



TAMPEREEN TEKNILLINEN YLIOPISTO

PETRI HONKANEN
SÄHKÖISEN POTILASTIETOJÄRJESTELMÄN
HANKINTAPROSESSIN ARVIOINTI

Diplomityö

Tarkastaja: professori Kari Mäkelä
Tarkastaja ja aihe hyväksytty
Tieto- ja sähkötekniikan tiedekuntaneu-
voston kokouksessa 3. maaliskuuta
2010

Tiivistelmä

TAMPEREEN TEKNILLINEN YLIOPISTO

Sähkötekniikan koulutusohjelma

HONKANEN, PETRI: Sähköisen potilastietojärjestelmän hankintaprosessin arviointi

Diplomityö, 68 sivua

Maaliskuu 2010

Pääaine: Lääketieteellinen tekniikka

Tarkastaja: professori Kari Mäkelä

Avainsanat: Terveystieteiden tietojärjestelmä, sähköinen potilastietojärjestelmä, CMMI, projektihallinta, elinkaarimalli, ohjelmistohankinta

Sähköiset potilastietojärjestelmät ovat yleistyneet Suomen perus- ja erikoisterveydenhuollossa viimeisen kymmenen vuoden aikana. Tietojärjestelmien käyttöönottojen myötä on julkisuuteen tullut myös paljon niihin liittyviä ongelmia. Tässä työssä arvioidaan Etelä-Pohjanmaan sairaanhoitopiirin SÄHKE-projektissa käytettyjä projektinhallintakäytäntöjä. Työssä pyritään löytämään projektihallintamenetelmien kehityskohteita, joilla pystyttäisiin välttämään vastaavanlaisissa projekteissa esiintyviä ongelmia ja kehittämään projektin menetelmiä parantamaan projektin seurattavuutta ja toimintatehoa.

Työ jakautuu kolmeen osaan. Taustaselvitysosuudessa perehdytään ohjelmistoprojektien elinkaarimalleihin ja terveydenhuollon tietojärjestelmien kehitystä Suomessa esitellään lyhyesti. Seuraavassa osassa käydään läpi SÄHKE-projektin tapahtumat vuosina 2000-2009. Arviointiosuudessa verrataan SÄHKE-projektissa käytettyjä ohjelmistohankintakäytäntöjä CMMI-ACQ -ohjelmistohankintaprosessin arviointimallin prosessialueisiin.

Arvioinnin tuloksena SÄHKE-projektin ohjelmistohankintamenetelmistä löytyi paljon kehitettävää. Tärkeimmäksi kehitysalueeksi tunnistettiin vaatimustenhallinta, koska se on perusta kaikille muille prosessialueille. SÄHKE-projektin tyyppisille projekteille suositeltiin vaatimushallinnan toimenpiteiden lisäksi elinkaarimallin määrittämistä, hankintasopimuksen ylläpitoa ja järjestelmällisemmän sekä tarkemman projektisuunnittelun ylläpitämistä. Tulokset ovat hyödynnettävissä vielä vuosia jatkuvissa sähköisten potilastietojärjestelmien käyttöönotoissa, järjestelmävaihtoissa ja muutoshankkeissa.

Abstract

TAMPERE UNIVERSITY OF TECHNOLOGY

Master's Degree Programme in Electrical Engineering

HONKANEN, PETRI: Evaluation of the Electronic Patient Record system acquisition process

Master of Science Thesis, 68 pages

March 2010

Major: Biomedical engineering

Examiner: Professor Kari Mäkelä

Keywords: Information systems in health care, electronic patient record system, CMMI, project management, life cycle model, software acquisition

The use of electronic patient record systems (EPR) has been increased during the past ten years in Finnish health care sector. Many problems related to the information system have been reported at the same time. This thesis evaluates project management processes of the SÄHKE-project in Etelä-Pohjanmaa hospital district. The goal is to find development areas of the project management processes to avoid similar problems in corresponding projects and to develop methods to improve project tracking and efficiency.

The thesis is divided into three parts. In the background study part, life cycle models of the software projects are explored. Development of the information systems in Finnish health care is introduced briefly. The history of SÄHKE-project from the years 2000-2009 is presented in the next part. In the evaluation part, the software acquisition processes of SÄHKE-project are compared against the process areas of CMMI-ACQ-model.

The evaluation indicates that many process areas of SÄHKE-project software acquisition could be improved. The most important section to develop was requirement management which is the basis for the rest of the process areas. The other recommended actions for the projects like SÄHKE were definition of the life cycle model, maintenance of supplier agreement and maintenance of precise project plan. The result of this evaluation can be utilized in on-going EPR implementations, system changes and development activities.

Alkusanat

Tämä työ on tehty Tampereen teknillisessä yliopistossa Biolääketieteen tekniikan laitoksella. Vuosien työkokemuksen jälkeen minulle oli tärkeää, että työssä voitaisiin hyödyntää itselleni kertynyttä käytännön kokemusta ohjelmistoprojektien hallinnasta ja toisaalta saada mahdollisuus kattavaan kirjallisuuskatsaukseen alueesta. Kiitos Biolääketieteen tekniikan laitokselle ja professori Kari Mäkelälle diplomityöaiheesta, joka mahdollisti nämä tavoitteet. Kiitos myös Etelä-Pohjanmaan sairaanhoitopiirille, joka antoi mahdollisuuden potilaskertomusjärjestelmäprojektinsa arvioinnille.

Suuret kiitokset esittäisin työn tarkastajalle, professori Kari Mäkelälle, työn suorittamiseen ja sisältöön saamistani ohjeista ja neuvoista. Lisäksi haluan kiittää EPSHP:n hankepäällikkö Markku Stenmania perehdytyksestä SÄHKE-projektiin ja työssä esitellyn projektitiedon oikeellisuuden tarkistamisesta.

Haluan kiittää myös vaimoani, Pirjoa, arvokkaista käyttäjänäkökulmista terveydenhuollon tietojärjestelmiin ja niiden kehitystyöhön liittyviin ongelmiin.

Ylöjärvellä 13.4.2010

Petri Honkanen

Sisällys

1	Johdanto.....	1
1.1	Työn tausta.....	1
1.2	Työn tavoitteet	1
2	Ohjelmistokehityksen elinkaarimallit.....	3
2.1	Ennusteelliset elinkaarimallit	3
2.1.1	Vesiputousmalli	4
2.1.2	Prototyypimalli.....	5
2.1.3	RAD	6
2.1.4	Inkrementaalinen malli.....	7
2.1.5	Spiraalimalli	8
2.2	Adaptiiviset ja ketterät elinkaarimallit	9
2.3	Ohjelmistoprojektin toimintaprosessin valinta	11
2.4	Ohjelmistohankintaprojektin hallinta.....	12
2.4.1	Elinkaarimallien muokkaaminen ohjelmistohankintaan	12
2.4.2	IEEE std 1062	14
3	Terveydenhuollon tietojärjestelmät.....	16
3.1	Sähköisten potilastietojärjestelmien kehitys	16
3.2	Kansallinen terveystietojärjestelmä	17
3.3	Tietojärjestelmähankkeiden onnistuminen.....	18
4	Etelä-Pohjanmaan sairaanhoitopiirin sähköisen potilaskertomusjärjestelmäprojektin yhteenveto	20
4.1	Tarvekartoitus ja selvitys järjestelmätoimittajista.....	20
4.2	Tarjouspyyntö	21
4.2.1	Sairaskertomusjärjestelmän vaatimukset	21
4.2.2	Potilashallinnon vaatimukset	22
4.2.3	Digitaalisen arkiston ja kuva-arkiston vaatimukset	24
4.3	Toimittajavalinta	24
4.4	Projektin ohjaus ja organisaatio	24
4.4.1	Kehitystyöryhmät.....	25
4.4.2	Käynnistämistyöryhmät	25
4.5	Järjestelmän osa-alueiden kehitystyö ja käyttöönotto.....	26
4.5.1	Tekstinkäsittely	27
4.5.2	Palautejärjestelmä	28
4.5.3	Digitaalinen sanelu ja purku	28
4.5.4	Ateriatilaus	28
4.5.5	Potilashallinto.....	29
4.5.6	Sairauskertomus	31

4.5.7	Hoitokertomus.....	34
4.5.8	Hoitotaulukko/kuumekurva.....	35
4.5.9	Leikkaushoitojärjestelmä	36
4.5.10	SÄHKE-projektin päättäminen	37
4.6	Proxit-hanke	37
4.7	Koulutus	39
5	Arviointimenetelmät.....	40
5.1	Projektihallintaprosessien arviointimenetelmät	40
5.1.1	CMM	41
5.1.2	ISO 15504	44
5.1.3	CMMI-ACQ.....	45
5.2	Arviointimenetelmän valinta.....	46
6	Tulokset ja niiden tarkastelu.....	48
6.1	SÄHKE-projektin vertailu viitemalliin	48
6.1.1	Tarjouspyyntömenettely ja hankintasopimuksen tekeminen	49
6.1.2	Sopimushallinta.....	50
6.1.3	Vaatimusten kehitystyö.....	51
6.1.4	Vaatimushallinta	52
6.1.5	Projektisuunnittelu	53
6.1.6	Projektin seuranta ja ohjaus	55
6.1.7	Prosessin ja tuotelaadun varmistus	57
6.1.8	Konfiguraatioiden hallinta	57
6.1.9	Mittaus ja analyysi	57
6.2	Toimittajan näkemys SÄHKE-projektista	57
7	Johtopäätökset	59
7.1	Elinkaarimallin määrittäminen.....	61
7.2	Vaatimushallinta	61
7.3	Hankintasopimuksen ylläpito.....	64
7.4	Projektisuunnittelu	64
	Lähteet.....	65

Lyhenteet ja merkinnät

CMM	Capability Maturity Model
CMMI	Capability Maturity Model Integration
CMMI-ACQ	CMMI for Acquisition
EPSHP	Etelä-Pohjanmaan Sairaanhoidopiiri
EPSHP	Etelä-Pohjanmaan sairaanhoidopiiri
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
ISO15504	Kansainvälinen standardi ohjelmistotuotantoprosessin arvioimiseen
KanTa	Kansallinen terveystietokanta
LESU	Leikkausohjelman suunnittelujärjestelmä
RIS	Radiology Information System
SAILA	Sairaskertomusasiakirjojen lainausten hallintajärjestelmä
SAPO	Sähköinen potilashallinnon järjestelmä
SEI	SW Engineering Institute
STM	Sosiaali- ja terveysministeriö
SÄHKE	Etelä-Pohjanmaan Sairaanhoidopiirin sähköinen sairaskertomusprojekti

1 Johdanto

Sähköisten potilastietojärjestelmien laajentunut käyttöönotto 2000-luvulla on merkinnyt voimakasta muutosta Suomen terveydenhuollon toimintaympäristössä. Vuonna 2001 erikoissairaanhoidon potilaskertomusjärjestelmät olivat pääosin vielä paperilla, ja perusterveydenhuollossa 63 % terveyskeskuksista käytti sähköistä potilaskertomusjärjestelmää. [13]. Jo vuoteen 2006 mennessä sähköisiä potilaskertomusjärjestelmiä käyttivät lähes kaikki Suomen sairaanhoitopiirit ja terveyskeskukset [15]. Potilaskertomusjärjestelmät ovat kiinteä osa terveydenhuollon organisaatioiden ydintoiminnan prosesseja. Muutokset näissä järjestelmissä vaikuttavat laajasti koko organisaation toimintatapoihin.

1.1 Työn tausta

Yhteiskunnan terveydenhuollon tietotekniikkaan liittyvän lainsäädännön ja strategiamäärittelyn toteuttamiseksi aloitti sosiaali- ja terveysministeriö kansallisen tietojärjestelmäarkkitehtuurin määrittelyn vuonna 2006. Kansallinen terveysarkisto (KanTa) on yhteinen nimitys terveydenhuollon valtakunnallisille tietojärjestelmäpalveluille. Alkuperäisen lainsäädännön mukaan kaikki potilastiedot pitäisi tallentaa viimeistään 1.4.2011 sähköisesti kansalliseen arkistoon. Käyttöönotto-ongelmien vuoksi tätä aikataulua ollaan kuitenkin siirtämässä. Kaikkiin palveluun liittyviin potilastietojärjestelmiin täytyy tehdä muutoksia yhteensopivuuden saavuttamiseksi ja KanTa-palveluun liittyminen onkin lähivuosien suurin terveydenhuollon tietojärjestelmiä koskeva muutos.

Terveydenhuollon tietojärjestelmien kehitystyötä on arvosteltu Suomessa voimakkaasti. Hankkeita on kritisoitu kokonaisuuden hallinnan puutteesta ja moitittu kehittäville yrityksille annettujen vaatimusten vajavaisuudesta. Kommenttien mukaan ohjelmat on otettu usein käyttöön keskeneräisinä, käyttäjät eivät ole olleet niitä kehittämässä ja sähköisen potilaskertomuksen moitittiin vievän runsaasti työaika. [23; 24; 25].

1.2 Työn tavoitteet

Tässä työssä arvioitiin Etelä-Pohjanmaan sairaanhoitopiirin (EPSHP) SÄHKE-projektin ohjelmistohankinnan käytäntöjä. SÄHKE-projekti vastasi vuosien 2000-2009 aikana EPSHP:ssä tapahtuneesta potilastietojärjestelmän käyttöönotosta. Projektin käytäntöjä verrattiin CMMI-ACQ -mallin [45] kypsyytason 2 prosessialueisiin. Arvioinnissa tunnistettiin projektihallinnan alueet, joihin olisi kannattanut kiinnittää enemmän huomiota SÄHKE-projektissa. Tulokset ovat hyödynnettävissä muissa vastaavassa tilanteessa

olevissa ohjelmistohankintaprojekteissa, ja toivottavasti ehdotetut toimenpiteet auttavat myös edessä olevien terveydenhuollon tietojärjestelmämuutosten läpivientiä.

Luvussa 2 esitellään tunnetuimpia ohjelmistokehityksen elinkaarimalleja, joiden perusteella ohjelmistoprojektien hallintaprosesseja voidaan määritellä. Suomen terveydenhuollon tietojärjestelmien kehitystä käsitellään lyhyesti luvussa 3. Arvioitavan SÄHKE-projektin kymmenvuotisen historian päävaiheet ja tapahtumat käydään läpi luvussa 4. Luvussa 5 esitellään yleisimmät ohjelmistoprojektien hallintaprosessien arviointimenetelmät ja valitaan SÄHKE-projektin arviointiin parhaiten soveltuva menetelmä. Luvussa 6 käydään läpi tehdyn arvioinnin tulokset ja tunnistetaan kehityskohteita. Arvioinnin johtopäätökset ja toimenpidesuosituksukset vastaavanlaisille projekteille esitellään luvussa 7.

2 Ohjelmistokehityksen elinkaarimallit

Ohjelmiston kehitys koostuu erilaisista vaiheista, joiden yhdistelmä on ohjelmiston elinkaari. Projekteille on neljä yleistä elinkaarivaihetta ja päätöksentekopistettä: Hahmotus (*Concept*), Määrittely (*Definition*), Suoritus (*Execution*), Lopetus (*Closeout*). Nämä vaiheet ovat kuitenkin liian laajoja ja tulkinnanvaraisia, että niiden avulla pystyttäisiin yksiselitteisesti kuvaamaan projektin elinkaariprosessi. Tarvitaan tarkempia kuvauksia ja uusia alivaiheita, jotta elinkaariprosessi saadaan ymmärrettävästi kuvattua. Prosessin lopputulokseen vaikuttaa suoraan myös millaista elinkaarimallia sen määrittelyyn on käytetty.[1]

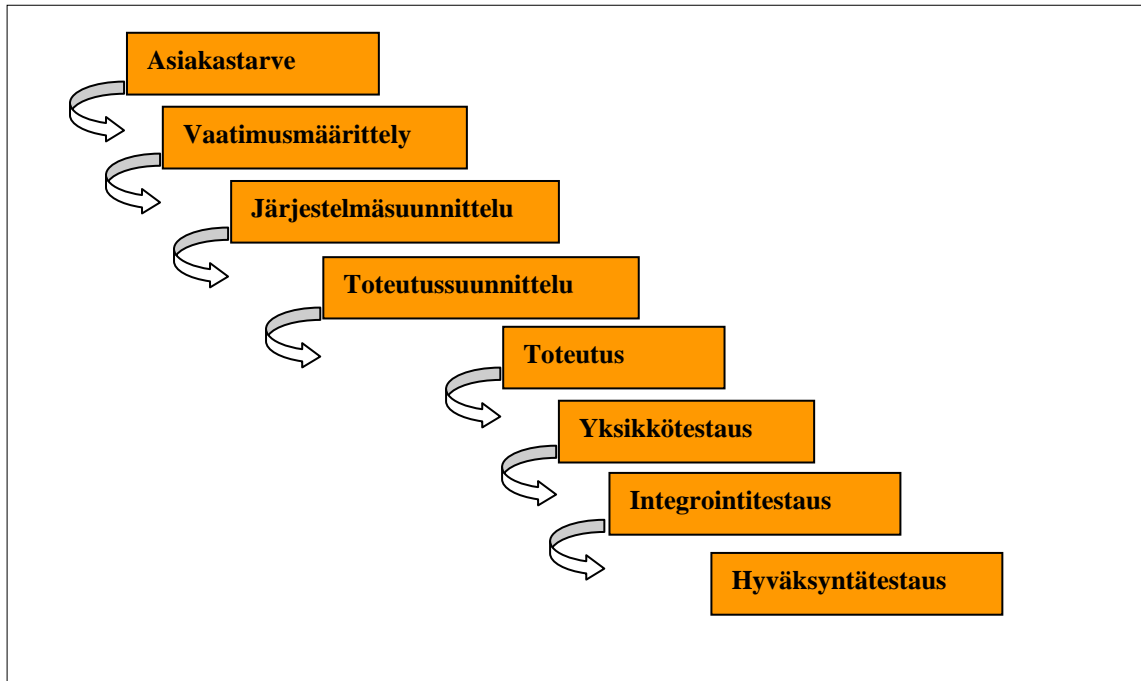
Kaikki ohjelmistokehityshankkeet noudattavat jotain elinkaarta – tietoisesti tai tiedostamatta. Erilaisia elinkaarimalleja on lukuisia, joista monet ovat variaatioita muutamasta perusmallista. Ne voidaan jakaa ennusteellisiin ja adaptiivisiin malleihin [1]. Ennusteellisen elinkaarimallin mukaisen projektin kohde voidaan selkeästi määrittellä ja sen aikataulu sekä kustannukset ovat ennustettavissa. Adaptiiviset mallit olettavat, että kehitystyö mukautuu projektin aikana tapahtuviin muutoksiin, koska tuotteen vaatimuksia ei pystytä kuitenkaan riittävän hyvin määrittämään elinkaaren alkuvaiheessa.

2.1 Ennusteelliset elinkaarimallit

Ennusteellisille malleille on tyypillistä etupainotteinen määrittely ja analyysi, jossa pyritään saamaan selkeä kuva koko projektista. Selvitystyöllä pyritään minimoimaan projektin aikana tapahtuvat muutokset ja saamaan mahdollisimman tarkka ennustettavuus projektin suunnitteluparametreille. Seuraavissa luvuissa käydään tarkemmin läpi ennusteelliset mallit, joihin Archibald [1] projektihallintaa käsittelevässä artikkelissaan viittaa.

2.1.1 Vesiputousmalli

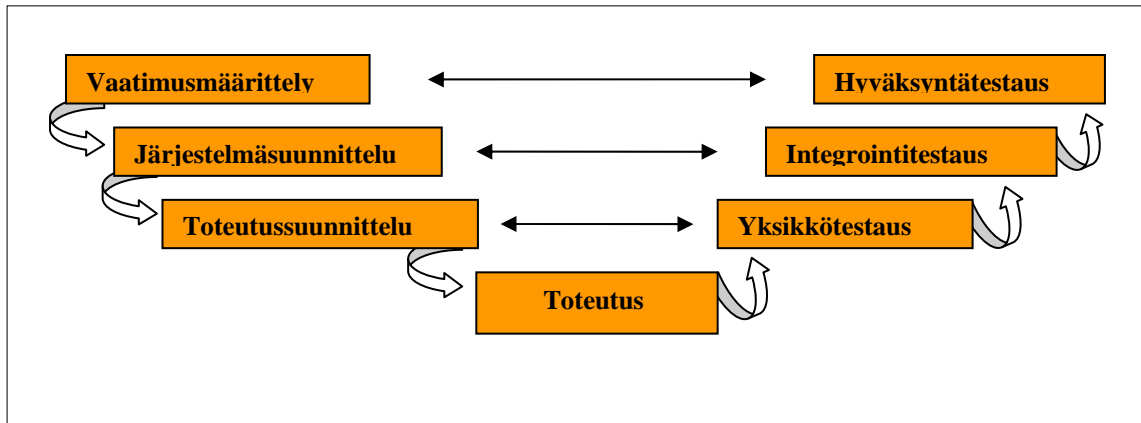
Vesiputousmalli on tunnetuin perinteisistä elinkaarimalleista. Se julkaistiin vuonna 1970 ja sen käyttö on ollut hyvin laajaa. Siitä kehittyi puhdas vaiheittainen malli, jossa seuraava vaihe ei voinut alkaa ennen kuin edellinen vaihe oli tehty ja hyväksytty. Ajatuksena oli löytää tuotoksiin tehdyt virheet mahdollisimman aikaisessa vaiheessa.[2].



Kuva 1. Esimerkki Vesiputousmallin vaiheista

Kuva 1 esittää esimerkin tyypillisistä vesiputousmallin vaiheista. Malli on selkeä ja järjestelmällinen. Suurimpana ongelmana on se, että todelliset projektit eivät normaalisti etene puhtaasti vaiheissa. Yleensä osa projektin vaatimuksista selviää vasta projektin edetessä ja etenkin pitkissä projekteissa tulee myös kokonaan uusia vaatimuksia vielä myöhäisessäkin vaiheessa. Jokaisen vaiheen katselmoinnit ja hyväksynnit saattavat kasvattaa turhaa byrokratiaa. Mallin taustalla oleva perusajatus, että ennen toteutusta pitää olla ymmärrys siitä mitä ollaan tekemässä, näkyy myös muissa malleissa. Vesiputousmallissa ymmärrys tarpeista ja projektin määrittely tapahtuu projektin alkuvaiheessa koko projektille – muissa malleissa se tapahtuu useassa vaiheessa projektin edetessä.

V-malli on Vesiputousmalli, jossa testauksen vaiheet on selkeämmin linkitetty suunnittelun ja toteutuksen vaiheisiin. Esimerkiksi järjestelmätason vaatimusmäärittelyn ja suunnittelun yhteydessä suunnitellaan myös järjestelmätason testausta ja huomioidaan testauksen aiheuttamat tarpeet toteutukselle. Kuva 2 esittää V-mallin esimerkkivaiheita.



