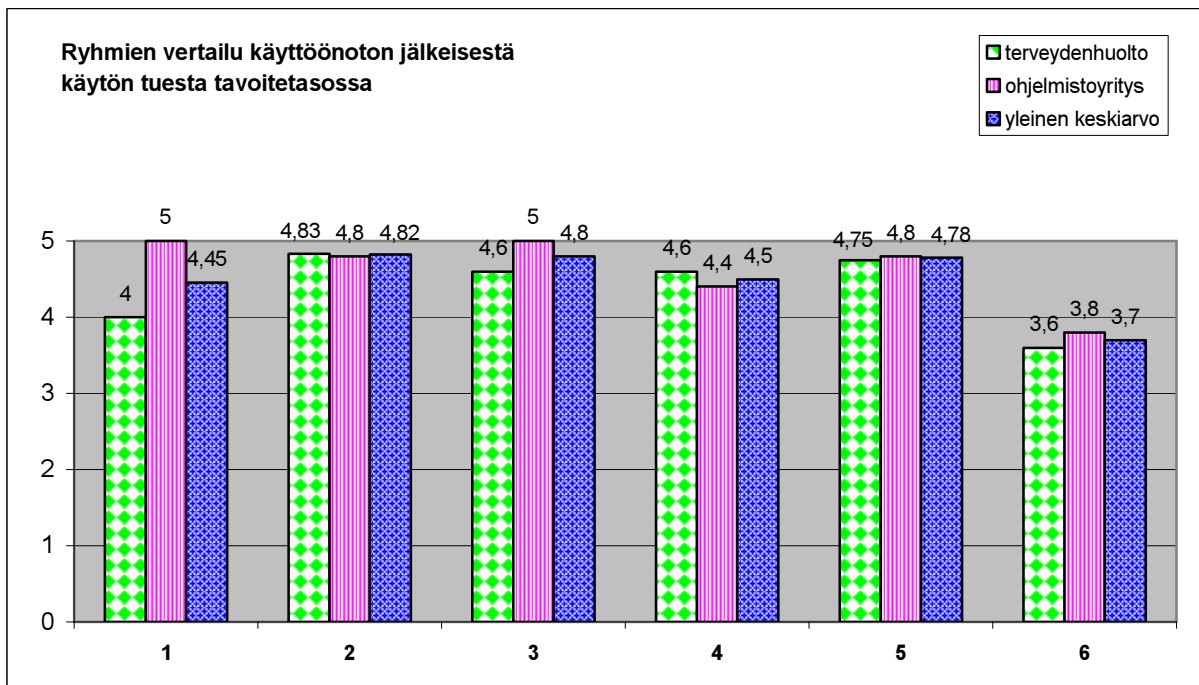
#### 6.2.5 Käyttöönottovaiheen jälkeinen tuki

Kuvissa 28 ja 29 on vertailtu terveydenhuoltoalalla ja ohjelmistoyrityksissä työskentelevien henkilöiden näkemyksiä käyttöönottovaiheen jälkeisestä tuesta sekä nykytilassa että tavoitetasossa. Käyttöönoton jälkeisessä tuessa ollaan ohjelmistoyritysten mielestä jo nyt "lähes aina" -tasolla tai lähellä "aina" -tasoa. Terveystieteiden alan työntekijöiden mielestä ei olla aivan niin korkealla. Tavoitetasona molempien ryhmien mielestä tulisi olla "lähes aina" tai "aina" -taso. Sekä terveydenhuolto-

alan että ohjelmistoyritysten henkilöstön mielestä sähköisen käyttöohjeen merkitys lisääntyy. Molempien ryhmien mielestä henkilökohtainen ohjaus on tukimuodoista vähiten käytetyin. Henkilökohtaisen tuen toivotaan lisääntyvän nykyiseltä tasolta.



Kuva 28. Ryhmien vertailu käyttöönoton jälkeisestä käytön tuesta, nykytila



Kuva 29. Ryhmien vertailu käyttöönoton jälkeisestä käytön tuesta, tavoitetaso

1. Kirjallinen käyttöohje
2. Sähköinen käyttöohje
3. Helpdesk
4. Puhelinpalvelu
5. Sähköpostipalvelu
6. Henkilökohtainen ohjaus

### 6.2.6 Käyttöönottovaiheen jälkeisten ongelmatilanteiden hoito

Sekä terveydenhuoltoalalla että ohjelmistoyrityksissä työskentelevien mielestä käyttöönottovaiheen jälkeisten ongelmatilanteiden hoidossa ollaan "usein"-tasolla. Ongelmatilanteissa oikea ihminen on helppo tavoittaa ja ratkaisu ongelmaan löytyy riittävän helposti. Molempien ryhmien mielestä käyttöönottovaiheen jälkeisten ongelmien hoidon tavoitetilaksi tulisi olla nykyistäkin parempi, vähintään "lähes aina" -tasolla.

## 6.3 Pohdintaa

Kyselyn perusteella voidaan todeta, että terveydenhuoltoalalla työskentelevien mielestä käyttöönottoon liittyvissä asioissa ei tällä hetkellä olla riittävän hyvällä tasolla. Ohjelmistoyrityksissä työskentelevien mielestä ollaan vähän paremmalla tasolla. Tavoitetilassa molemmat ryhmät toivovat kehitystä.

Vastaajien mielestä järjestelmän toimittajia tulisi olla vain yksi, joka sitten toimittaa koko pake-tin. Henkilökunnan selviytymisen huomioiminen toimittamalla hyvät manuaalit ja järjestämällä koulutus koettiin tärkeänä asiana. Käyttöönotto tulisi suunnitella erikseen jokaista projektia varten, ja suunnitelman sisältöön pitää kiinnittää erityistä huomiota. Samaa suunnitelmaa ei tule käyttää eri järjestelmien käyttöönotossa.