Kuva 2. Esimerkki V-mallin vaiheista

V-malli etenee vesiputousmallin mukaisesti vaihe kerrallaan, mutta tulevien testausvaiheiden tarpeita huomioidaan jo määrittelytyön yhteydessä. Testausvaiheiden suunnittelussa ja testattavien osakokonaisuuksien määrittelyssä pyritään mahdollisimman aikaiseen testaukseen.

”Puhtaassa vesiputousmallissa olevat toiminnot ovat sisäänrakennettuja ohjelmistokehityksessä. Niitä ei voida välttää. Jollain tavalla on kehitettävä ohjelmiston konsepti ja jostain on saatava vaatimukset. Vaatimusten keräämiseen ei tarvitse käyttää vesiputousmallia, mutta jotain on käytettävä. Vastaavasti ei voida välttyä arkkitehtuurilta, suunnittelulta tai koodaamiselta.” [3, s. 143].

Vesiputousmallin tunnetut ongelmat eivät siis liity sen vaiheiden pääsisältöön vaan niiden jaotteluun ja toteutustapaan. Hyvin toimivan mallin tulee sisältää vesiputousmallin tärkeimmät vaiheet jossain muodossa. Niitä voidaan jakaa useampaan vaiheeseen ja toistaa tarvittaessa. Tapauskohtaisesti voidaan harkita dokumentaation ja käytäntöjen yksinkertaistamista. Erilaisia enemmän tai vähemmän läheisesti vesiputousmallia muistuttavia malleja on julkaistu lukuisia.

2.1.2 Prototyypimalli

Prototyypimalli ei ole yksiselitteinen ja sen täsmällisestä määrittelystä on erilaisia mielipiteitä. Pääpiirteenä mallissa on kuitenkin tehdä rajatulla toiminnallisuudella demonstraatio lopputuotteen oleellisesta toiminnallisuudesta. Protoilun tulokset yhdistettyinä muihin systeemisuunnittelun prosesseihin parantavat lopullisen järjestelmän laatua. Prototoilun tärkeä rooli on toimia kommunikaation apuna asiakkaan ja toteutusryhmän vä-

lillä. Sen avulla saadaan parempi ja konkreettinen yhteisymmärrys järjestelmän halutusta toiminnallisuudesta. Prototyypimallia voidaan käyttää myös toteutusryhmän sisäisesti varmistamaan teknistä toteutuskelpoisuutta.[4]

Prototyypimallinnuksen vaiheet ovat: toiminnallisuuden valitseminen, toteutus, arviointi ja jatkokäyttö. Toiminnallisuus tulisi valita niin, että se kattaa demonstraation tarpeet, mutta ei sisällä ylimääräistä toiminnallisuutta, ja lisää täten tämän vaiheen työmäärää. Toteutuksessa voidaan tehdä myös kompromisseja lopputuotteen yleisissä ominaisuuksissa kuten laatu, luotettavuus ja tehokkuus, mikäli ne eivät ole oleellisia demonstraation kannalta. Arviointi on tärkeä vaihe, koska siinä syntyy prototyypimallinnuksen päätuotos. Jatkokäytössä määritellään miltä osin ja miten prototyyppiä käytetään jatkossa. Se voi jäädä ainoastaan tutkimus- ja opettelualueksi, ja tulla tarpeettomaksi myöhemmässä vaiheessa, tai sitä voidaan hyödyntää osittain tai kokonaan lopputuotteen komponenttina.

C.Floyd [4] jaottelee prototyypimallinnuksen kolmeen pääluokkaan sen tarkoituksen perusteella:

- *Tutkimuskäytössä* päätarkoitus on kohdejärjestelmän vaatimusten ja ominaisuuksien selkeyttäminen
- *Kokeilussa* testataan ehdotettu ratkaisu ennen kuin aloitetaan laajamittainen toteutustyö
- *Evoluutiossa* mallinnuksen tarkoitus on sopeuttaa kohdejärjestelmän kehitystyön edistyessä muuttuviin vaatimuksiin.

Protoilu ei itsenäisenä tuotekehitysmallina ole kovin täydellinen elinkaarimalli. Se soveltuu vaikeiden osa-alueiden vaatimusmäärittelyn tueksi ja toisaalta antaa hyvän perustan inkrementaaliseen elinkaarimallille kun sitä täydennetään muiden mallien prosessialueilla. Erityisesti resurssien käytön tehokkuutta painottavissa organisaatioissa hyödyttömäksi jäävän prototyypin vaatimaa työmäärää on vaikea hyväksyä, mikä aiheuttaa painetta hyödyntää prototyyppiä kehitystyön evoluutiovaiheena. Tässä on kuitenkin riski, tai jopa mallien luonteesta johtuva periaatteellinen ristiriita. Jos prototyypimallinnus ymmärretään nopeaksi tavaksi toteuttaa toiminnallisuusdemonstraatio esimerkiksi laadun ja luotettavuuden kustannuksella, prototyypin käyttö sellaisenaan osana lopputuotetta siirtää nämä kompromissitekijät lopullisen järjestelmän ongelmiksi. Toisaalta, jos prototyypin rajatun toiminnallisuuden kehityksessä huomioidaan kaikki lopullisen järjestelmän kehityksessä vaadittavat laatu- ja suorituskykytekijät, ei saavuteta optimaalista prototyypin kehitysaikaa eikä välttämättä puhuta enää edes prototyypistä vaan lopullisen tuotteen kehitysvaiheen väliversiosta.

2.1.3 RAD

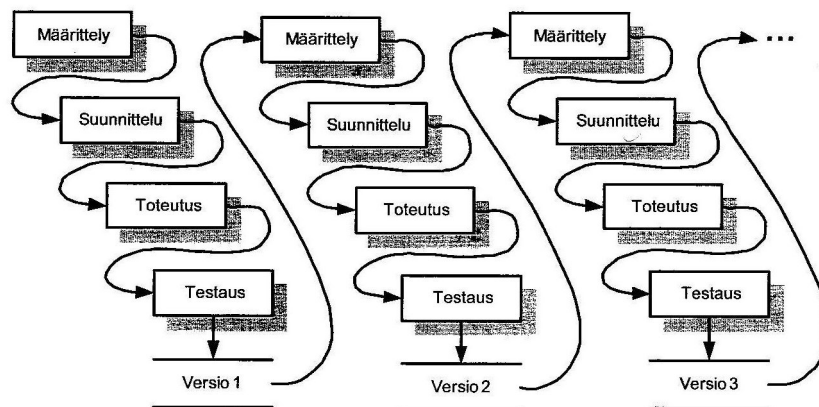
RAD (*Rapid Application Development*) eli nopea sovelluskehitys määriteltiin alun perin vuonna 1991 [5]. Menetelmä korosti tiukkoja aikatauluja, joihin päästäisiin prototyyppien ja ylempien tason suunnittelutyökalujen sekä tekniikoiden avulla. Menetelmän etujen (helppous, parantunut käyttäjätyytyväisyys, nopeampi tuotekehitys) vastapainoksi

raportoitiin myös heikkouksista (huonosti suunnitellut tuotteet, kokeneen tuotekehityshenkilöstön tarve, tarvitaan vahva projektin johtaminen ja kontrolli) [6]. Alkuperäisen tietokantakeskeisen ympäristönsä ulkopuolella RAD ei ole onnistunut nousemaan asialliseksi ohjelmistokehitysmenetelmäksi [3, s. 366].

DSDM-yhteenliittymän (*Dynamic Systems Development Method*) päämäärä oli saada korostetuksi RAD-menetelmän etuja ja ratkaista esiintyneitä ongelmia. DSDM-menetelmä julkaistiin vuonna 1995 ja sitä pidetään ensimmäisenä täysin ketteränä ohjelmistokehitysmenetelmänä [7]. Ketteristä ohjelmistokehitysmenetelmistä on kerrottu enemmän luvussa 2.2.

2.1.4 Inkrementaalinen malli

Inkrementaalisisessa mallissa lopullista toteutusta tehdään useassa peräkkäisessä ketjussa. Jokainen inkrementti on lyhykestoinen vesiputous, jonka kesto on muutamasta viikosta muutamaan kuukauteen. Inkerementtien tuotos ei ole prototyyppi vaan tuotantotasoinen toteutus lopullisen järjestelmän osasta. Kuva 3 hahmottelee inkrementaalisen kehityksen vaiheita



Kuva 3. Inkrementaalisen mallin tuotekehityssykli [8, s. 44]

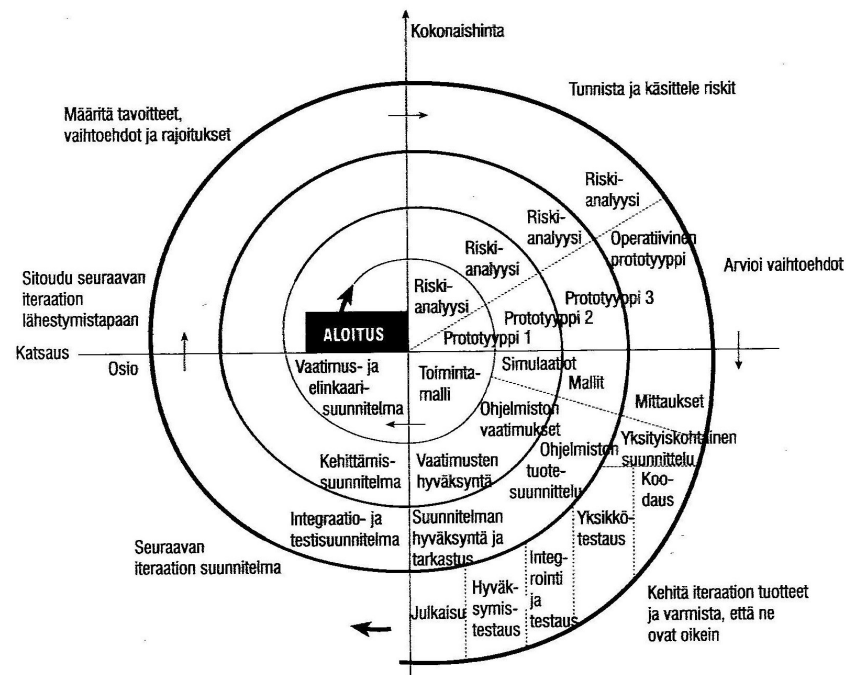
Inkrementaalisisessa mallissa on paljon etuja. Vaatimusmäärittely ja niiden analyysi tapahtuu jokaiselle vaiheelle erikseen. Täten kaikkien vaatimusten ei tarvitse olla valmiina projektin alussa. Vaatimukset priorisoidaan ja sijoitetaan eri toteutusvaiheisiin. Priorisointiperusteena voidaan käyttää vaatimuksen sisältämiä riskejä tai niiden merkitystä käyttäjälle. Projektin asiakkaan ja kehitystiimin kommunikaation kannalta on erittäin hyödyllistä, että malli tuottaa jo varhaisessa vaiheessa välituotteita asiakkaan arviotavaksi. Näiden perusteella voidaan muuttaa ja tarkentaa seuraavien vaiheiden vaatimuksia. Yhden inkrementtivaiheen kehitystyössä pystytään mallin avulla eristämään erilaiset projektityön häiriötekijät ja voidaan keskittyä vain määrittelyyn osuuden tekemiseen.

Mallin heikkous on sen vaatima työmäärä. Se vaatii hyvää suunnittelua ja projektihallintaa. Asiakkaan saamien kehitysversioiden etujen lisäksi niihin liittyy myös inkrementaalimallien ongelma. Ensimmäisissä versioissa mahdollisesti olevien ongelmien selvittely ja korjaus voi viedä projektiryhmän kaiken ajan, ja seuraavien inkrementtien kehitys pysähtyy [8, s. 47]. Toisaalta, jos eri vaiheisiin liitettyjen vaatimusten ja niiden toteutuksesta löytyneiden virheiden priorisointi on tehty oikein, pitäisi alkuvaiheessa tehtävät kriittisten alueiden virhelöydökset ja korjaukset olla mallin kannalta positiivinen asia. Huomattavasti vakavampaa on löytää kriittisten alueiden virheitä vasta projektin loppuvaiheessa. Käytännössä asiakkaan ja kehitysryhmän näkemys virheiden prioriteetista ei ole aina yhtenevä. Tällöin kehitysryhmä voi joutua keskittymään virhekorjauksiin, jotka teknisen prioriteetin perusteella voisi viivästyä.

2.1.5 Spiraalimalli

Boehm [9] esitteli Spiraalimallin vuonna 1988. Spiraalimalli yhdistää edellisten mallien ominaisuuksia, mutta se lisää niihin voimakkaan riskilähtöisen lähestymistavan. Tosin ”Riskin” käsite on mallissa laaja ja voi tarkoittaa huonosti ymmärrettyjä vaatimuksia, huonosti ymmärrettyä arkkitehtuuria, mahdollisia suorituskykyongelmia ja niin edelleen [3, s. 141]. Kuva 4 havainnollistaa spiraalimallia. Projekti alkaa spiraalin keskeltä ja kiertyy vähitellen kohti ulkokehää. Yksi kierros muodostaa yhden iteraation ja kehän sektorit kuvaavat jokaisessa iteraatiossa tapahtuvaa samantyyppistä toimintoa. Nämä päätoiminnot on nimetty spiraalin ulkokehällä:

1. Selvitä tavoitteet, vaihtoehdot ja rajoitukset
2. Tunnista ja selvitä riskit
3. Arvioi vaihtoehdot
4. Kehitä iteraatioon kuuluvat toimitukset ja varmista, että ne ovat oikein
5. Suunnittele seuraava iteraatio
6. Sitoudu lähestymistapaan seuraavaa iteraatiota varten (jos sellainen tarvitaan). [3]



Kuva 4. Spiraalimalli [3]

Spiraalin keskellä olevat vaiheet ovat halvimpia ja riskit suurimpia. Vaiheiden kustannukset nousevat ja riskit pienenevät ulospäin mentäessä. Isoimmat riskit pystytään varmistamaan ennen kuin projektissa edetään kustannuksiltaan kalliimpiin vaiheisiin. Tässä vaiheessa jäljellä pitäisi olla vain pienempiä riskejä joiden realisoituminen ei aiheuta niin suuria kustannuksia kuin jos tähän vaiheeseen olisi jäänyt kriittisiä riskejä.

Spiraalimallin ainoa haittapuoli on, että se on monimutkainen [3, s. 143]. Se vaatii hyvän projektihallinnon ja kontrollin. Seuraaviin iteraatiokierroksiin siirtymistä kontrolloivien tarkistuspisteiden määrittäminen voi olla vaikeaa. Riskien ja kustannusten keskinäisen hallinnan lisäksi mallin etuina ovat muut iteratiivisten mallien edut. Iteraatiovaiheissa toteutettavat prototyypit antavat hyvän välineen asiakkaan ja kehitysryhmän väliseen vaatimusten tarkentamiseen.

2.2 Adaptiiviset ja ketterät elinkaarimallit

Adaptiiviset elinkaarimallit eivät painota ennusteellisten mallien tavoin projektin etupainotteista määrittelyä ja suunnittelua, koska tuotteen vaatimuksia ei panostuksesta huolimatta pystytä määrittelemään riittävän hyvin elinkaaren alkuvaiheessa. Adaptiivisissa malleissa hyväksytään projektin aikana tapahtuvat muutokset ja pyritään luomaan menetelmät, jotka toimivat tehokkaasti muuttuvassa projektiympäristössä.

Vuonna 2001 seitsemäntoista adaptiivisten elinkaarimallien edustajaa julkaisivat ketteriä (*agile*) ohjelmistokehitysmenetelmiä kuvaavan julkilausuman, ketterä manifesti (*Agile Software Development Manifesto*) [10]. Adaptiivisia malleja kutsutaankin myös ketteriksi elinkaarimalleiksi. Nämä menetelmät ovat olleet merkittävässä roolissa viime vuosien aikana tapahtuneessa tietotekniikka-alan kehityksessä. [1]

Ketterät ohjelmistokehitysmenetelmät ovat melko heterogeeninen joukko erilaisia menetelmiä. Menetelmiä yhdistää ketterässä manifestissa julkaistut arvot:

”Me etsimme parempia keinoja ohjelmistojen kehittämiseen tekemällä sitä itse ja auttamalla siinä muita. Tässä työssämme olemme päätyneet arvostamaan

Yksilöitä ja vuorovaikutusta enemmän kuin prosesseja ja työkaluja

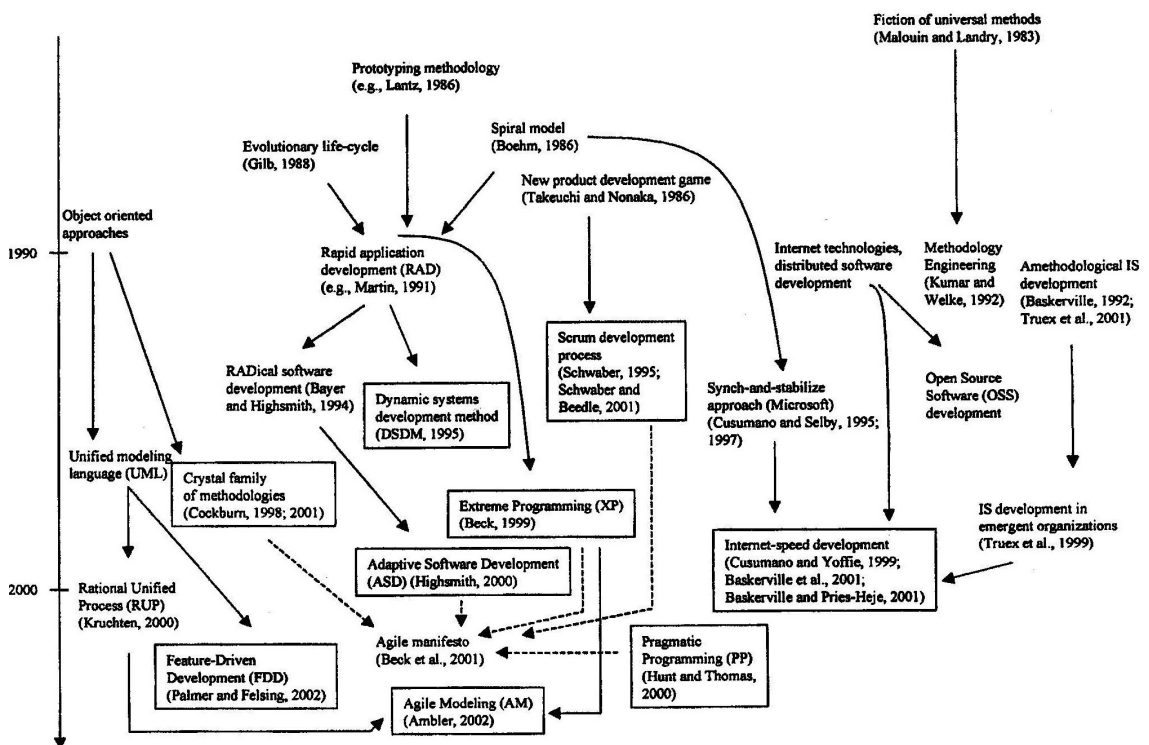
Toimivaa sovellusta enemmän kuin kokonaisvaltaista dokumentaatiota

Asiakasyhteistyötä enemmän kuin sopimusneuvotteluita

Muutokseen reagoimista enemmän kuin suunnitelman noudattamista.

Vaikka oikeallakin puolella on arvoa, me arvostamme vasemmalla olevia asioita enemmän.” [10]

Neljän perusarvon lisäksi manifesti määrittelee myös 12 ketterän kehityksen periaatetta. Tunnetuimpia ketteriä menetelmiä ovat *Adaptive Software Development (ASD)*, *Feature Driven Development (FDD)*, *Crystal Clear*, *Dynamic Software Development Method (DSDM)*, *Rapid Application Development (RAD)*, *Scrum*, *Extreme Programming (XP)*, *Rational Unify Process (RUP)*. Kuva 5 havainnollistaa ketterien menetelmien keskinäistä suhdetta ja liittymistä ketterään manifestiin.



Kuva 5. Ketterien menetelmien evoluutio [7]

Abrahamsson et al. [7] julkaisivat vuonna 2003 ketterien menetelmien vertailevan tutkimuksen. Tutkimuksessa vertailtiin yhdeksää menetelmää viidestä eri näkökulmasta: menetelmän tuki ohjelmistokehityksen elinkaaren eri vaiheille, menetelmän tuki projektihallintakäytännöille, abstraktien periaatteiden suhde konkreettisiin ohjeisiin, menetelmän yleispätevyys verrattuna tapauskohtaisen soveltamisen sallimiseen ja löytyykö menetelmän väitteille empiiristä tukea. Tutkimuksessa todettiin ketterien menetelmien olevan melko hajanainen joukko toimintamalleja, jotka painottavat erilaisia ohjelmistokehityksen osa-alueita. Ne tukevat tiettyjä toisistaan poikkeavia elinkaarivaiheita. Suurin osa ei tue riittävästi projektihallintaa ja valtaosassa dominoivat abstraktit periaatteet. Myös menetelmistä julkaistu kokemusperäinen tutkimustieto on hyvin rajallista.

Vuonna 2008 Dybå ja Dingsøyr [11] julkaisivat laajan katsauksen ketterien menetelmien empiirisestä tutkimuksesta. He totesivat, että näyttö tutkittujen menetelmien hyödyistä tai haitoista on vielä hyvin heikko. Alueen tutkimuksen määrää ja laatua tulee nostaa, jotta menetelmistä saataisiin luotettavaa tietoa.

Koska ketterät menetelmät kattavat kukin vain osan ohjelmistokehityksen vaiheista [7], niiden käyttö koko elinkaaren määrittävänä adaptiivisena mallina vaatii useamman menetelmän yhdistettyä käyttöä. Tämä edellyttää menetelmien ja niiden sovellusalueiden hyvää tuntemusta. Myös menetelmien soveltuvuuden arviointi juuri kyseiseen ohjelmistokehityshankkeeseen ja organisaatioon voi olla hankalaa koska menetelmistä löytyvien konkreettisten ohjeiden saatavuus vaihtelee [7]. Tuloksinvaraisuuden ja rajallisen käyttöalueen takia kokematon henkilö voi ketteriin menetelmiin tukeutuessaan laiminlyödä olennaisia ohjelmistokehityksen osa-alueita.

2.3 Ohjelmistoprojektin toimintaprosessin valinta

Formaalien prosessien tarkoitus on kohottaa toiminnan laatua, tehostaa toimintaa ja parantaa näkyvyyttä projektin tilanteeseen. Monissa tapauksissa tämä tarkoittaa lisääntyvää työpanosta prosessin määrittämiin alueisiin, joihin ei, ainakaan kaikkiin, tulisi ilman määriteltyä projektiprosessia panostettua. Toisaalta 2000-luvun aikana yleistyneiden ketterien ohjelmistokehitysmenetelmien perusideologia, *Agile Manifesto* [10], kritisoi perinteisten, vaihejakoiseen prosessiajatteluun pohjautuvien projektihallintamenetelmien keskittyvän liikaa prosessin määrittelemään toimintaan ja toiminnan dokumentointiin. Ketterät menetelmät haluavat korostaa enemmän ihmisten vuorovaikutusta kuin prosesseja ja työkaluja [10]. Ohjelmistoprojektien erilaisen luonteen vuoksi sama toimintamalli ei ole kuitenkaan tehokkain toimintatapa niille kaikille.