Käyttöönoton ongelmina nähtiin muun muassa useamman toimittajan yhteistyö, epärealistinen aikataulu sekä koulutuksen riittämättömyys. Koulutuksen järjestäminen tulisi suunnitella siten, että mahdollisimman moni pystyy osallistumaan siihen. Tämä ei terveydenhuoltoalalla ole helppoa, koska myös koulutuksen aikana työt osastolla pitää hoitaa. Koulutuksen ajaksi sijaisten saaminen on vaikeaa. Usein käy niin, että toimittajan toimesta koulutetaan vain pieni osa henkilökuntaa ja henki-lökunnasta joku kouluttaa sitten loput. Järjestelmän käytön alkuvaiheessa tukea pitäisi saada nykyis-tä enemmän.

Vastaajien esittämiä parannusehdotuksia oli muun muassa käyttäjän mukaanotto jo suunnittelu-vaiheessa, lisää resursseja tietojärjestelmän käytön opetteluun, osastojen toiminnan suunnittelu en-nen käyttöönottoa ja klinikoiden johdon mukaanotto käyttöönottoon. Vastaajien mielestä järjestel-miä tulisi ennen käyttöönottoa testata pienessä ryhmässä siten, että käyttäjät ovat mukana testauk-sessa todellisissa olosuhteissa.

# 7 TESTAUS JA TARKASTUS

## 7.1 Yleistä testauksesta ja tarkastuksesta

*Ohjelmiston testauksella* tarkoitetaan virheiden suunnitelmallista etsimistä ohjelmaa tai sen osaa suorittamalla. Testauksen vaiheet ovat: testauksen suunnittelu, testiympäristön luonti, testin suorittaminen ja tulosten tarkastelu (ks. Virkanen, 2002). Testauksessa voidaan käyttää hyväksi erilaisia testaustyökaluja.

Lyhyt kuvaus testausmenetelmistä:

*Moduulitestaus eli yksikkötestaus* on yksittäisen ohjelmamoduulin testaamista.

*Mustalaatikossa* ei ohjelmakoodi ole nähtävissä. Testitapaukset suunnitellaan vaatimusmäärittelyjen pohjalta.

*Lasilaatikossa* ohjelmakoodi on nähtävissä ja testitapaukset laaditaan siten, että koodi tulisi mahdollisimman kattavasti läpikäydyksi.

*Harmaa laatikko* on kahden edellisen yhdistelmä, eli hyödynnetään sekä vaatimuksia, että ohjelmakoodia.

*Integroititestauksessa* testataan toisiinsa liittyvien ohjelmamoduulien toimivuus yhdessä.

*Kertarysäyksessä* linkitetään kaikki moduulit kerralla yhteen ja testataan yhtenä kokonaisuutena.

*Jäsentävässä menetelmässä* lähdetään liikkeelle ensimmäisestä (ylimmästä) moduulista testaten se. Testaus etenee kutsuttujen moduulien mukaisesti alaspäin.

*Kokoavassa menetelmässä* lähdetään alimman tason moduuleista ja edetään ylöspäin kutsuvien moduulien mukaan.

*Kerrosvoileipämenetelmässä* edetään yhtä aikaa ylhäältä alas ja alhaalta ylös. Se on siis jäsentävän ja kokoavan sekoitus.

*Toiminnallisuustestaus eli suorituskykytestaus* mittaa, kuinka hyvin systeemi suoriutuu vaatimusmäärittelyissä sille asetetuista tavoitteista.

*Järjestelmätestauksessa* testataan koko systeemi (kaikki moduulit) yhtenä kokonaisuutena.

*Hyväksymistestaus* on asiakkaan ja valmistajan yhdessä suorittama viimeinen tarkistus ennen järjestelmän käyttöönottoa.

*Järjestelmän toimintaprofiilin tutkimisen* perusteella keskitytään testaamaan niitä osa-alueita, joita todennäköisesti käytetään eniten.

*Ekvivalenssiluokitus* on toiminnallisen testauksen (black-box) menetelmä, jossa jokainen arvoalue jaetaan ekvivalenssiluokkiin (=kelvollisten syötteiden osajoukot sekä epäkelvojen syötteiden joukko). Näille luokille pätee, että jokainen tiettyyn ekvivalenssiluokkaan kuuluva arvo käyttäytyy tes-

tauksen kannalta 'samalla tavalla'. Jokaisesta luokasta otetaan yksi tai useampi edustaja testitapaueksi. Ekvivalenssiluokituksella vähennetään tarvittavien testitapausten määrää.

*Tarkastusmenetelmä* on kokoustekniikka, jota voidaan käyttää minkä tahansa kirjallisen tuotteen laadun varmistamiseen V-mallin määrittely- ja suunnitteluvaiheista aina ohjelma-koodiin saakka. Menetelmän vaiheet ovat suunnittelu, esittely, valmistautuminen, tarkastustilaisuus, korjaukset ja seuranta.

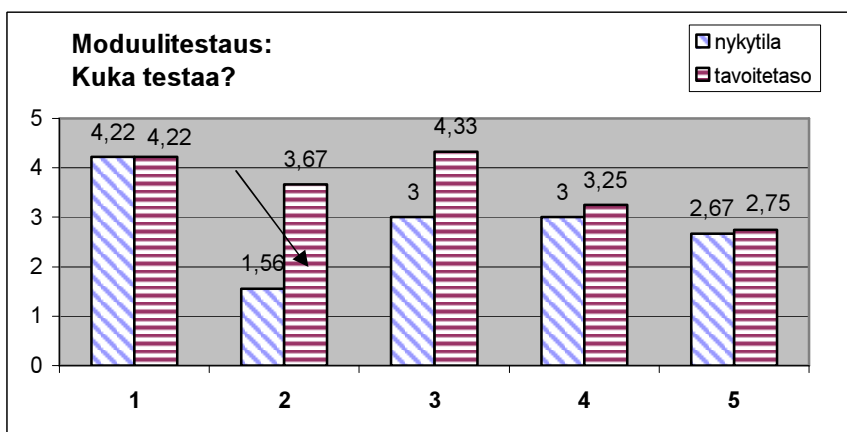
## 7.2 Kyselyn tulokset

Testaukseen ja tarkastukseen liittyvissä kysymyksissä vastausfrekvenssi vaihteli viiden ja yhdeksän välillä. Kaikki vastaajat eivät merkinneet tavoitetasoa. Vastauksista kuusi oli ohjelmistoyrityksistä. Sairaalan atk-osastolta ja erikoissairaanhoidosta vastauksia oli molemmista yksiköistä yksi. Yksi vastaajista ei ilmoittanut yksikkönsä tyyppiä. Vastaajia pyydettiin arvioimaan eri testausmenetelmien käyttötilannetta sekä menetelmien käyttökelpoisuutta. Seuraavana esitetään kyselyn tulokset osaluueittain.

Käyttötilanteen asteikko oli: 0 = Ei koskaan, 1 = Harvoin, 2 = Joskus, 3 = Usein, 4 = Lähes aina, 5 = Aina. Käyttökelpoisuuden arviointi-asteikko oli: 0 = kelvoton, 1 = menettelee, 2 = hyvä, 3 = erinomainen. Kuvissa olevat nuolet osoittavat niitä vaihtoehtoja, joissa vastaajien mielestä on eniten kehitettävää eli nyky- ja tavoitetilän välinen ero on huomattava.

### 7.2.1 Moduulitestaus

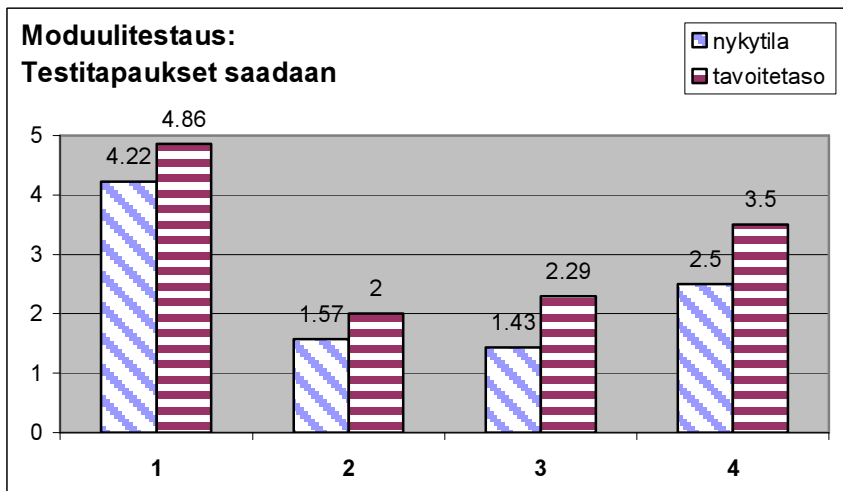
Moduulitestauksen suorittaa lähes aina ohjelmoija itse. Joskus ohjelmoijat testaavat vuorotellen toistensa moduulit. Tavoitetilassa moduulien *paritestaus* on huomattavasti nykyistä yleisempää. Yleisimmin moduulitestauksen suorittaa tavoitetilassa toimittajan testausryhmä. Kukaan vastaajista ei nimennyt ketään muuta, joka huolehtii moduulitestauksesta (Kuva 30). Käyttökelpoisimpana tapana moduulitestauksen suorittamiseksi pidetään toimittajan testausryhmän tekemää testausta. Käyttökelpoisuudeltaan huonoin on loppukäyttäjän suorittama moduulitestaus.



Kuva 30. Moduulitestauksen suorittajat

1. Ohjelmoija itse, käyttökelpoisuus 1,89
2. Ohjelmoijat testaavat vuorotellen toistensa moduulit, käyttökelpoisuus 1,64
3. Toimittajan testausryhmä, käyttökelpoisuus 2,13
4. Asiakkaan testausryhmä, käyttökelpoisuus 1,89
5. Loppukäyttäjä, käyttökelpoisuus 1,33

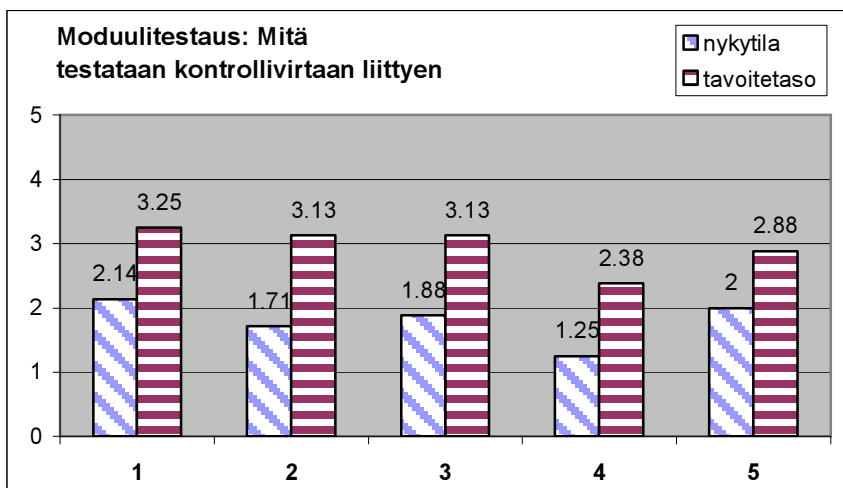
Moduulitestauksen testitapaukset saadaan lähes aina käyttötapauksista, tavoitetaso lähentelee aina-  
tasoa. Menettely koettiin myös käyttökelpoiseksi. Aktiviteetti- ja tilakaaviosta testitapauksia saa-  
daan harvoin. Moduulitestauksen testitapauksia saadaan vastaajien mukaan myös testaussuunnitel-  
masta ja prosessikaaviosta. (Kuva 31)



Kuva 31. Moduulitestauksen testitapausten saaminen

1. Käyttötapauksista, käyttökelpoisuus 2,5
2. Aktiviteettikaaviosta, käyttökelpoisuus, 0,88
3. Tilakaaviosta, käyttökelpoisuus 1
4. Muualta, käyttökelpoisuus 1