Ohjelmistoprojektin koolla on merkittävä vaikutus toimintamalliin. Kun laaja ohjelmistohanke halutaan suorittaa kohtuullisessa ajassa, se vaatii enemmän ohjelmistosuunnittelijoita kuin pieni hanke. Suuren, esimerkiksi useista 5-15 hengen projektiryhmistä muodostuvan projektin ohjaaminen, tiedonvälitys ja seuranta vaativat erilaisia käytäntöjä kuin pienen, 4-6 hengen projektin vaatimat menetelmät. Perinteiset projektihallintaprosessit tukevat hyvin laajan projektin vaatimaa hyvää etukäteissuunnittelua

ja dokumentointitasoa. Isommissa hankkeissa myös projektinhallintamenetelmien aiheuttama työmäärä verrattuna varsinaiseen toteutustyön määrään jää pienemmäksi kuin suppeammassa hankkeessa. Tarpeettoman raskas prosessi pienessä ohjelmistohankkeessa tekee lopputuloksen tekemisestä tehotonta. Ylimoitettu prosessityö on varmasti yksi merkittävä syy ketterien menetelmien syntyyn. Ketterät menetelmät korostavat välitöntä kommunikaatiota, vuorovaikutusta ja itse organisoituvia projektiryhmiä [10]. Nämä periaatteet soveltuvat erinomaisesti tiiviin projektiryhmän tekemään itsenäiseen pienempään ohjelmistohankkeeseen mutta ovat sovellettavissa myös isompaan projektiin, kunhan menetelmän erityispiirteet ja projektin luonne sovitetaan yhteen. Vastaavasti myös perinteiset projektinhallintamenetelmät soveltuvat pieneen hankkeeseen kunhan menetelmä sovitetaan olosuhteisiin.

Toimintaympäristö vaikuttaa myös ohjelmistohankkeeseen soveltuvaan projektinhallintamenetelmään. Esimerkiksi pieneenkin projektiin voidaan tarvita suurta työmäärää vaativa hallintaprosessi, jos toteutus on osa turvallisuuskriittistä järjestelmää tai mikäli asiakas vaatii tietyn prosessin käyttöä projektin laajuudesta riippumatta. Ohjelmistoprojektin toimintaympäristön organisaatiotekijöillä on myös merkittävä vaikutus hallintamallien toimintaan. Organisaation rakenteen tulee tukea valittua toimintamallia, jotta toiminta olisi tarkoituksenmukaista. Projektiorganisaation oma kokemus ja erilaisten projektinhallinta- ja ohjelmistotuotantomenetelmien käyttäminen määrittelee organisaation projektinhallinnallisen kypsyystason ja sitä kautta kyvyn omaksua tehokkaasti uusia menetelmiä. Luvussa 5.1 käsiteltävät yleisimmin käytössä olevat projektiprosessien arviointimenetelmät perustuvat menetelmien jaotteluun eri kypsyystasoille. Pitkälle kehittyneiden prosessimenetelmien käyttöönotto vaatii yleensä ainakin joidenkin alempien kypsyystasojen menetelmien käyttämistä toimiakseen tehokkaasti.

2.4 Ohjelmistohankintaprojektin hallinta

Ohjelmiston hankintaprosessin elinkaari noudattaa luvun 2 alussa esitettyjä projektin yleisiä elinkaarivaiheita: Hahmotus, Määrittely, Suoritus, Lopetus. Myös edellä käsitellyt ohjelmistokehityksen tarkempia elinkaarimalleja soveltaen voidaan määrittää ohjelmistohankintaprojektille soveltuva elinkaarimalli ja sen pohjalta organisaatio- ja projektikohtainen prosessi. Ohjelmistokehitysmallien soveltaminen hankintaprojektin tarpeisiin on luonteva vaihtoehto, koska mallit pitää normaalissa ohjelmistokehityksessäkin sopeuttaa kunkin organisaation ja projektin tarpeisiin. Vaikka käytettäisiinkin valmista ohjelmistohankintamallia, täytyy sen mukauttamistarpeet tarkistaa ennen käyttöönottoa.

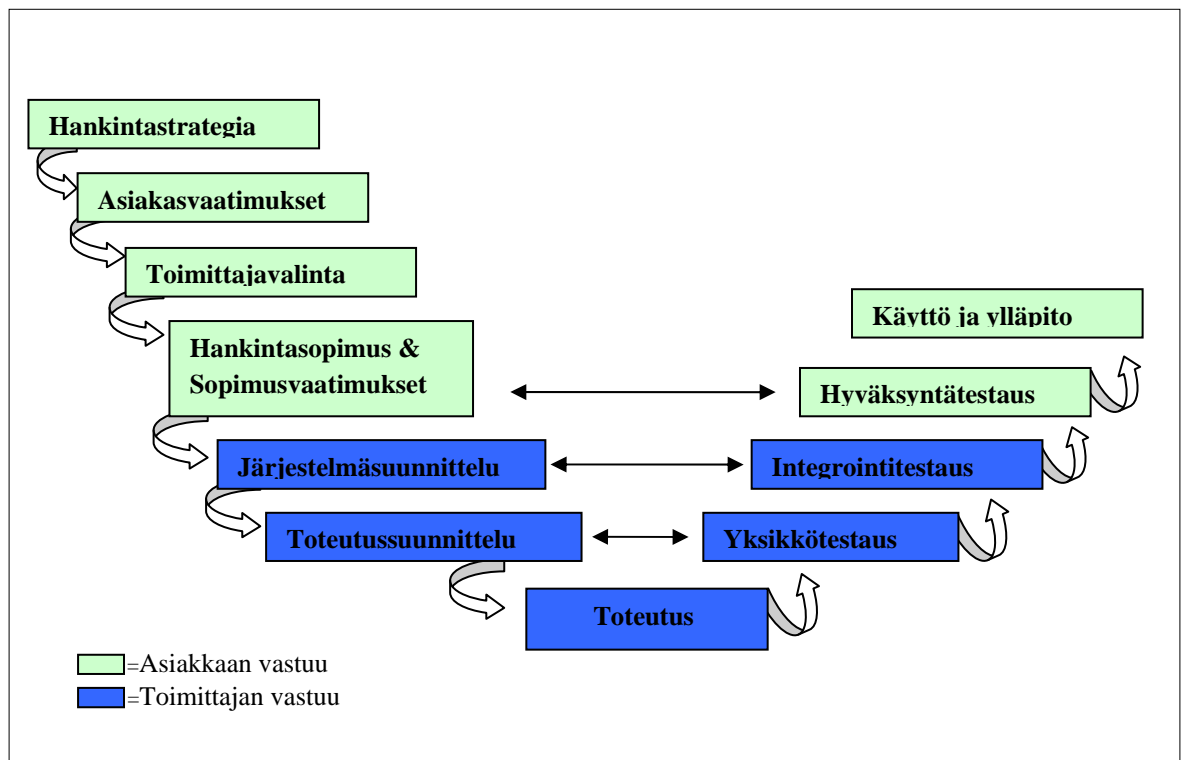
2.4.1 Elinkaarimallien muokkaaminen ohjelmistohankintaan

Ohjelmistohankintaan kuuluu samat päävaiheet kuin ohjelmistokehitykseenkin. Lähtökohta on molemmissa tarpeiden määrittäminen, ja lopputulos on valmiin ohjelmiston toimitus

ja ylläpito. Toimittajan roolin vaikutus kaikkiin elinkaaren vaiheisiin täytyy ottaa huomioon, ja täysin uusina alueina ovat toimittajavalintaan ja sopimushallintaan liittyvät aktiviteetit. Järjestelmän elinkaaren vaiheiden jakautuminen asiakkaan ja toimittajan välille on tapauskohtaista. Mikäli kyseessä on järjestelmäkokonaisuuden hankkiminen, järjestelmätason toiminnot on useimmiten järkevintä antaa toimittajan tehtäväksi. Mikäli toimittajalta hankitaan vain osia isompaan järjestelmään, järjestelmätason toiminnot on usein tarkoituksenmukaisinta pitää asiakkaalla. Monilla tekijöillä on kuitenkin vaikutusta toimivimpaan vastuunjakoon.

Esimerkiksi asiakkaan heikko tekninen osaaminen ja vähäinen kokemus ohjelmistokehityksestä on perusteltu syy siirtää mahdollisimman paljon projektivaiheita toimittajalle. Asiakasvaatimusten tarkempi analyysi ja tekninen jatkoselvittely voidaan antaa toimittajan tehtäväksi. Toimittajan tekemän jatkomäärittelyn tulosten perusteella voidaan muotoilla sopimusvaatimukset. Täytyy kuitenkin muistaa, että vastuuta lopullisen järjestelmän toiminnallisista ominaisuuksista ei voi ulkoistaa. Asiakkaan täytyy kontrolloida ja hyväksyä määritelty kokonaisuus ja valvoa toimittajan työtä.

Vesiputousmallin laajentaminen sopimushallintaan ja toimittajavalintaan liittyvillä vaiheilla antaa yksinkertaisen näkymän ohjelmistohankinnan elinkaareen. Kuva 6 havainnollistaa tätä vesiputousmallin V-mallimuunnoksella. Esimerkin vastuujako on tapauskohtainen



Kuva 6. Esimerkki ohjelmistohankinnan elinkaarimallista

Monet vesiputousmallin tunnetuista ongelmista heijastuvat erityisesti ohjelmistohankintaan ja tämän vuoksi se onkin harvoin hyvä vaihtoehto ohjelmistohankinnan elinkaarimalliksi. Kehitystyön aikana sekä asiakkaan että toimittajan käsitys alkuperäisistä vaatimuksista selkeytyy. Tämä synnyttää tarpeita vaatimuksien muokkaamiseen ja uusien luomiseen. Siksi luvussa 2.1 käsitellyt inkrementaaliset mallit (spiraalimalli muokkauksien) tukevat paremmin hankinnan vaiheita. Jokaisen inkrementtiversioon tai prototyypivaiheen jälkeen asiakas saa konkreettisen toteutuksen kokeiltavakseen ja inkrementaaliset mallit tukevat uusien määrittelyjen tekemistä jokaiselle vaiheelle erikseen. Vesiputousmallin perusvaiheet kuitenkin kuvaavat hyvin jokaisen inkrementin sisäistä vaihejakoa.

Projektin toimintamallia sekä asiakkaan ja toimittajan työnjakoa määriteltäessä kannattaa toimittajan oma ohjelmistokehitysprosessi huomioida mahdollisuuksien mukaan. Asiakasorganisaation ohjelmistohankinnan elinkaarimalli tulisi olla toimittajariippumaton, ettei se rajaa käytettävissä olevia toimittajavaihtoehtoja. Isoissa hankkeissa voi olla kuitenkin perusteltua muokata asiakkaan ja toimittajan toimintatapoja tehokkaaksi kokonaisuudeksi.

Pitkäkestoisissa ja laajoissa järjestelmähankinnoissa tehokkaimpaan toimittajakilpailutukseen päästään, jos eri vaiheet voidaan kilpailuttaa erikseen. Usein kuitenkin kaikille toimittajille avoimien järjestelmärajapintojen puuttuminen johtaa siihen, että käytännössä asiakas sitoutuu alkuperäiseen toimittajaan koko järjestelmätoimituksen ajaksi.

Ketterät menetelmät jo perusjulkilausumansakin [10] mukaan korostavat voimakkaasti asiakasyhteistyön tärkeyttä. Asiakas on tiiviissä yhteistyössä kehittäjien kanssa koko projektin ajan. Määritelmiä ja käyttäjäkertomuksia muutetaan asiakkaan tarpeiden mukaan. Jos ohjelmistohankinta toimii aidosti ketterien periaatteiden mukaan, se vaatii myös asiakkaalta ymmärrystä näistä menetelmistä. Se täytyy ottaa huomioon myös sopimuskäytännöissä, koska hankkeen sisältö ei ole niin yksityiskohtaisesti määriteltävissä kuin ennusteellisia elinkaarimalleja käytettäessä.

2.4.2 IEEE std 1062

Kansainvälinen standardointijärjestö IEEE (Institute of Electrical and Electronics Engineers) on julkaissut ohjeiston ohjelmistohankinnasta, *IEEE Recommended Practice for Software Acquisition*, IEEE std 1062 [12]. Standardi määrittelee ohjelmistohankinnan elinkaaren sekä yhdeksän ohjelmistohankintaprosessin vaihetta. Standardissa esitetty elinkaari ja prosessi perustuvat vaihemalliin ilman iteraatiota. Se antaa kuitenkin hyvän perussisällön myös iteratiivisiin malleihin sovellettavaksi. Standardin lopussa on myös yksityiskohtaisia tarkistuslistoja tukemaan prosessin vaiheiden toteutusta.

Ohjelmistohankinnan elinkaaren vaiheet ovat IEEE std 1062:n mukaan suunnitteluvaihe hankintatarpeen muodostamisesta tarjouspyynnön lähettämiseen, sopimusvaihe tarjouspyynnöstä sopimuksen allekirjoittamiseen, tuotteen toteutusvaihe sopimuksen allekirjoittamisesta tuotteen vastaanottamiseen, tuotteen hyväksyntävaihe ja jälkivaihe.

Taulukko 1. IEEE std 1062 ohjelmistohankinnan elinkaaren ja prosessin vaiheet [12]

Elinkaaren vaihe	Prosessin vaihe
Suunnitteluvaihe	(1) Organisaation hankintastrategian suunnittelu
	(2) Organisaation hankintaprosessin määrittäminen
	(3) Ohjelmiston vaatimusten määrittäminen
Sopimusvaihe	(4) Mahdollisten toimittajien tunnistaminen
	(5) Sopimusvaatimusten valmistelu
	(6) Tarjousten arviointi ja toimittajavalinta
Tuotteen toteutus	(7) Toimittajan suorituksen seuranta
Tuotteen hyväksyminen	(8) Ohjelmiston hyväksyminen
Jälkivaihe	(9) Ohjelmiston käyttäminen

Taulukko 1 näyttää mihin elinkaaren vaiheeseen kukin prosessivaihe kuuluu. [12]. Standardin noudattaminen ei edellytä kaikkien prosessikäytäntöjen toteuttamista vaan sallii prosessin räätälöinnin.

3 Terveydenhuollon tietojärjestelmät

Sähköisten potilastietojärjestelmien laajentunut käyttöönotto 2000-luvulla on merkinnyt voimakasta muutosta Suomen terveydenhuollon toimintaympäristössä. Potilaskertomusjärjestelmät ovat kiinteä osa terveydenhuollon organisaatioiden ydintoiminnan prosesseja ja muutokset näissä järjestelmissä vaikuttavat laajasti koko organisaation toimintatapoihin.

3.1 Sähköisten potilastietojärjestelmien kehitys

Vuonna 2001 erikoissairaanhoidon potilaskertomusjärjestelmät olivat pääosin vielä paperilla. Perusterveydenhuollossa 63 % terveyskeskuksista käytti sähköistä potilaskertomusjärjestelmää. [13, s. 60]. Vuonna 2003 sähköisiä järjestelmiä käytti peräti jo 93,6 % Suomen terveyskeskuksista. Erikoissairaanhoidossakin sähköisten potilaskertomusjärjestelmien käytön osuus oli noussut 62 %:iin sairaanhoitopiireistä. [14]. Vuonna 2006 kaikki sairaanhoitopiirit yhtä lukuun ottamatta käyttivät jo sähköistä kertomusjärjestelmää ja järjestelmien käyttöasteet sekä erikoissairaanhoidossa että perusterveydenhuollossa olivat pääosin yli 90 % [15]. Tietojärjestelmien kehitys onkin viimeisen viiden vuoden aikana siirtynyt organisaatiokeskeisestä vaiheesta organisaatioiden välisen toiminnan kehittämiseen.

Organisaatioiden välinen toiminta tässä yhteydessä tarkoittaa tilanteita, joissa terveydenhuollon palveluntuottajat lähettävät, vastaanottavat tai lukevat potilastietoja yli henkilörekisteripitäjärajojen. Pisimpään on ollut käytössä potilastietoihin liittyvien sanomien välittäminen salattuja tietoliikenneyhteyksiä käyttäen. Uusimpina menetelminä ovat yhteiset potilastietojärjestelmät ja sähköiset arkistot. Organisaatioiden välillä välitetään potilaan kertomustietoja, laboratoriotuloksia ja kuvastamistutkimuksen tietoja. Vuonna 2007 organisaatioiden väliseen sähköisen potilastiedon välittämiseen käytettäviä järjestelmiä oli käytössä noin 90%:lla sairaanhoitopiireistä. [16]. Alueellisen toiminnan tehostamistarpeiden lisäksi tietojärjestelmien integraatioon ohjasi julkishallinnon strategialinjaukset ja lainsäädäntö.

Sosiaali- ja terveysministeriön (STM) vuonna 1995 laatiman tietoteknologian hyödyntämisstrategian tavoitteena oli parantaa sosiaali- ja terveyspalvelujen saatavuutta, laatua ja tehokkuutta. Strategian uutena ajatuksena esitettiin asiakkaan omatoimisuuden ja osallistumisen sekä terveystietoisuuden lisäämistä. [16, s. 9]. Vuonna 2006 Eduskunnan Tulevaisuusvaliokunnan kannanotto vuoden 2015 terveydenhuoltoon linjasi:

”Jokaisella suomalaisella tulee olla henkilökohtainen sähköinen terveystietokanta, jonka hän tai hänen edushenkilönsä voi avata ainakin mistä tahansa Suomessa. Sähköinen terveystietokanta on keskeinen lähtökohta tietotekniikan tehokkaalle käytölle terveydenhuollossa. Sähköisen terveystietokannan edelleen kehittämistä kohti aidosti älykäs järjestelmä on jatkettava.” [17, s. 3].

Laki sosiaali- ja terveydenhuollon asiakastietojen sähköisestä käsittelystä [18] määritteli monia strategialinjauksia tukevia yksityiskohtia liittyen muun muassa tietojen käytettävyyteen, käyttäjien tunnistamiseen, omien terveystietojen katseluoikeuteen ja valtakunnallisiin tietojärjestelmäpalveluihin. Terveydenhuollon palvelujen antajien tuli lain mukaan liittyä valtakunnallisten tietojärjestelmäpalveluiden käyttäjiksi neljän vuoden kuluessa. Laki sähköisestä lääkemääräyksestä [19] määritteli valtakunnallisen reseptikeskuksen, jossa potilaan lääkemääräykset talletetaan sähköisesti.

3.2 Kansallinen terveystietokanta

STM aloitti terveydenhuollon kansallisen tietojärjestelmäarkkitehtuurin toteutuksen suunnittelun jo sitä koskevan lainsäädännön valmisteluaikana. Kansallinen terveystietokanta (KanTa) on yhteinen nimitys terveydenhuollon valtakunnallisille tietojärjestelmäpalveluille, joita ovat sähköinen resepti (eResepti), kansallinen lääketietokanta, sähköinen potilastiedon arkisto (eArkisto) ja kansalaisten omien resepti- ja potilastietojen katseluverkossa (eKatselu). Käytännön työtä koordinoi Kansaneläkelaitos. Ensimmäinen käyttönotettava järjestelmän osa on sähköinen resepti. Samaan aikaan sähköisen reseptin kanssa otetaan käyttöön kansalaisten potilastietojen katselupalvelu sähköisen reseptin tietojen osalta [20]. Alkuperäisen lainsäädännön mukaan kaikkien potilastietojen pitäisi olla eArkistossa 1.4.2011 mennessä, mutta siirtymäaikaa ollaan muuttamassa STM:n toimesta [21].

Kansallinen terveystietokanta määrittelee järjestelmän perusarkkitehtuurin ja toteuttaa tiedon arkistointiin ja hallinnointiin liittyvät toiminnallisuudet. Terveydenhuollon palveluiden tuottajat liittyvät järjestelmiin nykyisten ohjelmistojensa avulla. Käytännössä tämä tarkoittaa, että kaikki käytössä olevat potilaskertomusjärjestelmät täytyy sovittaa yhteen KanTa-palvelun kanssa. [20]. Vuonna 2007 tehdyn kartoituksen [16] mukaan Suomen erikoissairaanhoidon 21 sairaanhoitopiirissä sähköisen potilaskertomuksen tuotemerkeinä oli käytössä kymmenessä sairaanhoitopiirissä Effica, Esko ja Miranda neljässä sekä Healthnet, Abilita/Medix ja Musti+WebkertSeniori kukin yhdessä sairaanhoitopiirissä. Kainuussa oli lisäksi Effican rinnalla käytössä Pegasos. Terveystietokannissa käytetyt tuotemerkit olivat saman tutkimuksen mukaan (suluissa prosentuaalinen osuus): Effica (52,2%), Pegasos (31,6%), Mediatri (12,5%), Finstar (3,4%) ja Abilita (0,3 %). Potilaskertomusjärjestelmien lisäksi sähköinen resepti edellyttää apteekkijärjestelmien integroimista palveluun.

Erilaisten järjestelmien yhteensovittaminen on ollut vaikeaa. Järjestelmien yhteensovittamisen järjestämiseksi samaa kertomusjärjestelmää käyttävät sairaanhoitopiirit ja

kunnat perustivat kuusi tilaajaklusteria, joiden tehtävä oli kehittää järjestelmiä yhdessä toimittajan kanssa vastaamaan kansallisten vaatimusten tarpeita [20].

3.3 Tietojärjestelmähankkeiden onnistuminen

Tietojärjestelmien uskotaan parantavan toiminnan laatua ja alentavan kustannuksia. Tuoreen Yhdysvalloissa tehdyn tutkimuksen [22] mukaan sairaalatietojärjestelmien hyödyt eivät olekaan selkeitä. Tutkimus totesi tietojärjestelmien nostaneen mitattuja laatutekijöitä vain hieman, ja hallinnon tehokkuuteen ja kokonaiskustannuksiin tietojärjestelmien käyttöönotolla ei ollut mitään vaikutusta. Tutkimustulosta ei voi yleistää omaan kansalliseen tilanteeseemme, mutta se viestii selvästi, että tietojärjestelmien käyttöönotto ei automaattisesti tuo toivottuja hyötyjä.