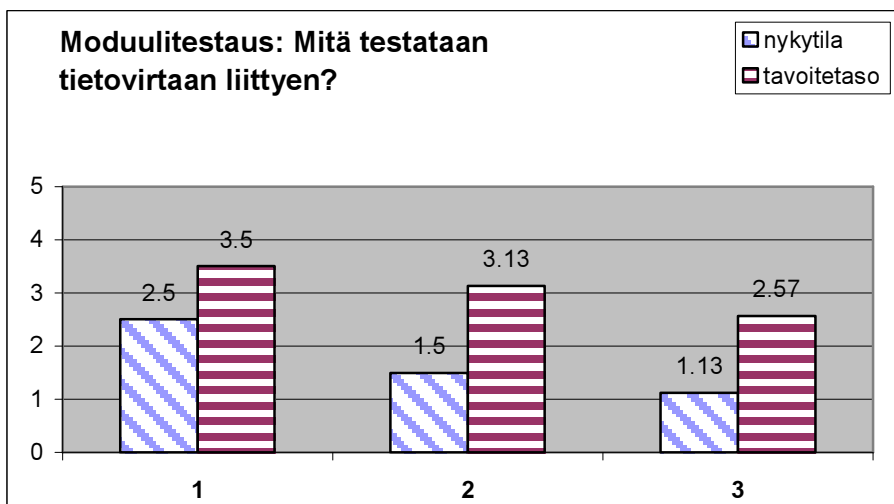
Kontrollivirtaan liittyviä eri kattavuuskriteerejä käytetään harvoin tai joskus. Tavoitetilassa niiden käyttöä lisätään (Kuva 32). Kattavuuskriteerien käyttökelpoisuus vaihteli 1,25:n ja 1,75:n välillä. Käyttökelpoisuudeltaan parhaat olivat päätöskattavuus ja lausekattavuus.



Kuva 32. Kontrollivirtaan liittyen testattavat asiat

1. Lausekattavuus= kaikkien lauseiden läpikäynti, käyttökelpoisuus 1,71
2. Päätökattavuus= kaikkien ohjelman vuokaavion kaarien läpikäynti, käyttökelpoisuus 1,75
3. Ehtokattavuus= kaikkien ehtolauseiden 'tosi' ja 'epätosi' arvojen läpikäynti, käyttökelpoisuus 1,38
4. Moniehtokattavuus= kaikkien ehtolauseiden 'tosi' ja 'epätosi' arvojen eri kombinaatioiden läpikäynti, käyttökelpoisuus, 1,25
5. Polkukattavuus= mahdollisimman monien ohjelman eri polkujen läpikäynti, käyttökelpoisuus 1,38

Tietovirtaan liittyvien asioiden testausmenetelmistä suosituin oli menetelmä, jossa kaikki määritellyt saavuttavat jonkin käyttönsä. Kahta muuta käytetään harvoin. Tavoitetilassa kaikkien menetelmien käytön toivotaan lisääntyvän (Kuva 33). Käyttökelpoisimpia ovat: "kaikki määritellyt saavuttavat jonkin käyttönsä" ja "kaikki määrittelyt ja käytöt saavuttavat toisensa kaikkia mahdollisia polkuja pitkin".



Kuva 33. Tietovirtaan liittyen testattavat asiat

1. Kaikki määrittelyt saavuttavat jonkin käyttönsä, käyttökelpoisuus 1,5
2. Kaikki määrittelyt saavuttavat kaikki käyttönsä, käyttökelpoisuus 1,38
3. Kaikki määrittelyt ja käytöt saavuttavat toisensa kaikkia mahdollisia polkuja pitkin, käyttökelpoisuus 1,5

Moduulitestauksen ongelmista vastaajat (n=6) mainitsivat:

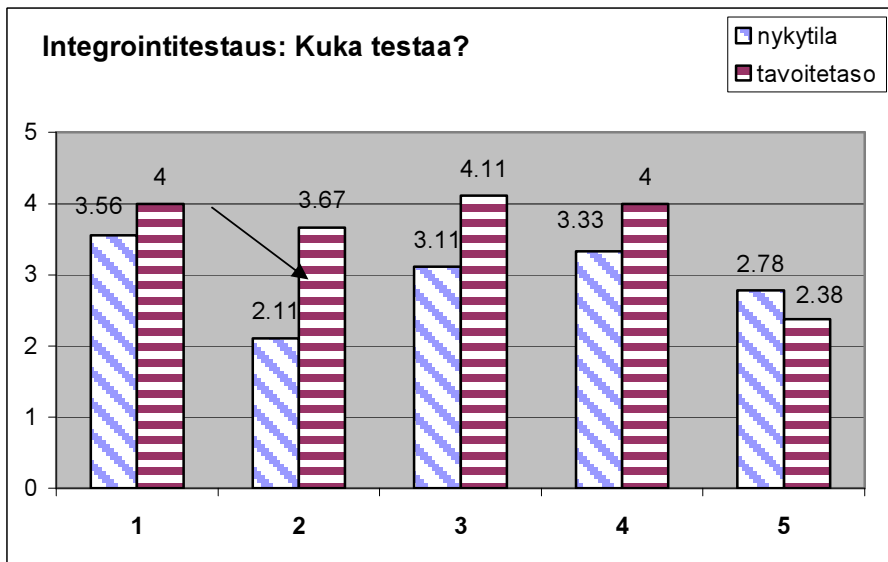
- erilaisten järjestelmien ja tekniikoiden yhteensopimattomuus
- aikapula
- työkalut ovat vaikeita käyttää
- on vaikea tietää milloin testaus riittävää
- puutteelliset määrittelyt

Vastaajien (n=3) mielestä moduulitestauksessa tulisi kehittää:

- testipenkkejä
- työkaluja
- työntekijöille koulutusta eri testausmenetelmistä
- standardointia
- aikataulua