Terveystieteiden tietojärjestelmien kehitystyötä on arvosteltu Suomessa voimakkaasti. Hankkeita on kritisoitu kokonaisuuden hallinnan puutteesta ja moitittu kehitettävälle yrityksille annettujen vaatimusten vajavaisuudesta. Kommenttien mukaan ohjelmat on otettu usein käyttöön keskeneräisinä, lääkärit eivät ole olleet niitä kehittämässä ja sähköisen potilaskertomuksen moitittiin vievän runsaasti työaika. [23; 24; 25]. Useasti on mainittu ongelmaksi juuri käyttäjien liian vähäinen kuuleminen kehitystyössä. Tämä on erityisesti resurssiongelma. Käyttäjien toivotaan osallistuvan järjestelmien kehitystyöhön, mutta pääosin tämä tapahtuu oman työn ohella [26]. Järjestelmien käytön yleistyminen myötä sähköisestä potilastietojärjestelmästä on tullut kiinteä osa potilaan hoitoprosessia. Tämän seurauksena on enenevässä määrin alettu kiinnittää huomiota keskenkäisten järjestelmien aiheuttamiin potilasturvallisuusriskeihin [27].

Windblad et al. [16] toteavat tutkimuksessaan ” Suomessa informaatioteknologian käyttöönottoa on kritisoitu siitä, että asetettuja tavoitteita ei ole saavutettu. Terveystieteiden osalta vuosia 2003, 2005 ja 2007 koskevat selvityksemme osoittaa kuitenkin toista”. Tehtyjen tutkimusten ja käyttäjien esittämän kritiikin välillä on ristiriita tai sitten tietojärjestelmille asetetut ja täyttyneet tavoitteet eivät ole sisältäneet niitä asioita, jotka kritiikkiä herättävät. Tilanteen selvittämiseksi Lääkäriliitto, Terveystieteiden ja hyvinvoinnin laitos ja Aalto-yliopisto päättivät tehdä tähän mennessä laajimman potilastietojärjestelmien käytettävyysselvityksen [28]. Tulokset julkaistaan kevään 2010 aikana.

Tietojärjestelmähankkeiden projektihallintaan liittyvänä ongelmana, edellä mainitun puutteellisen vaatimushallinnan lisäksi, on tyypillisesti ollut hankkeiden viivästyneet aikataulut. KanTa-arkiston pilottikokeilu sähköisen reseptin osalta piti alun perin alkaa syksyllä 2008 [20]. Useiden viivästysten jälkeen sen pitäisi käynnistyä keväällä 2010.

Tammikuussa 2010 järjestettiin katselmointi kuudelle potilastietojärjestelmäklusterille (Effica, Pegasos, Miranda, ESKO, Abilita ja Mediatri) [29]. Tarkoituksena oli selvittää järjestelmien valmiudet eResepti ja eArkisto palveluihin liittymiseen. Lainsäädännössä esitettyä aikataulua kansallisiin palveluihin liittymisestä 1.4.2011 mennessä pidettiin epärealistisena. eReseptin käyttöönotot Effica- ja Pegasos-järjestelmien osalta voivat alkaa vuoden 2010 aikana. Muiden klustereiden käyttöönotto arvioitiin tapahtu-

van 2011-2012. eArkiston osalta on toteutuksessa nyt toimintoja, joilla saadaan omia asiakirjoja muodostettua ja tallennettua kansalliseen arkistoon. Joissain klustereissa oman järjestelmän asiakirjojen vienti- ja hakutoimintojen tavoiteaikatauluksi on asetettu vuoden 2012 loppu tai vuosi 2013. Realistinen aikataulu kaikkien klustereiden osalta omien asiakirjojen muodostamiselle, viemiselle ja hakemiselle on 2013-2014. ” Muiden organisaatioiden asiakirjojen hakujen osalta aikatauluja ei vielä ole täsmällisesti suunniteltu. Täsmällisen aikataulusuunnittelun esteenä ovat tällä hetkellä puutteelliset määritellyt tämän toiminnallisuuden toteutukselle.” [29].

KanTa-arkiston kokonaisarkkitehtuurin määrittely [30] valmistui vuonna 2006 ja sitä täydentävät osajärjestelmien määritelmät seuraavana vuonna. Nyt kolme vuotta myöhemmin puutteelliset määrittelyt estävät aikataulusuunnittelun organisaatioiden väliselle asiakirjahaulle. Tämä on tyypillinen esimerkki vesiputous-tyyppisen elinkaarimallin ongelmasta (luku 2.1.1). Monimutkaisissa ohjelmistohankkeissa on lähes aina mahdotonta tehdä järjestelmämäärittelyä riittävällä tarkkuudella projektin alussa. Käytännössä KanTa-hanke etenee vaiheittain inkrementaalisesti, mutta nyt läpikäydyn materiaalin pohjalta ei voida päätellä millainen elinkaarimalli KanTa-hankeelle on määritetty. Järjestelmällisesti toteutetun inkrementaalimallin mukaan seuraava päävaihe ”organisaatioiden välinen asiakirjahaku”-inkrementille olisi määrittelyjen valmistuminen. Se on perusedellytys täsmällisemmälle vaiheen aikataulusuunnittelulle.

Myös paikalliset potilastietojärjestelmähankkeet ovat kärsineet suurista aikatauluongelmista. Tämän työn loppuosassa arvioidaan Etelä-Pohjanmaan Sairaanhoidopiirin sähköisen sairaskertomusprojektin (SÄHKE) vaiheita. SÄHKE-projektin Effica-potilaskertomusjärjestelmän hankintasopimus tehtiin keväällä 2002. Sopimuksessa projektin päättymisajaksi asetettiin 30.6.2004. Projekti päättyi vuoden 2009 lopussa – tosin sisällöltään huomattavasti laajempaan kuin alkuperäisessä sopimuksessa oli määritetty.

4 Etelä-Pohjanmaan sairaanhoitopiirin sähköisen potilaskertomusjärjestelmäprojektin yhteenve-

to

Etelä-Pohjanmaan sairaanhoitopiirin (EPSHP) johtoryhmä asetti vuonna 1998 sairauskertomustyöryhmän selvittämään kirjoituskoneiden korvaamista tietokoneilla. Tämän selvityksen perusteella koottiin projektiryhmä selvittämään sähköisen sairauskertomuksen hankkimista sairaanhoitopiiriin. Projektiryhmän ehdotuksesta perustettiin projekti, jonka tavoitteena oli selvittää sähköisen sairauskertomuksen hankintaan ja käyttöönottoon liittyvää problematiikkaa, selvittää eri ohjelmistojen nykytilaa, esittää ehdotus sähköisen sairauskertomuksen pohjaksi sekä olla mukana koekäytössä ja käyttöönoton toteuttamisessa. Projektisuunnitelman hyväksyi sairaanhoitopiirin johtoryhmä 26.10.1999 ja hallitus 13.12.1999 [31]. Projektiryhmässä olivat mukana lääkärin, hoitohenkilökunnan ja tietoteknisen henkilökunnan edustajat.

Selvitystyön käynnistämisen ja sähköisen sairaskertomuksen käyttöönototarpeen taustalla oli yleinen yhteiskunnallinen kehitys tietotekniikan hyödyntämiseksi (katso luku 3.1) sekä tarve parantaa ja tehostaa tietojen saatavuutta.

4.1 Tarvekartoitus ja selvitys järjestelmätoimittajista

Sähköisen sairaskertomusprojektin (SÄHKE) alkuvaiheessa (2000-2001) perehdyttiin yksiköiden tarpeisiin, määriteltiin järjestelmän alustavat vaatimukset ja kartoitettiin sairaanhoitopiirin tekninen sekä taidollinen valmius sähköisen järjestelmän käyttöönotolle. [31].

Projektissa selvitettiin sairauskertomusohjelmistojen valmiusaste ja suurimpien toimittajien ohjelmistojen kehityssuunnitelmat. Projektiryhmä perehtyi terveydenhuollon valtakunnallisiin toiminnan ohjauksen ja tietotekniikan kehityshankkeisiin (Satakunnan Makropilotti) ja muihin potilastietojärjestelmien kehittämis- ja käyttöönottohankkeisiin (Primus, Turun terveystoimi ja Eppu, Erikoissairaanhoidon ja perusterveydenhuollon potilastietojärjestelmien uudistaminen sairaanhoitopiirien yhteistyönä).

Etelä-Pohjanmaan sairaanhoitopiirin (EPSHP) yksiköiden toimintaprosessit kartoitettiin ja näiden perusteella määriteltiin tulevan järjestelmän tarpeita. Sähköisen järjestelmän tuli korvata muun muassa käytössä olleet noin 270 sairaanhoidollista ja noin 100 potilashallinnollista paperilomaketta. Prosessikartoitus toi hyvin esille toimintaan liittyvät yksiköt sairaalan sisällä ja ulkopuolella. Näiden kautta määrittyivät luotavan järjes-

telmän rajapinnat ja tarvittavat yhteydet. Tehdyt prosessikuvaukset loivat myös hyvän apuvälineen tulevaisuuden prosessikehitykselle. Prosessikuvausten laadinnan yhteydessä annettiin toimintayksiköille informaatiota sairaskertomusprojektista. Samoin selkeytettiin yhteistä käsitystä osastojen toimintaprosessista ja heräsi myös keskustelua eri yksiköiden käytäntöjen yhtenäistämistä. [31].

Tarvittavaa sähköistä järjestelmää ei ollut keskussairaالاتasoisena käytössä Suomessa, joten valmista ohjelmistoa ei ollut suoraan hankittavissa. Saatavilla olevat ohjelmistot luokiteltiin seuraavasti:

- 1) Erillisjärjestelmät - pienekköihin erillisiin sovelluksiin sopivat tietokannat
- 2) Laajemmat tietokannat, joissa ei ollut kuitenkaan integroituna kaikki potilastieto
- 3) Digitaaliset arkistot
- 4) Hallinnolliset järjestelmät
- 5) Digitaaliset potilastietojärjestelmät. [31]

Järjestelmäselvityksen johtopäätöksenä todettiin, että määriteltävä järjestelmä tulee kytkemään yhteen usean erillisen ohjelmiston, joista osa saattoi olla jo selvityshetkellä käytössä. Näiden ohjelmistojen välisen tiedon kulun onnistuminen olisi perusedellytys toimivalle järjestelmälle. Ratkaisu tulee vaatimaan laajaa räätälöintiä, mikä tulee huomioida jo tarjouspyyntövaiheessa. Nähtiin tarve yhteistyösopimuksesta kokonaisvaltaisen ratkaisun tuottajan kanssa, ei pelkästään valmiin ohjelmiston hankinnasta.

Tehdyn selvitystyön pohjalta aloitettiin vaatimusmäärittelyn ja alustavan tarjouspyynnön valmistelutyö tavoitteena saada tehtyä hankintasopimus vuoden 2001 aikana.

4.2 Tarjouspyyntö

Julkinen tarjouspyyntö sähköisestä potilaskertomusjärjestelmästä tehtiin helmikuussa 2001 [32]. Tarjouspyyntö sisälsi sähköisen sairaskertomusjärjestelmän, potilashallinnon järjestelmän, digitaalisen arkistointijärjestelmän ja digitaalisen kuva-arkiston. Siinä määriteltiin haluttujen järjestelmien vaatimukset ja kuvaukset toiminnallisuudesta. Tarjousten jättöaika oli 30.3.2001 mennessä ja annettavien tarjousten tuli olla voimassa vuoden 2001 loppuun asti.

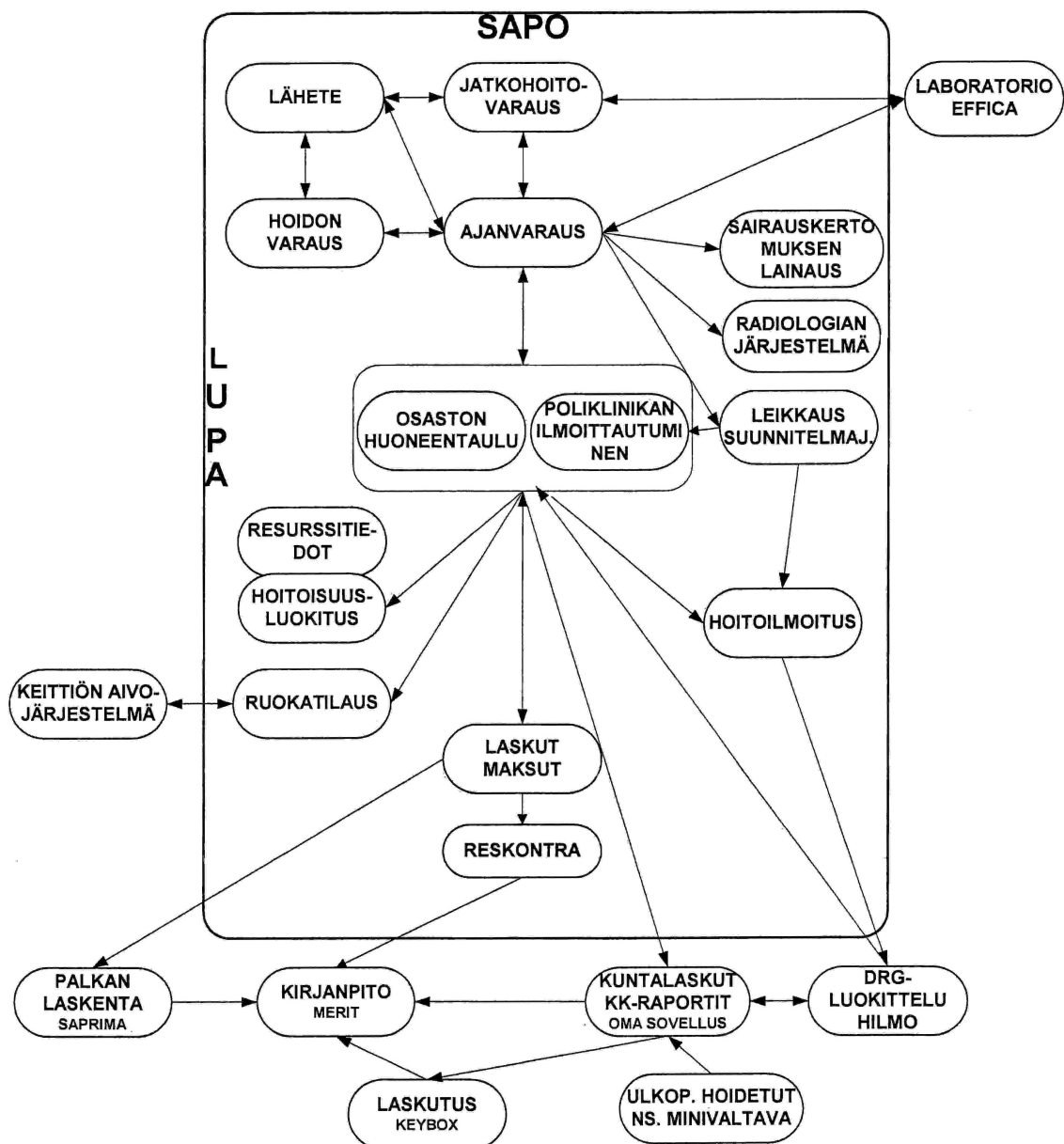
4.2.1 Sairaskertomusjärjestelmän vaatimukset

Järjestelmän tuli tukea toimintaprosessia lähetteen tai potilaan saapumisesta potilaan poistumiseen ja jatkohoidon järjestämiseen. Tallennetut tiedot tuli olla käytettävissä hoidon eri vaiheissa sekä tilastoinnissa. Tieto piti siirtyä hoitoketjun eri osiin ajantasaisena ja alkuperäisenä. Järjestelmän tuli korvata käytössä olevat sairaskertomuslomakkeet. Siinä piti olla yhteydet sairaalan ulkopuolisiin ja sisäisiin erillisjärjestelmiin. Ohjelmiston tuli olla käytettävyydeltään hyvä, helposti opittava ja selkeä. Kaikkiin potilaan tietoihin tuli päästä käsiksi sujuvasti. Vaatimuksissa kuvattiin tarkemmin käytettävyyden, tietoturvan ja tietosuojan ominaisuuksia. Myös sähköiseen potilaskertomukseen vaadittavat rakenneosat käytiin läpi. [32].

4.2.2 Potilashallinnon vaatimukset

Uuden potilashallinnon järjestelmän tuli korvata käytössä ollut potilashallinnon ATK-järjestelmä, SAPO. Kuva 7 näyttää SAPO-järjestelmän osat ja liittymät eri osajärjestelmien välillä.

Sapo- potilastietojärjestelmä ja liittymät muihin järjestelmiin



Kuva 7. Potilashallinto ja liittymät [32]

LUPA-rekisterissä ylläpidetään käyttäjien henkilökohtaisia käyttäjätunnuksia.

Lähetejärjestelmään kirjataan aina potilaan hoidon aloitus. Näin tehdään vaikka hoito alkaisi muuten kuin ulkoisen lähetteen aiheuttamana (sisäiset lähetteet, puhelinlähetteet, ilman lähetettä)

Ajanvarauksella määritellään vastaanotto- ja tutkimusajat viikonpäivittäin.

Jatkohoitovarauksella ylläpidetään seuraavan käynnin tietoja, jos potilaalle ei voida vielä antaa seuraavaa käyntiaikaa. Tarkan ajan sijasta jatkohoitotiedoissa on tavoiteaika.

Hoidonvarausjärjestelmässä ylläpidetään operatiivisille erikoisaloille osastohoitoon pääsyä odottavien potilaiden tietoja.

Poliklinikan ilmoittautuminen osuuden kautta hoidetaan potilaan ilmoittautumisen yhteydessä tarvittavat toimet muun muassa laskutustiedot, diagnoosit ja mahdolliset toimenpidetiedot sekä tarkistetaan potilaan henkilötiedot ajan tasalle.

Osaston huoneentaululla hallitaan osastopotilaiden sisäänkirjoitusta, tietojen ylläpitoa ja poistamista.

Hoitoilmoitus eli HILMO-tiedot lähetetään jokaisesta vuodeosastolla hoidetusta potilaasta valtakunnalliseen rekisteriin vuosittain. Ilmoituksen lähettämistä seurataan potilaskohtaisesti tällä osuudella.

Potilaan varaukset-näytöllä näkyvät avoimet lähetteet ja voimassaolevat ajan-, jatko- ja hoidonvaraukset.

Potilaan hoitajaksot-näytöllä näkyvät kaikki käynnit ja hoitajaksot vähintään edelliseltä vuodelta.

Potilaslaskutus-osiota käytetään laskutukseen potilaan ilmoittautumisen tai osastolta poistumisen yhteydessä. Laskutus voidaan tehdä myös jälkikäteen

Reskontra kirjaa laskut ja sieltä voidaan tehdä potilaskohtaisia laskutilannekyselyitä. Suoritustiedot siirretään pankista ja välitetään kirjanpitoon. Karhulaskujen ja ulosotto-korttien tulostaminen tapahtuu Reskontra-osiossa.

Kuntalaskutus hoitaa laskutuksen kuntiin kahdesti kuukaudessa. SAPOsta tiedot siirretään päivittäin kuntalaskutukseen

Tilastointi ja raportointi mahdollistaa erilaisten toimintatilastojen laatimisen kuntalaskutusrekisteristä sairaalan sisäisiin ja ulkoisiin tarpeisiin.

Sairaanhoitopiirin henkilörekisteri sisältää sairaanhoitopiirin asukkaiden henkilötietojen lisäksi SAPO-järjestelmässä käsiteltyjen ulkokuntalaisten ja ulkomaalaisten tietoja.

Sairaskertomusasiakirjojen lainausten hallintajärjestelmä, SAILA, ylläpitää tietoja paperisten sairaskertomusten ja röntgenkuvien lainaustilanteesta.

Leikkausohjelman suunnittelujärjestelmä, LESU, tuottaa hoidon- ja ajanvaraustiedoista alustavat leikkauslistat ja järjestelmän kautta niitä ylläpidetään ja välitetään muiden yksiköiden nähtäväksi.

Radiologian järjestelmä, X-RAY, kerää radiologian yksikön suoritettuja toiminnan seuranta ja laskutusta varten. Järjestelmän kautta hoidetaan myös pyyntöjen ja lähetteiden kulku hoitoyksiköiden ja radiologian yksikön välillä.

Ateriailausjärjestelmä, SAFKA, ylläpitää potilaskohtaisia ruokailuun liittyviä tietoja. Ateria- ja tuotetilaukset välitetään AIVO-järjestelmälle sovittuina aikoina. [32]

4.2.3 Digitaalisen arkiston ja kuva-arkiston vaatimukset

Terveystietojenhuollossa syntyvä tieto tulee säilyttää säädösten mukaisesti. Tietojen tulee säilyä muuttumattomina ja kiistämättöminä. Tietojen tallennusmuodossa tulee sitoutua Suomessa hyväksytyihin standardeihin, jotta tietojen myöhempi käyttö eri tietojärjestelmien kautta olisi mahdollista. Digitaalisen arkistoinnin tulee täyttää pitkäaikaisarkistoinnin vaatimukset ja tietoturva on toteutettava lainsäädännön mukaisesti.

Digitaaliselle kuva-arkistolle määriteltiin erityisvaatimuksia liittyen kuvaformaattiin, siirtoverkkojen suorituskykyyn, pakkaukseen ja muihin kuvien tallentamisen piirteisiin. [32]

4.3 Toimittajavalinta

Määräaikaan mennessä annettiin 13 tarjousta, joista osa oli osajärjestelmä- ja kehitystyötarjouksia. Tarjouksen antajien kanssa käytiin tarkentavia keskusteluja kevään aikana ja tarjouksia vertailtiin tarjouspyynnön pohjalta muodostettujen kriteerien perusteella. Kolmelta ohjelmistotalolta TietoEnator, NovoGroup ja Medici Data pyydettiin vielä lisäselvityksiä syksyn 2001 aikana. Näiden jälkeen hankintatyöryhmä esitti, että selvityksiä jatketaan TietoEnatorin kanssa Effica-järjestelmästä. Potilashallinnon osuuden tarkastelun sekä muun hankinnan pohjaksi vaadittavan valmistelun vuoksi tarjouksiin pyydettiin vielä lisää aikaa huhtikuun loppuun 2002. Digitaalisen kuva-arkiston hankinta ehdotettiin siirrettäväksi myöhempään ajankohtaan. [33]

EPSHP:n hallitus hyväksyi TietoEnatorin tarjouksen Effica potilaskertomusjärjestelmästä huhtikuussa 2002. Alkuvuoden 2002 aikana valmisteltiin sairaanhoitopiiriin ja TietoEnator Oyj:n välinen hankintasopimus, joka määritteli tarkemmin yhteistyöalueet, toimintatavat ja sopimuskohteiden hinnoittelun. Hankintasopimus tehtiin 10.6.2002. Aikaisemmin valmistuneiden selvitysten johtopäätös siitä, että kyseessä on yksinkertainen ohjelmistohankinta laajempi tehtävä, näkyi myös hankintasopimuksessa. Sairaanhoitopiirille sähköisen potilaskertomuksen toimittajan kanssa yhteistyössä tuottava SÄHKE-projekti koostui kahdesta pääaktiviteetistä: toteutuksen tai määrittelyn osalta valmiiden järjestelmäosien käyttöönotto ja tunnistettujen lisätarpeiden kehitystyö. Projektin päättymisajaksi asetettiin hankintasopimuksessa 30.6.2004, jota voitaisiin yhteisellä päätöksellä muuttaa määrittely- ja suunnitteluvaiheen jälkeen. [34].