## 7.2.2 Integroititestausta

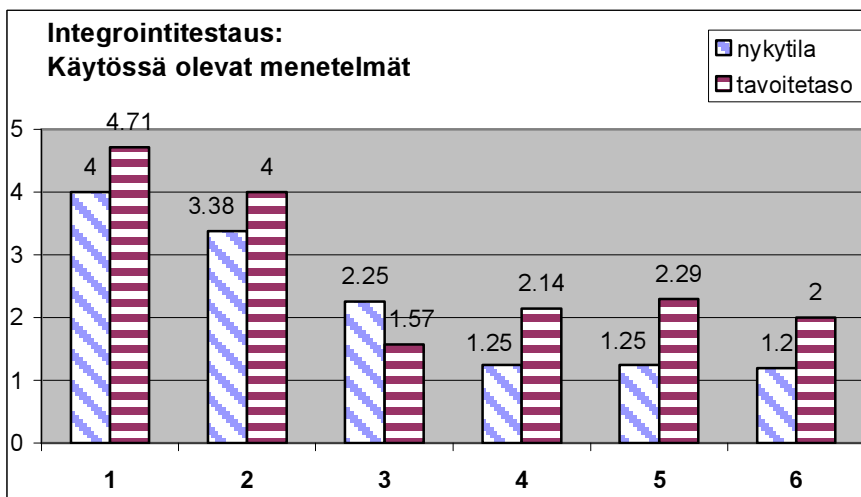
Tällä hetkellä integroititestausta suorittaa useimmiten ohjelmoija itse. Sekä asiakkaan testausryhmä että toimittajan testausryhmä osallistuvat usein testaamiseen. Tavoitetilassa kaikkien muiden paitsi loppukäyttäjän osuus integroititestausta lisääntyy. Tavoitetilassa paritestausta on nykyistä yleisempää (Kuva 34). Käyttökelpoisimpana vaihtoehtona integroititestausta suorittajaksi nähtiin toimittajan testausryhmä.



Kuva 34. Integroititestausta suorittajat

1. Ohjelmoija itse, käyttökelpoisuus 2
2. Tekijät testaavat vuorotellen toistensa työt, käyttökelpoisuus 2
3. Toimittajan testausryhmä, käyttökelpoisuus 2,5
4. Asiakkaan testausryhmä, käyttökelpoisuus 2,25
5. Loppukäyttäjä, käyttökelpoisuus 1,38

Integroititestauksessa käytetyistä menetelmistä mustalaatikkotestaus on suosituin. Myös tavoitetilassa se säilyy käytetyimpänä menetelmänä. Seuraavaksi suosituin on virheenjäljitys. Kertarysäystä lukuun ottamatta kaikkien menetelmien käyttö tavoitetilassa lisääntyy. Suhteellisesti eniten suositaan lisää kokoava integrointi (Kuva 35). Käyttökelpoisin menetelmä on mustalaatikkotestaus. Myös virheenjäljityksen käyttökelpoisuus on "hyvä". Muiden menetelmien käyttö "menettelee". Kukaan vastaajista ei ilmoittanut, että heillä käytetään integroititestauksessa jotain muuta menetelmää.

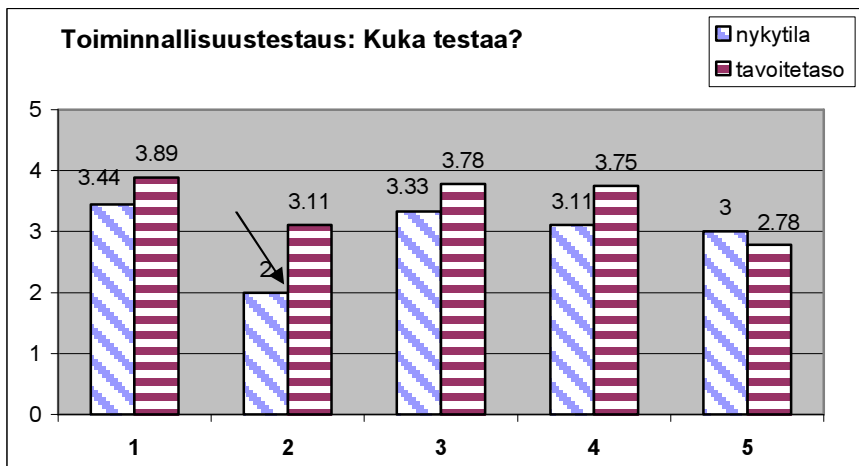


Kuva 35. Integroititestauksessa käytössä olevat menetelmät

1. Mustalaatikko, käyttökelpoisuus 2,43
2. Virheenjäljitys (debuggaus), käyttökelpoisuus 2,13
3. Kertarysäys (big-bang), käyttökelpoisuus 0,17
4. Jäsentävä (top-down), käyttökelpoisuus 1,29
5. Kokoava (bottom-up), käyttökelpoisuus 1,43
6. Kerrosvoileipä (sandwich), käyttökelpoisuus 1

### 7.2.3 Toiminnallisuustestaus

Toiminnallisuustestauksen suorittaa yleensä ohjelmoija itse, mutta samalla tasolla ovat myös toimittajan testausryhmä, asiakkaan testausryhmä sekä loppukäyttäjä. Loppukäyttäjän osallistuminen toiminnallisuustestaukseen pysyy melko ennallaan, kun taas muiden lisääntyy. Suhteellisesti eniten lisääntyy paritestaus (Kuva 36). Käyttökelpoisimpana nähdään asiakkaan testausryhmän suorittama toiminnallisuustestaus.

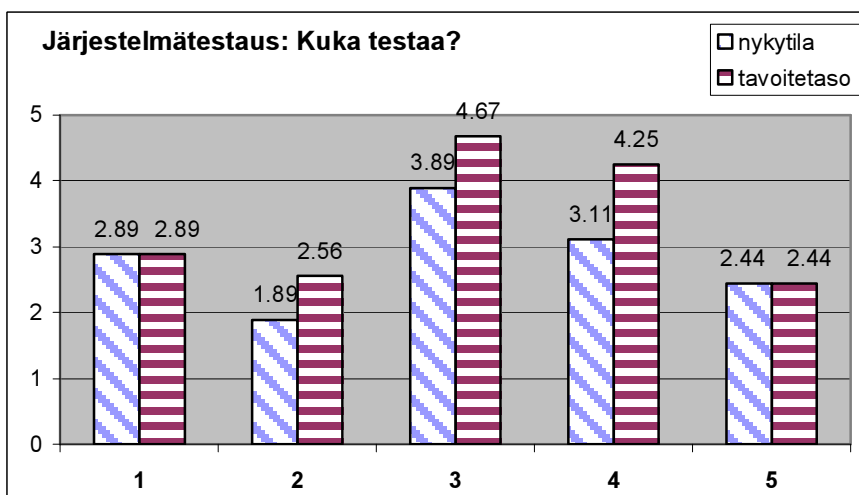


Kuva 36. Toiminnallisuustestauksen suorittajat

1. Ohjelmoija itse, käyttökelpoisuus 1,89
2. Tekijät testaavat vuorotellen toistensa työt, käyttökelpoisuus 1,89
3. Toimittajan testausryhmä, käyttökelpoisuus 2,11
4. Asiakkaan testausryhmä, käyttökelpoisuus 2,22
5. Loppukäyttäjä, käyttökelpoisuus 1,78

## 7.2.4 Järjestelmätestaus

Järjestelmätestausta suorittaa tällä hetkellä ja myös tavoitetilassa, useimmiten toimittajan testausryhmä. Suhteellisesti eniten lisääntyy asiakkaan testausryhmän suorittama järjestelmätestaus. Ohjelmoijan suorittama järjestelmätestaus säilyy ennallaan (Kuva 37). Käyttökelpoisimpia ovat toimittajan ja asiakkaan testausryhmän tekemä järjestelmätestaus.

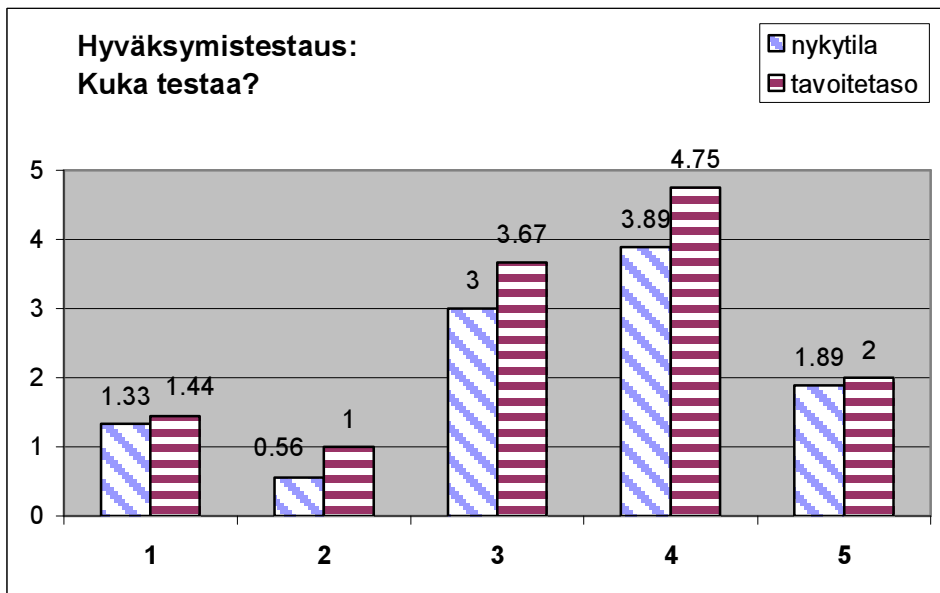


Kuva 37. Järjestelmätestauksen suorittajat

1. Ohjelmoija itse, käyttökelpoisuus 1,25
2. Tekijät testaavat vuorotellen toistensa työt, käyttökelpoisuus 1,38
3. Toimittajan testausryhmä, käyttökelpoisuus 2,5
4. Asiakkaan testausryhmä, käyttökelpoisuus 2,5
5. Loppukäyttäjä, käyttökelpoisuus 1,25

### 7.2.5 Hyväksymistestaus

Hyväksymistestauksen suorittaa lähes aina asiakkaan testausryhmä. Myös tavoitetilassa se pysyy selvästi käytetyimpänä vaihtoehtona. Toimittajan testausryhmä hyväksymistestauksen tekijänä on myös aika yleinen vaihtoehto. Ohjelmoijan osuus on vähäinen, samoin loppukäyttäjän (Kuva 38). Käyttökelpoisimpia ovat asiakkaan testausryhmän ja toimittajan testausryhmän suorittama hyväksymistestaus.

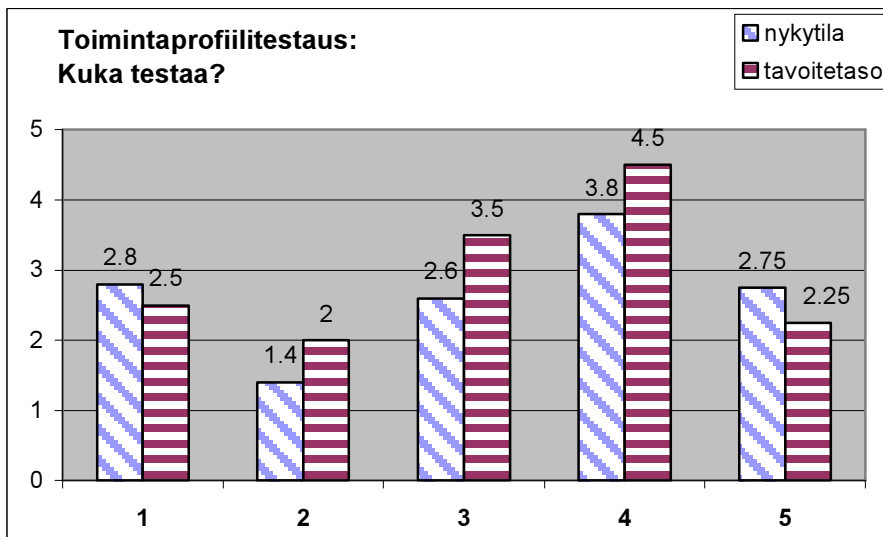


Kuva 38. Hyväksymistestauksen suorittajat

1. Ohjelmoija itse, käyttökelpoisuus 0,57
2. Tekijät testaavat vuorotellen toistensa työt, käyttökelpoisuus 0,57
3. Toimittajan testausryhmä, käyttökelpoisuus 2,25
4. Asiakkaan testausryhmä, käyttökelpoisuus 2,63
5. Loppukäyttäjä, käyttökelpoisuus 1,29

### 7.2.6 Toimintaprofiilitestaus

Toimintaprofiilitestauksen tekee useimmiten asiakkaan testausryhmä, myös tavoitetilassa se säilyttää ykkössijan. Tavoitetilassa yleistyy selvästi toimittajan testausryhmän suorittama toimintaprofiilitestaus. Ohjelmoijan ja loppukäyttäjän osuus on hieman vähäisempi kuin tällä hetkellä (Kuva 39). Käyttökelpoisimpia ovat asiakkaan testausryhmän ja toimittajan testausryhmän suorittama toimintaprofiilitestaus.