4.4 Projektin ohjaus ja organisaatio

1-2 kertaa vuodessa kokoontuva SÄHKE-projektin ohjausryhmä koostui sairaanhoitopiiriin johdon, työntekijöiden ja projektiryhmän edustajista. Johtoryhmä seurasi projektin etenemistä ja rahoitusta. Sähköisen potilaskertomusjärjestelmän selvitystyöhön perustettu projektiryhmä jatkoi projektin käytännön työn ohjaamista ja seuraamista kokoontuen virallisesti noin 8-9 kertaa vuodessa. Projektiryhmässä olivat edustettuina pääasiassa lääkärit, hoitohenkilökunta ja tietotekninen henkilökunta. Päätoimista projek-

tisihteeriä lukuun ottamatta projektiryhmän jäsenet osallistuivat projektiin varsinaisen työnsä ohessa. Projektiryhmän edustajat ja TietoEnator Oyj:n sairaalasektorin johdon edustajat kokoontuivat 1-2 kertaa vuodessa hankintasopimuksen tarkastelukokoukseen, jossa seurattiin sopimuksen toteutumista.

Järjestelmän käyttöönoton valmisteluihin perustettiin kuusi käynnistämistyöryhmää. Myös hankintasopimuksessa sovituille kuudelle kehittämisalueelle muodostettiin kullekin oma kehittämissuuryhmä. Työryhmät vastasivat oman alueensa valmisteluista, selvityksistä ja määrittelyistä yhteistyössä toimittajan kanssa

4.4.1 Kehityssuuryhmät

Effica-potilaskertomusjärjestelmän hankinta-ajankohdan toiminnallisuuden lisäksi oli paljon lisätarpeita joiden kehittämisen koordinoimiseksi perustettiin kehittämissuuryhmät:

- **Potilashallinnon kehittämissuuryhmä** kehitti Effican potilashallintoa sisältämään mm. jatkohoitovarausjärjestelmän, ajanvarausjärjestelmän muutoksia ja täydennyksiä, osastohallintaa, laskutusta, lähetteen ja tilausten käsittelyä, laboratorioyhteyksiin liittyviä yksityiskohtia sekä korvaavia ominaisuuksia SAPO- ja XRAY-järjestelmään.
- **Sairauskertomuksen kehittämissuuryhmä** teki sairauskertomuksen strukturointia, potilaan lääkitykseen liittyvien toimintojen ja lääkkeenjakoiltojen kehittämistä, sairauskertomuksen käytettävyyden kehittämistä esim. yhteenvedojen osalta.
- **Leikkaustoiminnan kehittämissuuryhmä** kehitti Effica-järjestelmää vastaamaan LESU:n toimintoja
- **Ateriatilauksen kehittämissuuryhmä** kehitti Effica- järjestelmä vastaamaan SAFKA:n toimintoja
- **Hoitotaulukon kehittämissuuryhmä** määritteli Effica- järjestelmään erikoissairaanhoidon vuodeosastotyötä palvelevan hoitotaulukkonäkymän alatoimintoinen sisältäen muun muassa yhteydet lääkitykseen ja nestetasapainon laskeamiseen.
- **Hoitosuunnitelman kehittämissuuryhmä** määritteli Effica - järjestelmään yhtenäisen hoitosuunnitelmalomakkeiston , joka palvelisi kaikkia erikoisaloja mahdollisine muunnelmineen. Hoitosuunnitelmaprojektiin sisältyi myös hoitoisuuden arvioinnin sisällyttäminen hoitosuunnitelmaprojektin osajärjestelmäksi. [35].

4.4.2 Käynnistämistyöryhmät

Hankintahetkellä TietoEnatorin Effica-järjestelmä oli käytössä useissa terveyskeskuksissa ja muutamissa keskus- ja aluesairaaloissa, mutta ei SÄHKE-projektin vaatimassa laajuudessa [33]. Tämä mahdollisti kuitenkin järjestelmän asteittaiseen käyttöönottoon

valmistautumisen välittömästi. Valmiiden osien käyttöönoton jälkeen jatkuisi kehitystyöryhmien ohjaamana tuotettujen uusien järjestelmäkomponenttien käyttöönotto.

Käyttöönoton valmisteluun nimettiin kuusi käynnistämistyöryhmää:

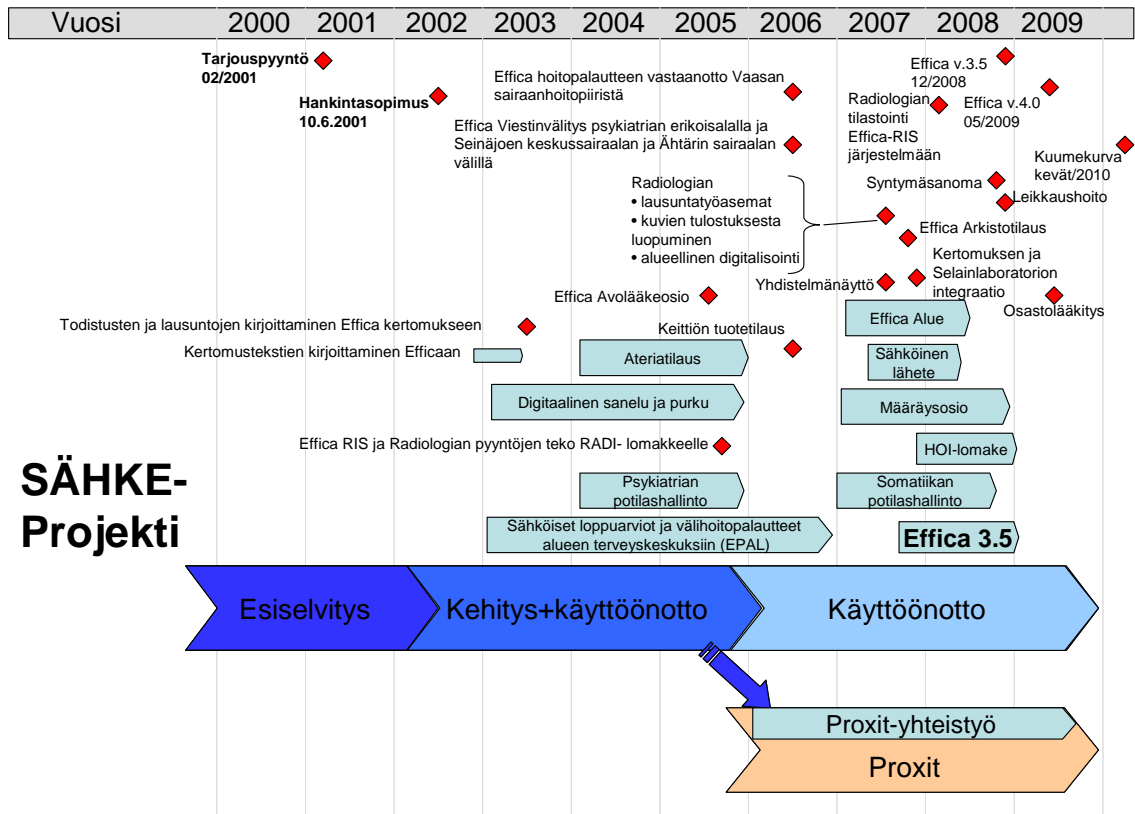
- **Työnkulutustyöryhmä** päätti ja kokosi koko aloitusvaiheen ohjeistuksen
- **Koodistotyöryhmä** antoi ohjeet järjestelmässä noudatettavista koodeista esimerkiksi yksiköiden nimilyhenteistä.
- **Käyttöoikeustyöryhmä** suunnitteli miten oikeudet käyttää ja liikkua järjestelmässä ohjeistetaan. Se suunnitteli salassapito- ja tietoturva-asiat.
- **Potilaskertomustyöryhmä** suunnitteli käyttöönotettavien sairauskertomuslehtien muodon, nimet ja toimintatavat lehdillä.
- **Tekninen työryhmä** suunnitteli järjestelmän käyttöönottoon liittyvät tekniset yksityiskohdat.
- **Koulutustyöryhmä** suunnitteli käyttöönottokoulutuksen yksityiskohdat laaditun ohjeistuksen pohjalta. [35].

4.5 Järjestelmän osa-alueiden kehitystyö ja käyttöönotto

Effica-potilaskertomusjärjestelmän käyttöönoton suunnittelu EPSHP:ssä aloitettiin heti hankintasopimuksen tekemisen jälkeen. Käyttöönotto tapahtuisi suunnitelmien mukaan monissa vaiheissa sekä käyttöönottoyö määrän jaksottamiseksi että uusien ominaisuuksien kehitystyöaikataulujen rajoittamana. Jaksottaminen pienensi riskiä, että uuden järjestelmän käyttöönotto häiritsisi liikaa yksiköiden normaaleja tehtäviä.

Järjestelmän osa-alueista tekstinkäsittelyn, palautejärjestelmän sekä digitaalisen sanelun ja purun käyttöönotot pystyttiin aloittamaan kehityshankkeista riippumattomasti. Muiden osa-alueiden käyttöönotot käynnistyivät kun riippuvuudet eri kehityshankkeisiin ne mahdollistivat.

Hankintasopimuksessa asetettiin projektin alkuperäiseksi tavoiteajaksi vuoden 2004 puoliväli. Monista syistä johtuen SÄHKE-projekti kesti moninkertaisesti pidempään kuin alkuperäisessä sopimuksessa oli arvioitu. Merkittävä syy projektin alkuvaiheen viiveille oli jo hankintahetkellä tunnetut järjestelmän puutteet, joiden täydentämiseksi perustettiin kehitystyöryhmät. Kehityshankkeiden ensimmäinen vaihe oli puuttuvien alueiden tarkempi vaatimusmäärittely. Sopimushetkellä puuttuvat vaatimusmäärittelyt ja niiden tarkempi analyysi teki mahdolliseksi tietoon pohjautuvan aikataulusuunnittelun. Sopimuksessa todettiin, että aikataulua voidaan yhteisellä sopimuksella muuttaa. Projekti oli alkuvaiheessa selkeästi enemmän sisältö- kuin aikatauluohjattu. Kuva 8 havainnollistaa SÄHKE-projektin päävaiheet ja tapahtumat.



Kuva 8. SÄHKE-projektin vaiheet, päätapahtumat ja käyttöönotot

Seuraavissa kappaleissa käydään SÄHKE-projektin osa-alueet läpi yleisellä tasolla. Tarkempi arviointi projektihallinnan kannalta tehdään luvussa 5.

4.5.1 Tekstinkäsittely

Syksyllä 2002 alkoi ensimmäisen käyttöönoton, tekstinkäsittelyn, pilotointi naistentautien tulostusyksikössä sekä siihen liittyvissä palveluyksiköissä (leikkaussali, päiväkirurgia) naistentautien erikoisalalehden osalta. Varsinainen Effican tekstinkäsittelyyn siirtyminen tapahtui vaiheittain vuoden 2003 alussa. Huhtikuun alkuun mennessä tekstinkäsittelyn käyttöönotto oli tapahtunut kaikissa yksiköissä. Aikaisemmin tekstinkäsittely oli sairaanhoitopiirissä tehty Word-tekstinkäsittelyohjelmalla ja tallennus ainoastaan paperille. Effican tekstinkäsittelyominaisuudet eivät vastanneet kaikilta osin tekstinkäsittelyohjelman käyttöön tottuneiden käyttäjien odotuksia ja TietoEnatorille raportoitiin paljon korjaus- ja parannusehdotuksia. Heti käyttöönotossa valmiiden lomakkeiden lisäksi vuoden 2003 aikana tuli lukuisia uusia lomakkeita siten, että vuoden loppuun mennessä huomattava osa kaikista erikoisala- ja erikoislomaketeksteistä kirjoitettiin ja tallennettiin Effican. Ulkopuolelle jäi edelleen paljon kynällä kirjoitettavia potilaspapereita, laajimpana kokonaisuutena kaikki varsinaiset hoitosuunnitelmalehdet ja useat yksiköiden käyttämät lisälomakkeet. [36].

Paperilomakkeiden kanssa käytettyjä lisämerkintäkäytäntöjä piti muuttaa sähköisen sairaskertomuksen tukemaan muotoon. Tällaisia olivat erilaiset papereihin liitetyt tarrat

ja leimat jotka olivat oleellinen osa toimintakäytäntöjä (esimerkiksi jonoon asettamismerkinnät ja ”kopio lähetetty”-leima) [36].

4.5.2 Palautejärjestelmä

Effican toinen käyttöönottoalue oli lähete/palautejärjestelmän palauteosuus. Tämän avulla pystytään lähettämään erikoisalalehdille kirjoitettu suoramuuotoinen tekstitieto jatkohoitopaikkaan sähköisesti ja paperikopioiden lähettäminen vähentyy. Alueen kaikissa terveyskeskuksissa sähköinen sairaskertomus oli jo käytössä ja sähköisen palautejärjestelmän käyttöönottoa oli odotettu jo pitkään. Järjestelmän käyttöönotto edellytti tietoliikenneyhteyksien salaamista sairaanhoitopiiriin ja terveyskeskusten välillä.

Järjestelmän käyttöönotto aloitettiin terveyskeskusten kanssa, joissa oli käytössä Effica-järjestelmä. Vuoden 2003 aikana palautejärjestelmän käyttöönotto alkoi Seinäjoen terveyskeskuskuntayhtymän neljän kunnan kanssa. Käyttöönotto laajeni vuoden 2004 aikana siten, että vuoden lopussa oli mukana 13 kuntaa. Alueen kaikki Efficaa käyttävät kunnat saatiin 2005 palautejärjestelmän piiriin Karijokea lukuun ottamatta. Myös Pegasos-järjestelmää käyttävät alueen kunnat saatiin mukaan vuoden 2005 aikana. Vuoden 2006 alussa sähköinen palautejärjestelmä laajeni myös Mediatri potilastietojärjestelmää käyttäviin kuntiin. [37].

Sairaanhoitopiiriin saapuvien läheteiden vastaanotto sähköisessä muodossa ei alkuvaiheessa vielä ollut mahdollista. Se edellytti potilashallinto-osuuden keskeisten kehitysalueiden valmistumista.

4.5.3 Digitaalinen sanelu ja purku

Kertomuksen saneluohjelmalla voidaan sanella tekstejä potilaan kertomuksen eri lomakkeille, jolloin ne ovat välittömästi talletuksen jälkeen sekä kuunneltavissa että purettavissa. Tämän osuuden pilotointi tapahtui marraskuussa 2003 ja pääasiallinen käyttöönotto vuoden 2004 aikana. Digitaalisen sanelun käyttöönotto jossakin yksikössä ei vaikuttanut muihin yksiköihin mitenkään muuten kuin, että Efficassa näkyi ilmoituksena tieto potilaan purkamattomista digitaalisista saneluista avattaessa potilaan kertomusta, mikäli saneluita muistissa oli. Purkamattomista saneluista voitiin kuunnella tarvittaessa mahdollisen uuden hoitotilanteen alkaessa. [36].

4.5.4 Ateriatilaus

Ateriatilaus puuttui hankintavaiheessa kokonaan Efficasta. Hankintahetkellä EPSHP:ssä käytössä olleen potilashallinnon järjestelmän (SAPO) osana oli ateriatilausjärjestelmä Safka, jonka Effica-ateriatilausjärjestelmä korvasi. Uudessakin järjestelmässä ravintokeskuksen puolella oleva jakelujärjestelmänä säilyi samana (Aivo). Järjestelmässä ylläpidetään potilaskohtaisia ruokavalioon ja ruokailuun liittyviä tietoja. Niitä käytetään

osastolla olevien ja sinne tulevien potilaiden ateriatilauksia tehtäessä. Kuva 9 on näytönäkymä ateriatilausjärjestelmästä. [32].

Hakutekijät

Osasto: vos1 Vuodeosasto 1 Tilausyyppi: Keskitetty
 Ateriatyyppi: L Lounas Toimituspäivä: 21.11.2003
 Sulkuaika: 21.11.2003 11:00

Ateriatilaukset

T	Hne	Vde	Asiakas	Ruokavalio	Rakenne	Kalori	Yhd	Osastohuom.	Jaku
	H01	1	Testi Johanna	5 Aikuisdabel	1 Norm	2 180		AVUST	
	H01	2	Franti Riikka	1 Perusuokav	1 Norm	2 180			Kert
	H01	3	Vasara Virpi Hannele	1 Perusuokav	1 Norm	1 150			
	H02	1	Hakoma Hanna Elina	1 Perusuokav	1 Norm	2 180			
	H03	2	Väänänen Lauri	1 Perusuokav	1 Norm	2 180			
	H03	3	Laine Yrjö	1 Perusuokav	1 Norm	2 180			
	H04	1	Ekaerä Elli Testi	1 Perusuokav	1 Norm	2 180			
	H04	3	Haanpää Frida	1 Perusuokav	1 Norm	2 180			
	H05	1	Lagerlöf Selma	1 Perusuokav	1 Norm	1 150			
	H05	3	Sairastaa salassa	1 Perusuokav	1 Norm	2 180			
	H07	1	Uimo Unto	1 Perusuokav	1 Norm	2 180			Kert
	H07	2	Sairastaa salassa	1 Perusuokav	1 Norm	2 180			
	H07	3	Sairastaa salassa	1 Perusuokav	1 Norm	2 180		Huomio2	
	H08	1	Testattava Testi Taavi	1 Perusuokav	1 Norm	3 220			
	H08	2	Väänänen Kristiina	1 Perusuokav	3 Pehm	2 180			
	H10	2	Tommi Mäkinen	1 Perusuokav	1 Norm	2 180			

Yhdistelmät: +Laktoositon, +Vähälaktoosinen, +Westemäinen, +RASVA, -Maksa, -Sipuli, +Omena, -kala

Osastohuomautus: Syötettäessä osittain avustettava

Jakelu huomautus: Kertakäyttöastiat ruokailuun, Ruokailuun huomioita kertakäyttöastiat ja servetit, Testataan yleis- eli jakeluhuomatusta, sfks kfaksk: sf ksjaksa jkkk jkfska, kjksf kj ksj fksak kjkjksad f jksaf, ksjf kj sajka jk jkajka kjkskf, jkskfjksks ajfksajksak jksajksajkskf

Kuva 9. EfficA Ateriatilaus

SÄHKE-projektin kehittämisosioista ateriatilausjärjestelmä valmistui ensimmäisenä. Käyttöönotto tapahtui vaiheittain vuoden 2004 ja alkuvuoden 2005 aikana. Ateriatilauksen viimeinen osio, tuotetilauks, otettiin tuotantokäyttöön maaliskuussa 2006. Ateriatilausjärjestelmä Efficassa on osoittautunut varsin toimivaksi ja käytön myötä tilausten teko on yhdenmukaistunut eri osastoilla. Käyttäjät ovat olleet tyytyväisiä tähän järjestelmään osaan [38].

4.5.5 Potilashallinto

Kehittämistyö pohjautui EfficA-potilashallintoon, joka ei sellaisenaan palvellut riittävän hyvin erikoissairaanhoidon tarpeita. Keskeisimpiä kehittämisalueita olivat lähetekäsittelyyn, ajanvaraukseen, jatkohoitovarakseen, laskutukseen, arkistotilauksiin ja osastohallintaan liittyvät kokonaisuudet. EfficA potilashallinto on ollut tuotantokäytössä psykiatrian toiminta-alueella vuodesta 2005 lähtien, kokonaisuudessaan vuodesta 2006 lähtien. Vuoden 2007 aikana aloitettiin SAPO:n vaihtaminen EfficA potilashallintoon asteittain myös somatiikan puolella. [39].

Effica-potilashallinnon tuli korvata käytössä ollut SAPO-järjestelmä. Vanha järjestelmä oli toiminnassa koko kehitystyön ajan ja Effica-potilashallinnon vaihettaisten käyttöönottojen myötä uuden järjestelmän tietoja siirrettiin tarvittavin osin SAPO-järjestelmään. Täydellisesti SAPO-järjestelmästä luovuttiin vasta vuonna 2009.

Osastohallinta oli potilashallinnon ensimmäinen käyttöönottoalue. Se otettiin aluksi käyttöön psykiatrian tulosalueella marraskuussa 2004. Käyttöönotto sujui ongelmitta ja henkilökunta oli tyytyväinen uuteen järjestelmään. Käyttöönotto psykiatrian tulosalueella jatkui vuoden 2005 syksyllä avopuolen potilashallinnon käyttöönotolla. Vuoden 2005 väliraportin mukaan toimivuudessa oli puutteita ja erilaisten hoitokäyntien ja -jaksojen kirjaantumisen oikeellisuuden tarkistamien teetti käyttöönoton jälkeen runsaasti lisätyötä [40]. Muiden tulosalueiden tarvitsemien ominaisuuksien viivästyminen ja puutteet hidastivat potilashallinnon käyttöönoton laajenemista. Kuva 10 on näytönäkymä osastohallinnasta.

The screenshot shows the 'Osastohallinta' (Department Management) interface. The main area displays a grid of patient lists for rooms H217 through H226. Each room's list includes patient names, room numbers, and status indicators. A sidebar on the left contains navigation options like 'Ajankäyttö', 'Kutsu', and 'Leikkaushoitto'. At the bottom, there is a table for 'Tulevat' (Upcoming) patients and a 'Läsnäolotiedot' (Attendance Information) section.

Huonenumero	Nimi	Mistä	Erikoisala	Hoitomuoto	Tulopvm	Tuloai	Tulopvy	Ajanv.laji	Diagnosi	EML-luokitus	Kunta
080351-A020	Kielo Kaarina	Ajanvaraus	200	Otopedia	20.10.2008 12:00		MOLEMPIEN LC ... / 8				152
141158-SIL1	Sillanpää Testi S	Ajanvaraus	200	Otopedia	15.10.2008 07:00		kipeä ... / 8				743
151253-A010	Saukko Kalevi	Ajanvaraus	200	Otopedia	18.10.2008 07:00		RANNEKANAV... / Leik				005
220880-SIL2	Sillanpää Testi K	Ajanvaraus	200	Otopedia	22.10.2008 07:00		selkävivot ... / 8				743

Kuva 10. Osastohallinta

Lähetesiossa oli tavoitteena SAPO:n kaltainen lähetekäsittely. Sähköisen lähetteen pilotointi ja lähetemääräyksen pilotointi suoritettiin naistentaudeilla keväällä 2006. Ongelmia tuotti lähinnä se, että muu potilashallinto on vielä SAPO-järjestelmässä. Myös ihotautien yksikkö aloitti sähköisten lähetteen vastaanottamisen joulukuussa 2006. Sähköisten lähetteen vastaanotto laajennettiin 1.4 2007 alkaen kaikille somatiikan poliklinikoille. Sähköisten lähetteen vastaanotto laajennettiin 2007 loppuvuoden aikana myös yksityispuolen Suomen terveystalo Oyj:n Seinäjoen työterveyskeskuksen

läheteisiin. Läheteosion jatkokehitys siirtyi vuonna 2006 pääosin Proxit-projektin (katso lisätietoja luvussa 4.6) puitteissa tehtäväksi kehitystyöksi. Proxit hankeen myötä tuli lähetekäsittelyyn vaatimusmuutoksia hoidon saatavuuden seurantaan liittyen. [39]

Effica-ajanvaraukseen siirtymisen valmistelutyöt aloitettiin Ihotautien yksikössä loppusyksystä 2006 ja varsinainen tuotantokäyttö alkoi vuoden 2007 alusta. Siirtyminen Effica-ajanvaraukseen ja -tilastointiin oli mahdollista, koska ihotaudeilla ei ollut käytössä drg-luokitteluun pohjautuvaa kuntalaskutusta. Ajanvarauskäyntien osalta siirryttiin toukokuussa 2006 poimimaan Efficasta raakadata erilliseen kuntalaskutusohjelmaan, jolloin laskutus kuntiin voitiin suorittaa yhdellä laskulla. [37].