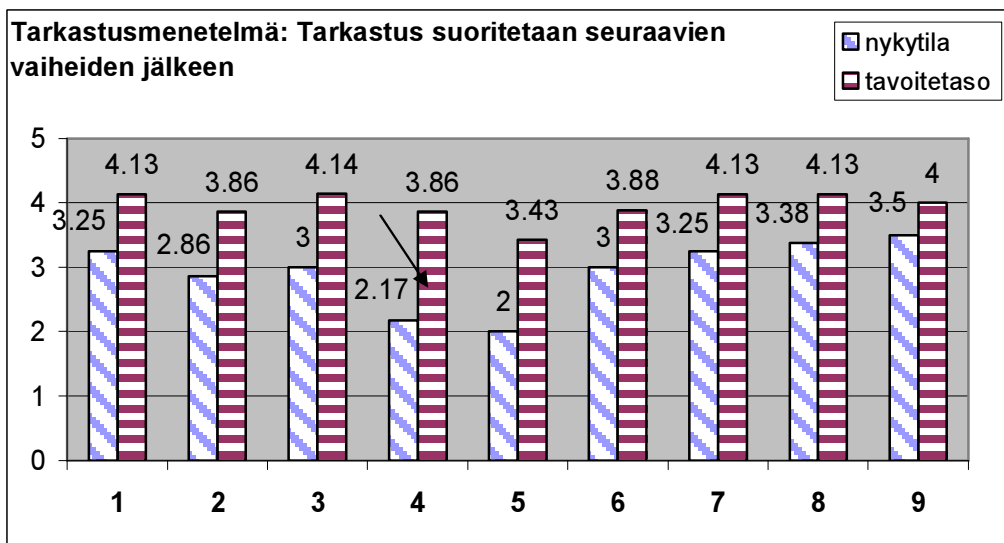


Kuva 39. Toimintaprofiilitestauksen suorittajat

1. Ohjelmoija itse, käyttökelpoisuus 1
2. Työpari vuorotellen toistensa työt, käyttökelpoisuus 1
3. Toimittajan testausryhmä, käyttökelpoisuus 1,5
4. Asiakkaan testausryhmä, käyttökelpoisuus 2
5. Loppukäyttäjä, käyttökelpoisuus 1,4

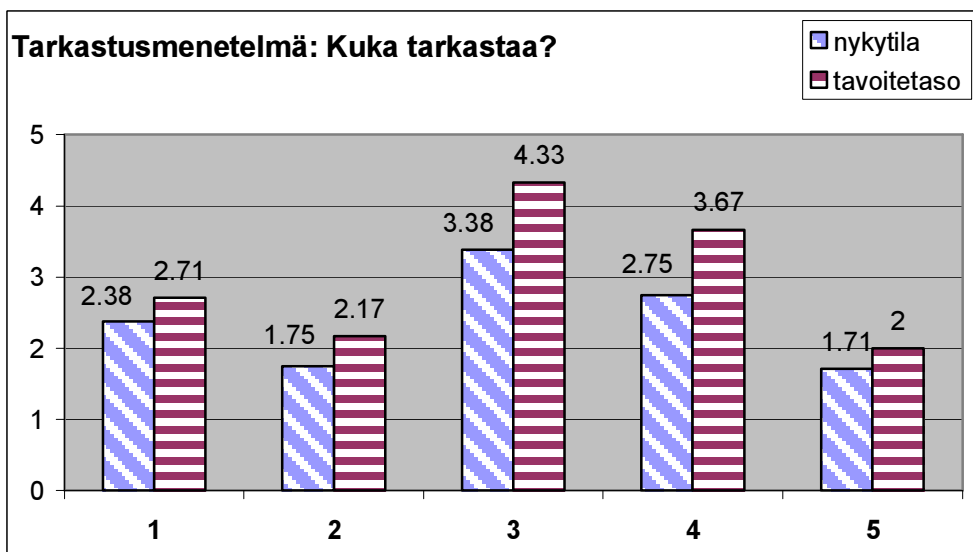
## 7.2.7 Tarkastusmenetelmä

Eniten tarkastusta suoritetaan hyväksymistestauksen jälkeen. Tarkastus on melko yleistä jokaisen vaiheen jälkeen, mutta kuvasta 40 nähdään selvästi, että tavoitetilassa tarkastusta lisätään huomattavasti kaikkien vaiheiden kohdalla. Suhteellisesti eniten pitää lisätä koodauksen jälkeistä tarkastusta. Tarkastusmenetelmän käyttökelpoisuudessa ei ole paljon vaihteluita eri vaiheiden välillä. Tarkastuksen suorittaa yleensä tarkastusryhmä (Kuva 41). Käyttökelpoisimpana koetaan tarkastusryhmän ja asiakkaan suorittama tarkastus.



Kuva 40. Vaiheet, joiden jälkeen tarkastusta suoritetaan

- |  |   |
|--|---|
| 1. Vaatimusten määrittely, käyttökelpoisuus 2,38   | 6. Integrointitestaus, käyttökelpoisuus 2,25        |
| 2. Arkkitehtuurisuunnittelu, käyttökelpoisuus 2,14 | 7. Toiminnallisuuden testaus, käyttökelpoisuus 2,25 |
| 3. Ohjelmistosuunnittelu, käyttökelpoisuus 2,29    | 8. Järjestelmätestaus, käyttökelpoisuus 2,13        |
| 4. Koodaus, käyttökelpoisuus 2,14                  | 9. Hyväksymistestaus, käyttökelpoisuus 2,13         |
| 5. Yksikkötestaus, käyttökelpoisuus 2              |   |



Kuva 41. Tarkastuksen suorittajat

1. Suunnittelija itse, käyttökelpoisuus 1,43
2. Työpari vuorotellen toistensa työt, käyttökelpoisuus 1,71
3. Tarkastusryhmä, käyttökelpoisuus 2,57
4. Asiakas, käyttökelpoisuus 2,57
5. Loppukäyttäjä, käyttökelpoisuus 1,29

## 7.2.8 Automatisoidun testauksen työkalut

Vastaajia pyydettiin arvioimaan heillä käytössä olevia automatisoidun testauksen työkaluja (ks. Pohjolainen, 2003). Kyselyyn vastaajista neljä oli vastannut tähän kysymykseen. Kahdella vastaajista oli käytössä kaksi eri työkalua ja kahdella yksi. Kaiken kaikkiaan käytössä oli kuusi eri työkalua. Käytössä olevat työkalut olivat:

- WebStress
- Rational TeamTest
- Compuware QARun
- TestDirector
- WinRunner
- Jtest

Työkalujen keskimääräinen käyttöaika oli kaksi vuotta. Oppimisaika viikkoina oli noin kolme. Käyttökelpoisuutta, tulosten tulkintaa ja sijoituksen kannattavuutta pidettiin hyvinä. Työkaluja käytetään jonkin verran tai usein. Ohjelmistojen toimivuuden katsottiin työkalujen käytön johdosta parantuneen jonkin verran tai usein.

## 7.3 Pohdintaa

Kyselystä kävi ilmi, että testauksessa yleisesti ottaen ollaan enenevässä määrin siirtymässä kohti testausryhmän suorittamaa testausta. Ohjelmoijan osuus pysyy aika samana. Moduulitestauksessa ja integrointitestauksessa ohjelmoijien suorittama työparin ohjelmien ja määrittelyjen testaaminen lisääntyy huomattavasti.

Kertarysäyksen käyttäminen integrointitestauksessa vähenee ja muiden menetelmien käyttö lisääntyy. Tämä on hyvä asia, koska kertarysäys soveltuukin ainoastaan pienten ohjelmistojen testaukseen.

Tarkastusta suoritetaan usein lähes jokaisen ohjelmistotuotannon vaiheen jälkeen. Tavoitetilassa tarkastukset vielä lisääntyvät. Tarkastukset ovatkin testausta tehokkaampi virheiden etsintäkeino. Mitä aikaisemmin virheet havaitaan, sitä halvempaa niiden korjaaminen on.

Ainoastaan neljä oli vastannut työkaluihin liittyviin kysymyksiin. Vastausten perusteella automatisoidun testauksen työkaluihin ollaan tyytyväisiä.

Testauksen ongelmista vastaajat mainitsivat muun muassa resurssien riittämättömyyden, testausympäristön puutteen ja apuvälineiden vähyden. Testaus ei myöskään tunnu luonnolliselta joka vaiheessa. Monet virheistä saattavat, testauksesta huolimatta, tulla esille vasta, kun on paljon käyttäjiä ja tapahtumia. Testausta ei myöskään mielletä tärkeäksi osaksi ohjelmistotuotantoa.

Testauksen kehittämiskohteina vastaajat mainitsivat määritysten testaamisen lisäämisen sekä suunnitelmallisen yksikkö-, integraatio- ja suorituskykytestauksen.

Tarkastusten ongelmia vastaajien mielestä ovat: ajankäytön suunnittelu, toimintamallin puuttuminen, väärä ajoitus ja suunnitelmadokumenttien vähyys. Tarkastuksia voitaisiin heidän mielestä kehittää lisäämällä tarkastuksia koodaus- ja yksikkötestausvaiheeseen ja tekemällä tarkastuksesta pakollinen.

# LÄHTEET

- Fowler M. *UML distilled: A brief guide to the standard object modeling language*. Addison-Wesley, 1999.
- Haikala I, Märijärvi J. *Ohjelmistotuotanto*. Rt-Print Oy, 2000.
- Hartikainen R. *Vaatimusten mallintaminen*. Pro gradu -tutkielma, Kuopion yliopisto, 2003.
- Herzum P, Sims O. *Business component factory: A comprehensive overview of component-based development for the enterprise*. John Wiley & Sons, Inc., 1999.
- Immonen M. *Suunnittelumallit*. Erikoistyö, Kuopion yliopisto, 2002, <http://www.plugit.fi/julkaisut>.
- Jäntti M. *Testitapausten suunnittelu UML-mallinnuksen avulla*. Pro gradu -tutkielma, Kuopion yliopisto, 2003, <http://www.plugit.fi/julkaisut>.
- Kärkkäinen T-L. *PlugIT-yrityskyselyn analysointi*. Kuopion yliopisto 2003.
- Kruchten P.B. The 4+1 view model architecture, *IEEE Software*, 1995.
- Mykkänen J. *Komponentti-FixIT: Terveystuotannon komponenttipohjainen sovellustuotanto - toiminnallisuus, arkkitehtuuri, siirtymästrategiat ja välineet*. ATK-keskus, Kuopion yliopisto, 2000.
- Myöhänen H. *Jäljitettävyys ohjelmistotuotannon tukena*. Pro gradu -tutkielma, Kuopion yliopisto, 2002, <http://www.plugit.fi/julkaisut>.
- Pohjolainen P. *Ohjelmiston testauksen automatisointi*. Pro gradu -tutkielma, Kuopion yliopisto, 2003, <http://www.plugit.fi/julkaisut>.
- Riekkinen A. *Ohjelmistotuotannon nykytilatutkimus*. PlugIT-seminaari, Kuopio, 2003.
- Ripatti S. Tietojärjestelmän käyttöönotto, kirjassa Saranto K., Korpela M. *Tietotekniikka ja tiedonhallinta sosiaali- ja terveydenhuollossa*. WSOY, 1999.
- Robertson S, Robertson J. *Mastering the requirements process*. Addison-Wesley, 1999.
- Virkanen H. *Ohjelmistojen testaus ja virheenjäljitys*. Pro gradu -tutkielma, Kuopion yliopisto, 2002, <http://www.plugit.fi/julkaisut>.