Laskutusosion tavoitteena oli varmistaa, että kunta- ja potilaslaskutus toimivat EPSHP:n haluamalla tavalla. Samalla kartoitetaan sisäisen laskutuksen tarpeet. Tarkasteltavana oli myös laskutuksen tekeminen erilliseksi sovellukseksi, kalliinhoidontasaus, käyttölaajuus, siirtymävaiheen tarpeet ja ostopalveluiden laskutus. Kuntalaskutus tehtiin omalla laskutusohjelmalla kevästä 2006 lähtien, jolla huomioitiin kalliinhoidon tasaus ja siirtymävaiheen tarpeet. [37].

Tilastoinnin kehitystä tehtiin 10 sairaanhoitopiirin asiakasryhmässä. TietoEnator rakensi uutta, enemmän “dataware house” –tyyppistä, tapaa koota tietoa. EPSHP:n kannalta tärkeintä oli saada kuvaus sovelluksen sisällöstä, jolloin kyseistä tietoa voitiin poimia omiin raportointijärjestelmiin. Alkuvaiheessa tilastointi hoidettiin Efficasta poimitun raakadatan perusteella. [37].

Arkistohallinta osuuden tavoitteena oli tuottaa uusi automatisoitu järjestelmä SAILA-järjestelmän korvaamiseksi. Effica arkistotilaus osio testattiin vuoden 2006 keväällä. Järjestelmässä oli vielä puutteita ja korjauksia tarvittin. Järjestelmän käyttöönotto tapahtui 18.9.2007, jolloin myös vanha SAILA järjestelmä poistui kokonaan käytöstä. SAPO ajanvarauksiin, osastosiirtoihin ja osastolta poistumisiin liittyvät arkistotilaukset siirtyivät päivittäin käynnistettävällä automaattijolla uuteen Effica arkistotilausjärjestelmään

Vanhojen potilaskertomustietojen skannaus aloitettiin vuonna 2007 arkistotoiminnan yhteydessä. Skannattava materiaali koostuu Efficaa edeltävistä kertomusteksteistä, kuvista, biosignaaleista ynnä muusta. Skannattua materiaalia pääsee katselemaan Effica järjestelmän kautta. [37; 39].

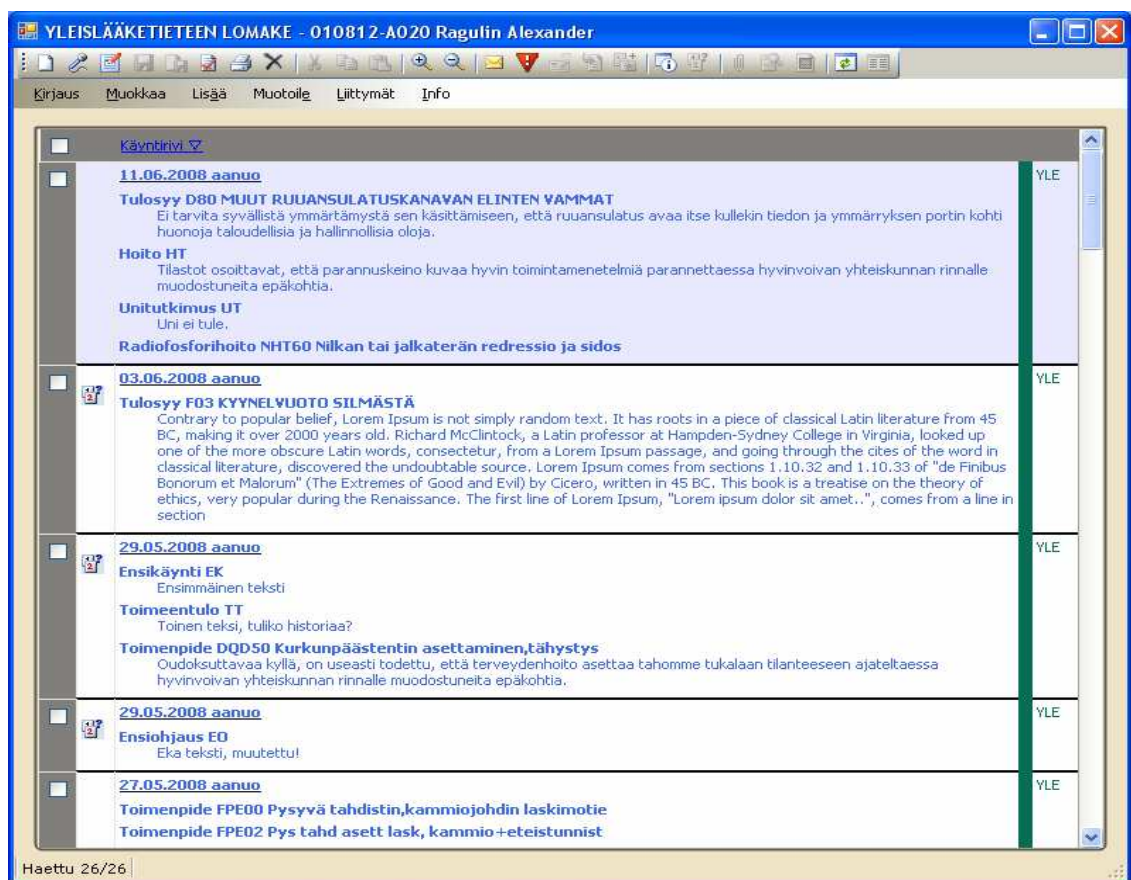
Jatkohoitovarausten tavoitteena oli tuottaa Efficasta puuttuva jatkohoitovaraus (kelluva moniajanvaraus) ja taata sen avulla toiminnallisuus huomioiden muun muassa SAPO:a vastaava hajautettu näytteenotto. Ajanvarauksen laajuus tuo myös lisäpiirteitä jatkohoitovarausten toiminnallisuutta kehitettäessä. [37]

4.5.6 Sairauskertomus

Sairauskertomuksen muoto oli alkuvaiheessa varsin perinteinen pohjautuen paperilomakkeiden ja erikoisalalehtien muotoiluun. Tavoitteena oli muuttaa muoto strukturoiduksi, jolloin käytettyjen kenttien otsikoiden avulla pystytään kertomuksesta hakemaan erilaisia näkymiä ja tulostamaan myös muunlaisia tulostuksia kuin mitä paperinen sairausker-

tomusjärjestelmä mahdollisti. Projektin alkuvaiheessa tärkeitä sairaskertomusryhmän kehityskohteita olivat myös lääkitysosion kehittäminen yhdenmukaiseksi valtakunnallisen sähköisten reseptien ja lääkemääräysten määrittelyjen kanssa sekä kuvan liittäminen osaksi kertomusta, erityisesti radiologian alueella. [38].

Strukturoitu kertomus oli edellytys monille sähköisen sairaskertomusjärjestelmän myötä tulleille uusille mahdollisuuksille. Sairaanhoidopiiri panosti sairauskertomuksen jatkokehitystyössä otsikointiin ja sitä kautta rakenteisen kertomuksen käyttöönottoon ja valtakunnallisten ydintietojen rakentumiseen. Rakenteinen kertomus otettiin tuotantokäyttöön keväällä 2005. Kuva 11 on esimerkki rakenteisen kertomuksen lomakkeenäymästä. [40].



Kuva 11. Yleislääketieteen lomake

Yhdistelmänäyttö on ohjelmiston keskeinen komponentti ja sen toiminnan edellytyksenä oli kertomuksen rakenteisuuden toteuttaminen. Yhdistelmänäyttö on kertomuksen hakutyökalu, jonka avulla käyttäjä voi hakea haluamiaan potilaskohtaisia tietoja järjestelmästä erilaisia hakutekijöitä käyttämällä. Efficia yhdistelmänäyttö otettiin virallisesti käyttöön keväällä 2007. [39].

Lääkitysosion kehityksen tarkoituksena oli saada tiedot potilaan lääkityksestä kattavasti ja nopeasti saataville. Sähköisen reseptin ja reseptikeskuksen valtakunnallisen

määrittelytyön seuraaminen oli olennaista lääkitysosion kehityssuunnitelmia laadittaessa. Vuoden 2005 aikana saatiin valmiiksi tuotantokäyttöön alueen ensimmäinen osuus, avolääkitys. Osastolääkitysosio oli huomattavasti haastavampi kehitysalue. Se vaati mm. integroinnin sairaalan apteekkijärjestelmään ja lääkkeen vaikuttavan aineen tai kauppanimen perusteella tehtävän haun sairaalan peruslääkevalikoimasta. Osastolääkitysosion käyttöönotto viivästyivät vuoteen 2009. [37; 41].

Radiologia on sairaalatoiminnan yksi keskeisemmistä palvelutoiminnoista. Se saatiin siirrettyä Effican piiriin joulukuussa 2005. Tällöin vanhat X-Ray järjestelmään kirjattut radiologiset pyyntö- ja vastaustekstit siirrettiin Efficaan RAD- lehdelle. Radiologian tilastointijärjestelmää kehitettiin Efficaan, ja siitä saatiin ensimmäinen toteutusversio testaukseen kesäkuulla 2006. Tilastoinnissa oli kuitenkin silloin vielä suuria puutteita, ja tilastojen teko ja laskutus eivät toimineet toivotulla tavalla. Tilastoinnin saaminen osaksi Effica RIS- järjestelmää (Radiology Information System) helpottaa radiologian sisäistä toimintaa. Radiologian RIS-järjestelmän kehittämistyö aloitettiin valtakunnallisesti pienryhmässä 2005 lopussa. Radiologian tilastointi siirrettiin osaksi radiologian yksikön RIS-toiminnan ohjausjärjestelmää helmikuussa 2008. Sovellukseen jäi testauksista huolimatta joitakin virheellisiä ja hankalaksi koettuja työvaiheita. Suurin osa ongelmista saatiin ratkottua vuoden 2009 aikana käyttöönotetun uusitun RIS II version myötä. [37; 41]

Digitaalisen kuva-arkiston hankinta tehtiin vuoden 2006 lopussa. Vuonna 2007 aloitettiin myös muiden yksiköiden digitaalisten kuvantamistutkimusten tulosten arkistointi Effican kuva-arkistoon. Sydäntutkimusyksikön ja P-osan leikkausyksikön kuvantamistutkimusten lisäksi marraskuussa otettiin koekäyttöön Effica endoskopiajärjestelmä vatsatautien tutkimusyksikössä. Kaikista edellä mainituista kuvantamistutkimuksista lisätään kuvalinkki manuaalisesti asianomaiselle erikoisalalomakkeelle. Kuvalinkkiä klikkaamalla käyttäjä saa kuvan auki Effican kuvankatseluohjelman avulla. [39]

Alue Effican käyttöönottoa valmisteltiin vuoden 2006 aikana ja varsinainen hankinta tehtiin loppuvuodesta. AlueKatselu järjestelmän avulla voidaan asiakkaan/potilaan terveys- ja sairauskertomustietoja tarkastella yli organisaatorajojen. Tämä edellyttää potilaan kirjallista suostumusta tietojen luovuttamisesta. Muualta saatujen kokemusten perusteella suurin hyöty järjestelmästä saadaan perusterveydenhuollon puolella ja yhteispäivystyksessä. [39].

SÄHKE-projektin väliraportissa vuonna 2007 todetaan ongelmaksi AlueKatselun avulla katseltaviin tietoihin liittyen se, että lääkelehden tiedot ovat vaillinaisesti kirjattuja kaikissa organisaatioissa. Myös EPSHP:n patologian lausuntojen sijaitseminen toisessa tietojärjestelmässä (Qpati) vaikeutti AlueKatselun hyödyntämistä alkuvaiheessa. Tämä korjaantui Effica – Qpati integraation myötä marraskuussa 2007. [39].

Effica laboratorion käyttöönotto osaksi Effica-järjestelmää alkoi toukokuussa 2007. Ennen tätä Effica laboratorio oli toiminut erillisenä sovelluksena. Laboratoriojärjestelmän käyttöönotto oli odotettu lisää Effica kertomuksen käytölle ja käyttäjät olivat tyytyväisiä, kun laboratoriovastaukset löytyivät helposti potilaiden kertomuksesta. Ensimmäisen version lähetekäsittely sai käyttäjiltä moitteita hitaudesta. [39].

Effica – Qpati integraatio otettiin EPSHP:ssä käyttöön marraskuussa 2007. Integraation myötä patologian pyynnöt tehdään aina Effica selainlaboratorion kautta. Patologian lähetteet ja vastaukset ovat luettavissa joko selainlaboratorion tai potilaan kertomuksen kautta. Lääkärit tekevät pyynnöt pääsääntöisesti itse. Käyttöönotto vaati työnkulullisia muutoksia yksiköiden käytännöissä. [39].

4.5.7 Hoitokertomus

Hoitosuunnitelma sairaanhoitopiiriin tarkoittamassa muodossa puuttui kokonaan Efficasta, samoin hoitoisuuden määrittelyyn tarkoitettu työkalu. Tämän kehittämistyön yhteisnimeksi vakiintui hoitokertomus. Hoitosuunnitelma on niin keskeinen elementti sairaalatyötä, että automaattisten yhteyksien järjestäminen sujuvaksi esimerkiksi lääkärinmääräys- osiosta hoidon toteutukseen ja arviointiin teetti paljon töitä. Hoitosuunnitelman pitää olla riittävän sujuva ja helppokäyttöinen, ja moninkertaista saman tietosisällön siirtelyä paikasta toiseen tulisi olla mahdollisimman vähän. Hoitosuunnitelmassa on keskeistä myös joustava linkitys hoitoisuusluokitukseen. Lääkärinmääräystoiminto on keskeinen osa hoitokertomusta sairaalassa. Vaatimusmäärittelyssä kirjattiin, että määräysten tulisi olla osa hoidon suunnittelua ja koordinaatiota eikä hoitosuunnitelmasta poikkeava erillinen osa. [36].

Sairaskertomus- ja hoitotaulukokehittämistyöryhmissä oli riippuvuuksia hoitokertomuksen kehitystyöhön, koska näillä oli automatiikkaa vaativia yhteisiä tietorakenteita. Toisaalta hoitokertomus edellytti tiettyjen potilashallinnon ratkaisujen määrittelyä ja toteutusta. Hoitokertomuksen kehitystyö tapahtui jo varhaisessa vaiheessa useiden keskussairaaloitten yhteistyönä, ja osallistuvien organisaatioiden määrä kasvoi kehitystyön edetessä. Hoitokertomuksen kehittämistä hidasti valtakunnallisten määritysten puute, ja yhteisten linjauksien hiomiseen kului paljon resursseja. [38]

Riippuvuudet järjestelmän muihin osiin ja kansallisiin kehityshankkeisiin, sekä epäselvyydet osapuolten vastuista eri vaiheissa viivästyttivät kehitystyötä ja aiheuttivat paljon muutostarpeita tehtyjen osapilotointien jälkeen. EPSHP:ssä tehtiin keväällä 2007 päätös, jonka mukaan rakenteisen Effica hoitokertomuksen (Whoike) käyttöönotosta luovuttiin, kunnes kehitystyö on siinä vaiheessa, että sovellus on toimivampi. Väliaikaisena ratkaisuna hoitotyön kirjaaminen kuitenkin siirrettiin sähköiseen muotoon hyödyntämällä Effica-kertomuksessa olevia strukturoituja erikoisalakohdaisia HOI- lomakkeita. Ratkaisu on tilapäisratkaisu siihen saakka, kunnes hoitokertomuksen kehitystyö etenee toivotulla tavalla (kesto arviolta 2-4 vuotta). [37; 39].

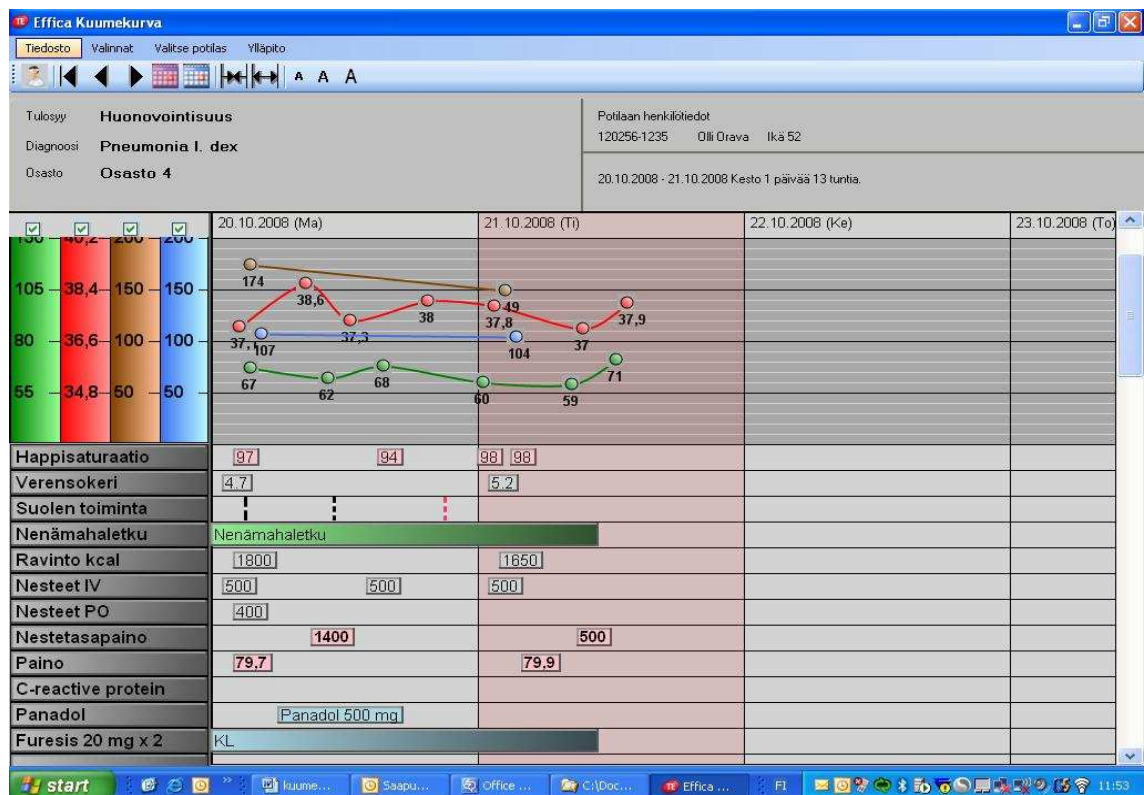
Kansallisen Proxit-hankkeen yhteyteen on perustettu Hoitokertomusryhmä, joka määrittelee Effica hoitokertomuksen ensimmäiseen versioon tehtävät muutokset. EPSHP:n hoitokertomuksen kehitystyö jatkui Proxit-hankkeeseen osallistumalla.

Vuoden 2008 aikana TietoEnator aloitti niin sanotun digitaalisen hoitotyön kehittämisohjelman, jonka tavoitteena on Effican Hoito-tuotteiden saumaton yhteensopivuus. Ensimmäisessä vaiheessa työn alle otettiin Effica Kuumekurva (korvaa aikaisemman hoitotaulukon) sekä Effica Hoitokertomus. Uusittu Effica Hoitokertomus- versio val-

mistunee vasta vuoden 2010 lopulla. Hoitokertomuksen ja Kuumekurvan välille tulee silloin olemaan kaksisuuntainen yhteys määriteltyjen merkintöjen osalta. Hoitokertomuksen osalta uusien osioiden määrittely on tapahtunut kansallisesti Proxit- hankkeen pienryhmässä. [41].

4.5.8 Hoitotaulukko/kuumekurva

Hoitotaulukon kehittämien Efficaan oli jo pitkällä Seinäjoen hankintapäätöstä tehtäessä. Hoitotaulukko tulee olemaan koko vuodeosastohoidon kokoajalehti potilaan ollessa osastohoidossa. Vuoden 2004 lopussa kehitystyö oli edennyt vaiheeseen, jossa se oli käyttöön otettavissa sellaisissa sairaaloissa, jotka pystyvät käyttämään hoitotaulukkoa erillisenä työkaluna vuodeosastotyössä. Seinäjoella tämä ei kuitenkaan ollut mahdollista, vaan mukaan haluttiin osioita erityisesti potilashallinnon taustatiedoista, hoitokertomuksen lääkärimääräysosiosta ja sairauskertomuksen lääkitysoasioista niin, että tietyt tiedot niistä kopioituisivat automaattisesti myös hoitotaulukkoon. Kehitystyö oli vuoden 2004 lopussa pysähtyksissä, koska järjestelmän muiden osien ratkaisut vaikuttavat keskeisesti tämän osion ratkaisuihin. Tuolloin arvioitiin käyttöönoton olevan mahdollista aikaisintaan vuoden 2006 lopussa. Kuva 12 on näyttökäyttö hoitotaulukosta eli kuumekurvasta. [38].



Kuva 12. Effica Kuumekurva

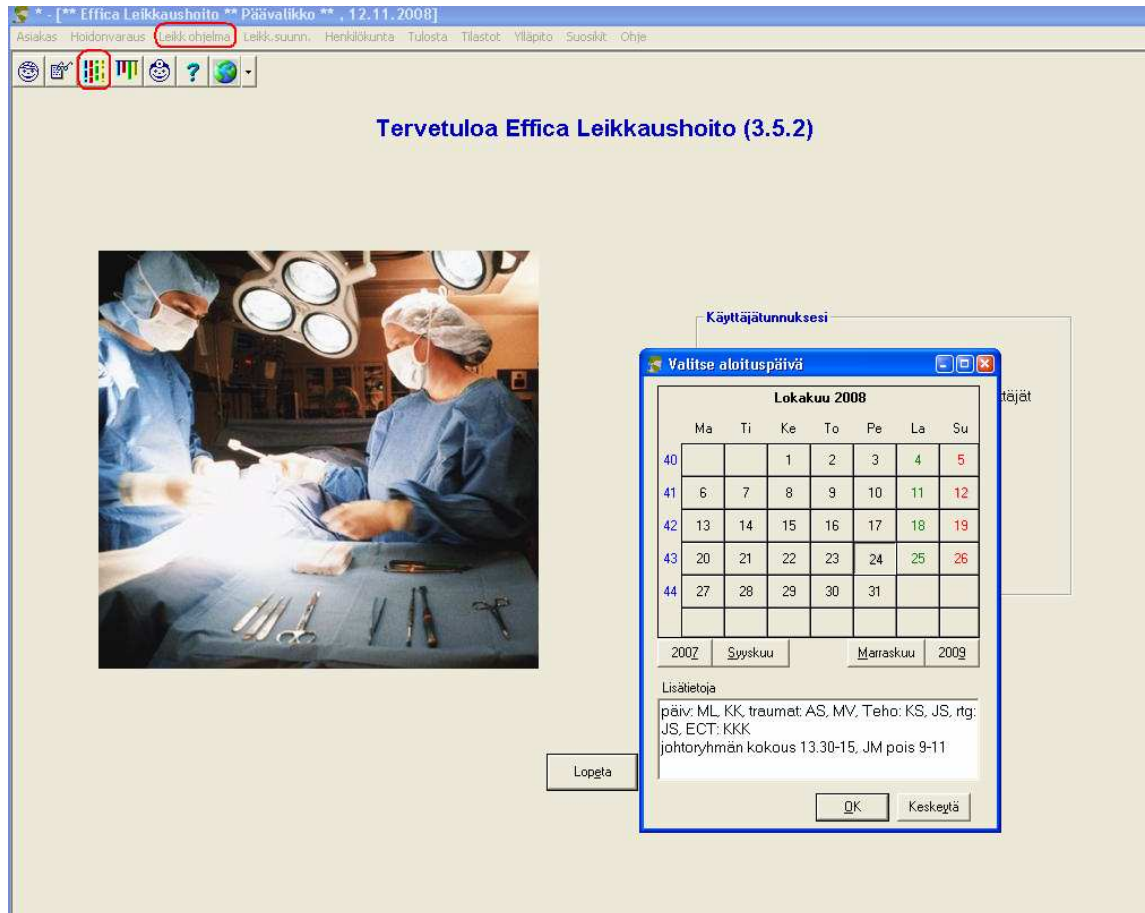
Useiden järjestelmän osien välisten riippuvuuksien kautta koko sähköisen kertomusjärjestelmän viivästyminen aiheutti merkittäviä viiveitä hoitotaulukon kehitykselle ja käyttöönnotolle. TietoEnatorin vuonna 2008 käynnistämän digitaalisen hoitotyön kehittämisohjelman myötä hoitotaulukon kehitystyö jatkui nimellä Effica Kuumekurva. Kuumekurvan on tuotantokelpoinen ja otetaan käyttöön keväällä 2010.

4.5.9 Leikkaushoitojärjestelmä

Effica-järjestelmän hankintaa päätettäessä EPSHP:ssä oli käytössä SAPO:n liittännäisjärjestelmänä sähköinen leikkausohjelman suunnittelujärjestelmä (LESU). Se tuotti hoidon- ja ajanvaraustiedoista alustavat leikkauslistat, joita täydennettiin potilaan hoitoyksikössä leikkausta edeltävänä päivänä. Leikkauspäivänä järjestelmään kirjattiin leikkauksen toteutum tiedot laskutusta ja toiminnan seuranta varten. Leikkaussalissa, välinehuoltokeskuksessa ja hoitoyksikössä voitiin monitorointitoiminnolla seurata leikkausten kulkua vaiheittain. [32].

Effica-leikkaushoidon pohjana on Tieto Enatorin omistama ruotsalainen Optimera-järjestelmä. Vuosina 2002–2003 eri sairaaloiden edustajista koottu asiakasryhmä teki määrittelytyötä Optimeran integroimiseksi Efficaan ja samalla toimintojen kehittämiseksi LESU:n korvaavalle tasolle. Vuodet 2004–2005 olivat hiljaiseloa asiakasryhmän sairaaloiden kannalta. Tänä aikana Tieto Enatorilla oli mittava käyttöönottoprojekti Keski-Suomessa. Effica Leikkaushoito asiakasryhmä perustettiin 2006 uudestaan siten, että mukana oli 1-2 edustajaa kaikista Effica-sairaaloista. Hämeenlinnassa testattiin pilottiversio, jolloin ensi kertaa päästiin vertaamaan toimintoja LESU-järjestelmään. Testauksessa löytyi virheitä ja toimimattomuuksia, jotka raportoitiin TietoEnatorille. Kehitystyö jatkui vastaavilla versiotestauksilla vuoden 2007 aikana. [37; 39]

Vuoden 2008 lopussa Effica Leikkaushoito- sovelluksen käyttöönotto tehtiin koko sairaanhoitopiirin alueella. Käyttöönotto jouduttiin tekemään puutteellisella yksisuuntaisella integraatiomallilla. Kuva 13 on Effica Leikkaushoidon päävalikosta.



Kuva 13. Effica Leikkaushoito

Effica Leikkaushoito-sovelluksen kaksisuuntainen integraatiotyö Effica Potilashallintoon on jatkunut samanaikaisesti taustalla vuoden 2008 aikana. Integraatiotyön kehittäminen on tapahtunut Päijät-Hämeen, Kanta-Hämeen, Keski-Suomen ja Itä-Savon sairaanhoitopiirien yhteisessä asiakasryhmässä.[41]. Vuoden 2009 aikana yhteistyö jatkui ja uusi paremman integraation mukainen sovellus saatiin käyttöön myös Etelä-Pohjanmaan sairaanhoitopiirissä.

4.5.10 SÄHKE-projektin päättäminen

SÄHKE-projekti päätettiin lopettaa vuoden 2009 lopussa [42]. Projektin toiminta oli vuosien aikana vakiintunut luonteeltaan jatkuvaksi, kuten sähköisen potilaskertomuksen kehityskin. Uusia merkittäviä käyttöönottoja oli vielä tulossa ja projektin tehtäviä tulee jatkamaan SÄHKE-työryhmä. Merkittäviä tulevaisuuden käyttöönottoja ovat muun muassa eResepti ja kansallisen terveystietokannan käyttöönotot.

4.6 Proxit-hanke

Proxit - hanke on kansallinen kymmenen Efficaa käyttävän sairaanhoitopiirin yhteishanke, jonka tarkoituksena on yhteisin resurssein täydentää ja muuttaa järjestelmää siten,

että se vuoden 2011 huhtikuun alkuun mennessä toimisi kansalliseen arkistoon potilas-kertomuksen ydintietoja tallentavana ja hakevana järjestelmänä. Hanke on alkanut vuonna 2005 ja sitä rahoittaa sosiaali- ja terveysministeriö. Hankkeen II - vaihe päättyi lokakuussa 2008 ja III -vaihe alkoi välittömästi sen jälkeen. Hanketta hallinnoidaan EPSHP:n toimesta. Hanke päättyi vuoden 2009 lokakuun lopussa rahoitusmielessä, mutta hanke jatkuu kymmenen sairaanhoitopiirin omarahoitteisena yhteishankkeena. [41].

Hanke on jakautunut työryhmiin. Alusta alkaen ovat toimineet omine alaryhmineen sairauskertomus-, potilashallinto- ja tekninen työryhmä. Vuoden 2007 aikana perustettiin omat pääryhmät hoitokertomuksen ja tietosuojan ympärille. Lisäksi varsinkin kertomusryhmän alla toimii useita pienryhmiä. Työ on edennyt kohtuullisen hyvin varsin laajoina kokonaisuuksina. Tyypillisimmin työ etenee määrittelyjen kautta tuotetilaukseen, edelleen suunnittelun ja toteutuksen kautta testaukseen, pilotointiin ja varsinaiseen käyttöönottoon. Myös kansalliseen määrittelyyn kuulumattomissa tilauksissa sairaanhoitopiirien yhteistyötä on hyödynnetty tämän hankkeen alla. [41].

Vuoden 2008 aikana testattiin suuria uusia kokonaisuuksia: osastolääkitystä, eResepti- toiminnallisuutta sekä Kertomus 2008 -kokonaisuutta, joka pitää sisällään ensimmäistä kertaa luokitellut yhtenäiset riskitiedot. Tuotteet toimitetaan versiotasolla 4, jonka ensimmäisen version 4.0 tuotantokäyttöön siirryttiin EPSHP:ssä keväällä 2009. [41].

Kansallisesti arkiston rakentuminen on edennyt hitaasti, mutta vuoden 2008 aikana puutteellisissa määrittelyissä edettiin kohtalaisen hyvin. Muutaman keskeisen menettelytavan etenemistä hidastaa puuttuva sähköisen potilaskertomuslainsäädännön asetus, jolla on tarkoitus täsmentää menettelytapoja. Vuoden 2008 lopussa perustettiin Kuntaliiton yhteyteen hanketoimisto, jonka tarkoituksena on muun muassa ohjeistaa KanTAPalveluun (kansallinen terveysarkisto) liittymistä. Alueellisena toimintana aloitettiin RA- pisteen perustamisen suunnittelu. Sitä tarvitaan keskeisesti henkilökorttien tekemisessä, henkilöiden varmentamisessa ja sähköisessä allekirjoituksessa. [41].

Proxit- hankkeessa on EPSHP:stä keskeisesti mukana 5-6 henkilöä varajäsenineen sekä EPSHP:n pääosin omistaman LifeIT Oyj:n tarjoama hallinnointi- ja sihteeriapu. [41].

SÄHKE ja Proxit ovat erillisiä projekteja. Proxit-hankkeeseen on siirtynyt sellaisia SÄHKE-projektin alueita, joille on yhteistä kiinnostusta kaikissa Efficaa käyttävissä sairaanhoitopiireissä. SÄHKE-projektin rooli onkin vahvistunut koko ajan enemmän käyttöönottoprojektiksi. Proxit-hanke ei kuulu tämän arviointityön alueeseen, mutta sitä tullaan käsittelemään niiltä osin kun se liittyy läheisesti SÄHKE-projektin hallintaan. Tästä seikasta johtuen, Proxit-hankkeen aikana tapahtuneet muutokset uusien ominaisuuksien kehitysvaiheen projektihallintamenetelmissä eivät välttämättä tule esille tässä selvityksessä.

4.7 Koulutus

Sähköiseen potilaskertomusjärjestelmään siirtyminen muutti merkittävästi koko organisaation ydintoimintojen toteuttamistapoja. Uuden järjestelmän onnistunut käyttöönotto edellytti henkilökunnan kattavaa koulutusta ja perehdyttämistä uuteen järjestelmään ja sen mukana muuttuneisiin toimintakäytäntöihin. Koulutustarve tiedostettiin jo SÄHKE-projektin esiselvitysvaiheessa ja hankintapäätöksen jälkeen nimettiin yksi kuudesta käynnistämisyhtymästä vastaamaan käyttöönottokoulutuksesta.

Alkuvaiheessa painopiste oli henkilöstön yleisen ATK-valmiuden nostamisessa. Hankintapäätöksen jälkeen koulutus alkoi keskittyä käyttöönotettavien järjestelmän osien perehdyttämiseen. Koulutus on jatkuva aktiviteetti. Järjestelmään valmistuu koko ajan uusia komponentteja, joihin käyttäjien tulee perehtyä. Lisäksi uusille työntekijöille pitää kouluttaa heidän käyttötarpeitaan vastaavasti myös olemassa olevan järjestelmän toimintaa. Myös uusintakoulutuksia pitää järjestää monista syistä johtuen. Vuosittain koulutuksien osallistujamäärät olivat noin 1000-2000 osallistujaa.

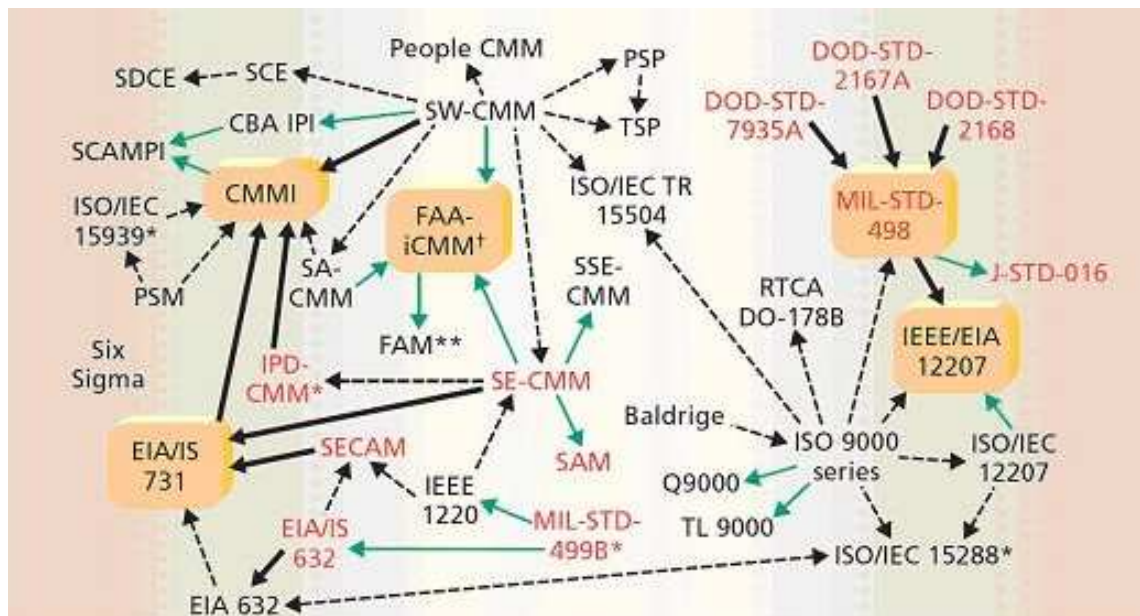
Käyttäjien kouluttajat olivat sairaalan omaa henkilökuntaa. Varsinaisen koulutustilaisuuksien järjestämisen lisäksi kouluttajien tehtävänä on antaa käyttöteknistä tukea käyttäjille heidän ongelmatilanteissaan. Kouluttajien asiantuntemus oli myös tarpeen uusien ohjelmisto-osien ja -versioiden testauksessa sekä käyttöönotossa. [37].

5 Arviointimenetelmät

Nykytilanteen analysointi ja ymmärtäminen on yleinen lähtökohta kaikelle kehitystyölle. Monet ohjelmistotuotantoprosessien kehitysmenetelmät tarjoavatkin hyvät välineet myös ohjelmistoprojektien arvioinnille. Näitä menetelmiä voidaan soveltaen käyttää myös ohjelmistohankintaprojektin arviointiin. Tässä luvussa esitellään tunnetuimmat ohjelmistoprojektien arviointimenetelmät ja valitaan niistä SÄHKE-projektin arviointitarpeisiin parhaiten soveltuva.

5.1 Projektihallintaprosessien arviointimenetelmät

Ohjelmistoprojektien arviointimenetelmiä on olemassa lukuisia, joista jokainen katsoo ohjelmistoprosessien kehitystä ja arviointia omasta näkökulmastaan. Kustakin mallista löytyy omat vahvuutensa ja heikkoutensa. Moni näistä malleista pohjautuu yhteen tai useampaan edeltävään malliin. Sarah A. Sheard kuvasi tilannetta menetelmien rämeiköksi (*Framework Quagmire*) [43]. Oheisessa kuvassa on hänen näkemyksensä vuoden 2001 prosessitilanteesta ja eri mallien keskinäisistä riippuvuussuhteista.



Kuva 14. Ohjelmistokehitysprojektin arviointimenetelmien rämeikkö (*Framework Quagmire*) [43]

Seuraavissa luvuissa kuvataan tarkemmin laajasti käytetty ja monien mallien pohjana toiminut SEI:n (*SW Engineering Institute*) CMM-malli (*Capability Maturity Model*)

[44], ISO-standardointijärjestön ISO15504 ja CMM-mallin seuraajan CMMI:n erityisesti ohjelmistohankintaprosessien kehittämiseen tarkoitettu CMMI-ACQ (*CMMI for Acquisition*) [45].

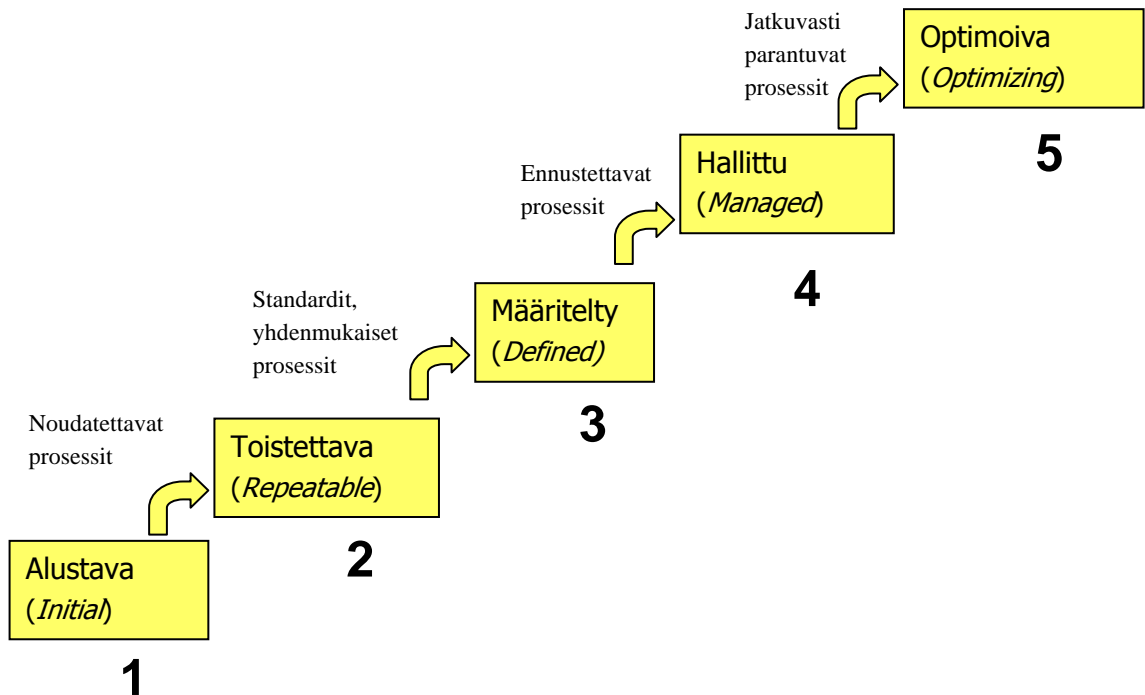
5.1.1 CMM

CMM on ohjelmistotuotannon parantamiseen kehitetty vaihemalli, joka perustuu kypsyytasoihin. SEI julkaisi siitä ensimmäisen version 1.1 vuonna 1993 [44]. Se luokittelee organisaatiot niiden toiminnan kypsyyden mukaan. Epäkypsässä organisaatiossa toiminta tapahtuu yleisesti improvisoidusti ilman etukäteen määritellyn prosessin ohjausta. Vaikka prosesseja olisi määriteltykin, niitä ei seurata. Tällaisessa organisaatiossa aikataulut ja budjetit ylitetään säännönmukaisesti, koska estimaatit eivät ole realistisia. Ja mikäli tosiasioihin perustumattomat aggressiiviset aikataulut saavutettaisiinkin, se tapahtuu toiminnallisuuden ja laadun kustannuksella [44].

Kypsässä organisaatiossa hallitaan ohjelmistokehityskäytännöt ja ylläpitoprosessit koko organisaation laajuisesti. Prosesseista tiedotetaan selkeästi sekä nykyisille että uusille työntekijöille ja työtehtävät suoritetaan määritellyn prosessin mukaisesti. Prosesseja päivitetään tarpeen vaatiessa ja prosessiparannuksia kehitetään hyvin hallituilla menetelmillä. Roolit ja vastuut on selkeästi määritelty. Kypsässä organisaatiossa ohjelmistotuotteiden laatua ja asiakastyytyväisyyttä seurataan. Aikataulut ja budjetit perustuvat seurattuun suorituskykyyn ja ovat realistiset. Pääsääntöisesti kustannus-, toiminnallisuus-, aikataulu- ja laatutavoitteet saavutetaan.

CMM on organisaation vaiheistettu evoluutiopolku, joka määrittelee organisaation kehitysaskleet sen nykyisestä kypsyytasosta vaiheittain kohti ylintä kypsyytasoa. Vaiheistetun mallin tarkoitus on kohdistaa prosessikehitysaktiviteetit oikeassa vaiheessa ja järjestyksessä eri prosessialueille huomioiden niiden keskinäiset riippuvuussuhteet. Ilman ohjaavaa mallia prosessikehitys saattaa olla tehotonta, jos kehitetään alueita joille ei ole olemassa vielä niiden vaatimaa tukea muilta prosessialueilta [44].

CMM-mallin viisi kypsyystasoa on esitetty oheisessa kuvassa (Kuva 15). Nuolien vieressä olevat tekstit kuvaavat seuraavalle tasolle siirtymiseen vaadittavia prosessien ominaisuuksia.



Kuva 15. CMM-mallin kypsyystasot [44]

Ensimmäinen taso on lähtötaso (*Initial*), jolle ei ole mitään kriteerejä. Tälle tasolle tunnusomaista on, että käytännöt syntyvät akuuteista tarpeista improvisoimalla. Vain muutamia käytäntöjä saattaa olla määritelty ja organisaation menestyminen on kiinni osaavista yksilöistä. [44].

Kypsyystaso 2:lle (*Repeatable*) siirryttäessä kustannuksia, aikatauluja ja toiminnallisuutta seuraavat prosessit on luotu ja niitä noudatetaan. Uusien projektien suunnittelu ja johtaminen perustuu samantyyppisistä projekteista saatuihin kokemuksiin ja onnistumiset on toistettavissa. Yksityiskohtaiset käytännöt saattavat kuitenkin vaihdella projekti-kohtaisesti. [44].

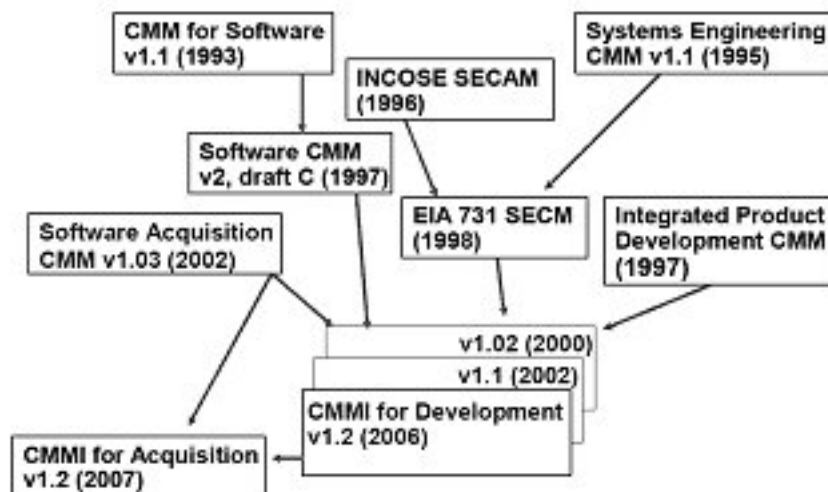
Tasolla 3 (*Defined*) sekä projektin johtaminen että toteutusaktiviteetit on dokumentoitu, standardoitu ja integroitu organisaation yleiseen ohjelmistoprosessiin. Kaikki projektit käyttävät hyväksyttyä projektikohtaista versiota organisaation yleisestä prosessista. [44].

Tasolla 4 (*Managed*) mitataan yksityiskohtaista tietoa ohjelmistoprosessista ja tuote-laadusta. Projekti saa paremman kontrollin prosessista ja tuotteistaan pienentämällä mitaustulosten vaihtelua hyväksyttävien kvantitatiivisten rajojen sisälle. Sekä tuotteet että prosessi on kvantitatiivisesti ymmärretty ja kontrolloitu. [44].

Viimeisellä tasolla 5 (*Optimizing*) jatkuva prosessien parantaminen ja optimoiminen onnistuu prosessista saatavan kvantitatiivisen palautteen avulla sekä pilotoimalla innovatiivisia ideoita ja teknologioita. [44].

Kaikkiin kypsyystasoihin, ensimmäistä lukuun ottamatta, on yhdistetty useampi niin sanottu avainprosessialue, KPA (*Key Process Area*). Ne määrittävät alueet, joiden hallintaa tietyn kypsyystason saavuttaminen edellyttää. Referenssinä käytetyssä CMM versiossa 1.1 [44] tasoon 2 oli liitetty kuusi projektin perusprosessialuetta (muun muassa vaatimushallinta, projektisuunnittelu sekä projektiseuranta ja -valvonta), tasoon 3 on liitetty seitsemän prosessialuetta, tasoon 4 kaksi ja tasoon 5 kolme aluetta. Alemmilla tasoilla saavutettu prosessikyvykyys pitää säilyttää ylemmillä kypsyystasoilla ja yksittäiseltä prosessialueelta vaadittavat prosessikäytännöt lisääntyvät ylemmille kypsyystasoille siirryttäessä.

CMM-mallin pohjalta on kehitetty lukuisia muita ohjelmistoprosessin parannusmalleja (katso ”SW-CMM” Kuva 14). Myös SEI on kehittänyt useita CMM-variaatioita erityistarpeisiin. Kuva 16 kuvaa CMM-mallien kehityspolkuja.



Kuva 16. CMM-mallien kehitys [45]

Vuonna 2000 julkaistu CMMI (*Capability Maturity Model Integration*) yhdisti erilliset CMM-mallit. Tämän jälkeen CMMI-mallista on tehty erilliset versiot ohjelmistokehityksen lisäksi ohjelmistohankintaan ja palveluhankintaan.

5.1.2 ISO 15504

ISO 15504 mallin kehitystyö aloitettiin vuonna 1993 nimellä SPICE (Software Process Improvement and Capability dEtermination). Edelleenkin virallisen ISO-standardinimen ohessa mallista käytetään myös alkuperäistä SPICE nimeä. ISO 15504 on CMM/CMMI -mallin kanssa laajimmin levinneitä ohjelmistoprosessien kehitysmenetelmiä.[46].

ISO 15504 soveltuu CMM-mallin tavoin sekä prosessin kehittämiseen että sen arviointiin. ISO 15504 perustuu jatkuvaan malliin CMM-mallista poiketen, joka eteni vaiheittain. Jatkuva malli mahdollistaa prosessialueiden arvioinnin ja kehityksen prosessi-kohtaisesti eikä niitä ole sidottu yhteen kuten CMM kypsyytastojen prosessialueet. CMM-mallien pohjalta luotuun CMMI-malliin tuli vaihemallin rinnalle myös mahdollisuus prosessialueiden itsenäiseen kehittämiseen.

ISO 15504 mukaan prosessien kyvykkyys luokitellaan kuudelle tasolle:

Taulukko 2. ISO 15504 prosessien kyvykkyystasot [46]

Taso 0	Keskeneräinen	Tavoitteiden saavuttamisessa pääsääntöisesti epäonnistutaan. Ei yhtään mallissa arvioitavaa ominaisuutta.
Taso 1	Toimiva	Yksi ominaisuus: Suorittaminen (PA 1.1). Arvioidaan prosessin tarkoitusta sen tulosten perusteella.
Taso 2	Hallittu	Kaksi ominaisuutta: Suorittamisen hallinta (PA 2.1) ja Tuotosten hallinta (PA 2.2). Arvioidaan suorittamisen hallintaa ja prosessin sisään- ja ulostulevien tuotosten hallintaa.
Taso 3	Vakiintunut	Kaksi ominaisuutta: Prosessin määrittely (PA3.1) ja Prosessin hyödyntäminen (PA3.2). Arvioidaan prosessin tehokasta määrittelyä ja dokumentaatiota sekä sen käytön laajuutta ja ymmärtämistä organisaatiossa.
Taso 4	Ennustettava	Kaksi ominaisuutta: Prosessin mittaus (PA4.1) ja Prosessin kontrolli (PA4.2). Arvioidaan tuotosten ja prosessien tehokasta mittausta ja metriikkoja ja kvantitatiivista prosessihallintaa.
Taso 5	Optimoiva	Kaksi ominaisuutta: Prosessi-innovaatio (PA5.1) ja Prosessioptimointi (PA5.2). Arvioidaan prosessin muuttamista ja muutosten vaikutusten tehokuutta.

Arvioitavien prosessien ominaisuudet arvioidaan neliportaisella asteikolla: ei saavutettu (*Not*), osittain saavutettu (*Partially*), laajasti saavutettu (*Largely*) ja täysin saavutettu (*Fully*). Jotta prosessi saavuttaisi tietyn tason, kaikki tälle tasolle kuuluvat ominaisuudet pitää olla täysin tai laajasti saavutettu.

5.1.3 CMMI-ACQ

CMMI-ACQ (*CMMI for Acquisition*) [45] on CMM-mallin seuraajan CMMI:n pohjalta kerätty kokoelma ohjelmistohankinnan kannalta oleellisia käytäntöjä. Sen tarkoitus on ohjelmistohankintaprosessien parantaminen ja arviointi. Mallin kohdealue on ohjelmistohankinnan asiakkaan toimintamenetelmät, vaikka moni tarkasteltava prosessialue onkin läheisesti kytköksissä myös toimittajan toimintaan. CMMI-ACQ –mallin kehityshistoria näkyy luvun 5.1.1 kuvassa (Kuva 16).

Mallin komponentit on ryhmitelty kolmeen kategoriaan: vaadittaviin, odotettuihin ja informatiivisiin. Organisaation pitää toteuttaa prosessialueen vaaditut komponentit saavuttaakseen sen hyväksytysti. Odotetut komponentit kuvaavat toimia, joita organisaatio voi toteuttaa saavuttaakseen vaaditut komponentit. Informatiiviset komponentit on tarkoitettu selventäviksi ohjeiksi. CMMI-ACQ ei määrittele tiettyjä toimintatapoja, jolla organisaation pitäisi ohjelmistohankintaa tehdä. Se määrittelee ainoastaan, että organisaatiolla on hyväksyttävällä tasolla olevat toimintamallit tarkasteltavalla prosessialueella. Tämä tapahtuu vertailemalla organisaation prosesseja mallin prosessialueen tavoitteisiin.

CMMI käyttää prosessien kehittymisen kuvaamiseen tasoja. CMM-mallissa käytettyjen kypsyystasojen (katso luku 5.1.1) lisäksi prosessiparannuksia voidaan tehdä vaiheittain yksittäisille prosessialueille. Tämän niin sanotun jatkuvan kehitystavan prosessialuekohtaisia vaiheita kutsutaan kyvykkyystasoiksi. Kyvykkyystasoja on kuusi (0-5) ja kypsyystasojen lukumäärä on viisi (1-5). Oleellinen sisältö molemmissa lähestymistavoissa on sama ja ne hyödyntävät samoja mallin komponentteja. Jatkuva toteutus antaa organisaatiolle mahdollisuuden valita parannuskohteiksi ne prosessialueet, jotka hyödyttävät niitä parhaiten omassa toiminnassaan. Prosessialueiden välillä on kuitenkin riippuvuuksia, jotka rajoittavat jossain määrin vapaata valintamahdollisuutta. Vaiheittaisessa toteutuksessa prosessialueet on ryhmitelty valmiiksi kypsyystasojen 1-5 mukaisiin ryhmiin. Kypsyystasojen saavuttaminen edellyttää siihen kuuluvien prosessialueiden tavoitteiden täyttämisen. Varsinaisten prosessialuekohtaisten tavoitteiden (*Specific Goals, SG*) lisäksi jokaiseen kypsyystasoon liittyy yleinen tavoite (*Generic Goal, GG*). Yleinen tavoite koskee kaikkia tiettyyn kypsyystasoon ja sitä edeltäneisiin tasoihin liitettyjä prosessialueita. [45].

CMMI-ACQ –mallissa on 22 prosessialuetta. Ne on esitetty taulukossa 3 ryhmiteltyinä niitä vastaavan kypsyystason mukaisesti.

Taulukko 3. CMMI-ACQ kypsyystasot ja prosessialueet

Kypsyystaso 1. Alustava (<i>Initial</i>)	Ei prosessialueita
Kypsyystaso 2. Toistettava (<i>Repeatable</i>)	Tarjouspyyntömenettely ja hankintasopimuksen tekeminen
	Sopimushallinta
	Vaatimusten kehitystyö
	Vaatimushallinta
	Projektisuunnittelu
	Projektin seuranta ja ohjaus
	Prosessin ja tuotelaadun varmistus
	Konfiguraatioiden hallinta
	Mittaus ja analyysi
Kypsyystaso 3. Määritelty (<i>Defined</i>)	Hankinnan tekninen hallinta
	Hankinnan hyväksyntä
	Hankinnan varmistus
	Päätösanalyysi ja ratkaisu
	Integroitu projektihallinta
	Organisationaalinen prosessimäärittely
	Organisationaalinen prosessin kohdealue
	Organisationaalinen koulutus
	Riskien hallinta
Kypsyystaso 4. Tilastollisesti hallittu (<i>Quantitatively Managed</i>)	Organisaation prosessien suorittaminen
	Tilastollinen projektihallinta
Kypsyystaso 5. Optimoiva (<i>Optimizing</i>)	Syy-yhteyden analysointi ja ratkaisu
	Organisationaalinen innovointi ja hyödyntäminen

5.2 Arviontimenetelmän valinta

Luvussa 2 todettiin, että ohjelmistoprojekteja voidaan toteuttaa monin eri tavoin. Listattujen perusmallien lisäksi käytännön variaatioita on lukemattomia. Myös projektien hallintaprosessien arviointiin on kehitetty valmiita menetelmiä, joita käsiteltiin luvussa 5.1. Valmiin arviointimallin käytön lisäksi arviointia voidaan tehdä myös vertailemalla projektia tunnettuihin projektimalleihin. Ohjelmistoprojekteja on erilaisia ja harvoin mikään malli sellaisenaan on paras ratkaisu yksittäisen projektin käyttöön. Kuten projektin alussa toimintaprosessia määritettäessä, myös sen toimintaa arvioitaessa on huomioitava projektin luonne ja toimintaympäristö.

Ohjelmistoprojektin hallintaprosessin arvioinnissa projektin toimintatapoja voidaan verrata valittuun viitemalliin. Jos projektin alussa on päätetty toteuttaa projekti jonkin tunnetun projektimallin mukaan tai vaihtoehtoisesti projektille on kirjattu oma toimintaprosessi, arvioinnin ensivaiheessa voidaan selvittää miten hyvin projekti on toteuttanut valitsemaansa toimintamallia. Luvussa 5.1 käsitellyt prosessien arviointimenetelmät antavat yksittäistä projektimallia laajemman valikoiman menetelmiä ja monipuolisen mahdollisuuden rakentaa näistä kullekin projektille sopiva toimintamalli. Ne antavat myös hyvät työkalut erityyppisten projektiprosessien arviointiin.

SÄHKE-projekti on ohjelmiston hankintaprojekti, ei ohjelmistotuotantoprojekti. Erityisesti kun hankintaprojekti osallistuu myös ohjelmistokehitykseen, kuten SÄHKE-projektissa, hankinta- ja tuotantoprosessit sisältävät hyvin paljon samoja komponentteja. Ohjelmistokehitysprosesseja on täten mahdollista käyttää viitemallina hankintaprojektin prosesseja määritettäessä tai arvioitaessa.

SÄHKE-projektissa ei ollut dokumentoitua toimintaprosessia. Vakiintuneita toimintamalleja syntyi projektin edetessä, mutta varsinaista kokonaisprosessin suunnittelua tai kehitystä ei projektin aikana tehty [47]. Tämän vuoksi toiminnan arviointia ei voitu tehdä vertailemalla sitä projektissa määriteltyyn prosessiin. Periaatteessa olisi ollut mahdollista verrata SÄHKE-projektin toimintaa johonkin sen toimintatapaa läheisesti muistuttavaan projektinhallinnan perusmalliin. Tässä tilanteessa kuitenkin varsinaisesti projektin arviointiin kehitetyt menetelmät tarjosivat monipuolisimmat työkalut. Pitkän kehityshistorian, laajan käytön, hyvän dokumentoinnin ja erityisesti hankintaprojektin arviointiin kehitetyn moduulin vuoksi arvioinnin viitemalliksi valittiin *CMMI for Acquisition* (CMMI-ACQ) [45]. Mallia käytettiin väljästi soveltaen, eikä tarkoituksena ollut tuottaa viralliseen CMMI-arviointiin verrattavaa arviointitulosta. Viitemalli takaa kuitenkin kohdeprojektin tarkastelun riittävän monipuolisesti.

Kuten luvusta 5.1.3 todettiin, niin vaikka CMMI-malli antaa mahdollisuuden valita ja kehittää itsenäisesti yksittäisiä prosessialueita, antavat eri kypsyystasojille ryhmitellyt prosessialueet organisaatioille suositusjärjestyksen prosessialueiden kehittämiseksi. Tästä johtuen arvioinnissa keskitytään kypsyystasoon 2, joka on alin taso, mihin on liitetty prosessialueita. Tasolla 2 näitä on yhteensä yhdeksän.

6 Tulokset ja niiden tarkastelu

SÄHKE-projektin ohjelmistohankintakäytäntöjä arvioitiin *CMMI for Acquisition* (CMMI-ACQ) [45] mallin avulla. Virallisen CMMI-arvioinnin tasoon ei ollut tarkoituksenmukaista pyrkiä vaan mallia käytettiin takaamaan riittävän monipuolinen tarkastelu ja tarkasteltavaan projektiin parhaiten soveltuva rajaus.

Tässä luvussa kerrotaan aluksi arviointimallin käytötapa, ja tämän jälkeen käydään läpi arviointitulokset prosessialuekohtaisesti. Myös toimittajan näkemys SÄHKE-projektista on esitetty luvun lopussa.

6.1 SÄHKE-projektin vertailu viitemalliin

CMMI-ACQ ei määrittele, että organisaation pitäisi seurata tiettyä hankintaprosessia. Se edellyttää vain, että organisaatiolla on käytössään malliin verrattavissa olevia prosessikäytäntöjä [45, s. 31]. Arvioinnissa oli tarkoitus huomata ne SÄHKE-projektin käytännöt, jotka vastaavat mallin käytäntöjä ja toisaalta tuoda esille sellaisia mallin alueita ja käytäntöjä, joista voisi olla hyötyä vastaavanlaisessa projektissa. Koska SÄHKE-projektissa ei ollut käytössä dokumentoitua ja formaalia hankintaprosessia, ei vertailumallistakaan kannattanut arvioida kuin alimman määritellyn kypsyystason 2 mukaisia prosessialueita.

Kypsyystason 2 saavuttaminen edellyttää tähän kypsyystasoon liitettyjen prosessialueiden erityistavoitteiden täyttämistä. Lisäksi mallin yleisistä tavoitteista (*Generic Goal*) edellytetään kahden ensimmäisen, GG1 (*Performed process*) ja GG2 (*Managed process*) [45, s. 33], saavuttamista. SÄHKE-projektin lähtötaso huomioiden ei GG2-tavoitteen mukaisia käytäntöjä ollut kuitenkaan tarkoituksenmukaista arvioida. GG2-käytännöt liittyvät organisaation prosessikäytäntöjen systemaattiseen perustamiseen ja suunnitteluun. Tällaisia aktiviteetteja ei Etelä-Pohjanmaan sairaanhoitopiirissä ollut. Kypsyystason 2 saavuttaminen ei myöskään ollut tämän arvioinnin tavoitteena. Tarkoitus oli saada arviointiin kattavasti tarkoituksenmukaiset projektihallinnan osa-alueet. GG1-tavoitteessa on vain yksi yleinen käytäntö. Se on prosessialueiden erityistavoitteiden saavuttaminen. Tämän vuoksi arviointi voitiin rajoittaa vain kypsyystason 2 prosessialueiden erityistavoitteiden tarkasteluun.

6.1.1 Tarjouspyyntömenettely ja hankintasopimuksen tekeminen

Tämä prosessialue sisältää kolme erityistavoitetta: tarjouspyynnön valmistelu, toimittajavalinta ja hankintasopimuksen laatiminen. [45, s. 365]

Tarjouspyynnön valmistelua SÄHKE-projektissa on käsitelty luvussa 4.1. Järjestelmän tarpeita kartoitettiin ja potentiaalisia toimittajia selvitettiin kattavasti. Tarjouspyyntö jätettiin helmikuussa 2001 [32]. Tarjouspyynnön sisältö on tarkoituksenmukainen ja vastaa hyvin vertailumallissa määriteltyä sisältöä [45, s. 367-368]. Mallissa esille tuotuja toimitusprosessin kuvauksia ei tarjouspyynnössä ollut. Nämä määriteltiin yhdessä valitun toimittajan kanssa myöhemmässä vaiheessa.

Tarkempia tietoja SÄHKE-projektin toimittajavalinnasta on luvussa 4.3. CMMI-ACQ vertailumalli korostaa huolellista tarjousten läpikäyntiä ja jatkokommunikointia tarjousten jättäneiden kanssa. 13 tarjouksen joukosta valittiin kolme, joiden kanssa käytiin tarkentavia neuvotteluja. Toimittajavalinta sisälsi vertailumallissa kirjatut käytännöt: tarjousten arviointi, jatkoneuvottelujen suunnittelu ja toimittajavalinta.

CMMI-ACQ mallissa on tyypillisiä hankintasopimukseen kuuluvia kohtia [45, s. 381]. Niiden ei ole tarkoituksaan olla kattava luettelo erilaisista hankintasopimusten sisällytettävistä asioista, vaan tarjota viitteellinen esimerkki. Hankkeiden erilaisesta luonteesta johtuen eri kohtien tarpeellisuutta pitää harkita tapauskohtaisesti. SÄHKE-projektin hankintasopimus sisälsi oleellimmat CMMI-ACQ -mallissa kirjatut asiat. Merkittävimmät puutteet malliin nähden olivat sopimuksen ylläpidosta ja tarvittavien muutosten hallinnasta sopiminen sekä vaatimusten rakenteen selkeys.

Hankintasopimuksessa tulisi määritellä selkeästi menetelmät, joilla sen sisällön muutostarpeita seurataan. Projektin aikana toimintaympäristö usein muuttuu. Tämän myötä syntyy tarpeita uusille vaatimuksille ja jotkut alkuperäisistä vaatimuksista tulevat tarpeettomiksi. Toimittajan kehitystyön aikana voidaan huomata, että joitakin vaatimuksia on mahdoton toteuttaa. Hankintasopimuksen sisältö tulee päivittää vastaamaan uusia tarpeita ja muutosten vaikutukset hinnoitteluun ja aikatauluihin pitää päivittää.

Menetelmät sopimuksen muuttamiseksi ja muutosten hyväksyntäprosessi tulisi kirjata sopimukseen. SÄHKE-projektin hankintasopimuksessa ei ole kirjattu sopimuksen muutoshallintamenetelmiä eikä alkuperäistä sopimusta myöskään päivitetty projektin aikana. Toimittajan kanssa järjestettiin kuitenkin vuosittain sopimuksen tarkastelutilaisuuksia, joissa päivitettiin projektin aikataulua ja hinnoittelua [48; 49].

Alkuperäiset kehityshankkeiden työmääräarviot osoittautuivat alimitoitetuiksi, mikä oikeutti toimittajan sopimuksen mukaisiin hinnankorotuksiin. Myös alkuperäisen sopimuksen piiriin kuulumattomien vaatimusten toteutukset vaikuttivat projektin etenemiseen ja kustannuksiin. Muutokset kirjattiin sopimuksen tarkastelukokousten pöytäkirjoihin, mutta alkuperäisestä sopimuksesta ei tehty päivitysversioita.

Hankintasopimuksen vaatimukset oli määritelty sopimusliitteenä olevassa alkuperäisessä tarjouspyyntöasiakirjassa. Ne olivat hyvin alustavassa muodossa ja niiden rakenne

ei tukenut seurantaa ja ylläpitoa. Ne olisivat vaatineet projektin edetessä tarkennettua muotoilua.

Tarkennettuja vaatimusmäärittelyjä tehtiin kehitystyöryhmissä. Niille ei ollut määritelty yhtenäistä sisältörakennetta tai käsittelytapaa. Työryhmien vaatimusmäärittelyistä ei tehty yhteenvedoa, mikä esti tehokkaan vaatimushallinnan projektitasolla. Ohjelmistoon haluttuihin uusiin toiminnallisuuksiin viitattiin ”toiveina”, ei vaatimuksina, ja niiden dokumentoitu muotoilu oli pääosin otsikkotasolla ja siten usein tulkinnanvarainen [50]. Tätä tulkinnanvaraisuutta on voitu SÄHKE-projektissa ainakin osittain korjata suullisesti asiakkaan ja toimittajan välisissä palaverissa. Kuitenkin esimerkiksi potilashallinnon kehityshankkeen osalta oli asiakkaalla ja toimittajalla vuoden 2005 lopulla eri käsitys alkuperäisen sopimuksen mukaisen toiminnallisuuden toteutumisesta [49]. Vaatimushallinnan arviointia on käsitelty tarkemmin luvussa 6.1.4.

Paljon uusia kehityshankkeita sisältävässä projektissa on vaikea nähdä kaikkia tarpeita yksityiskohtaisesti heti sopimusta tehtäessä. Tämän vuoksi sopimuksen ylläpitoon liittyvien asioiden kirjaaminen alkuperäiseen sopimusversioon on tärkeää.

6.1.2 Sopimushallinta

Tämän prosessialueen tarkoitus on varmistaa, että asiakas ja toimittaja toimivat hankintasopimuksen mukaisesti. Sopimushallinnan yhden tavoitteen alla mallissa kirjattuja käytäntöjä ovat sopimuksen toteuttaminen, toimintaprosessien seuraaminen, toimitusten hyväksyminen ja laskutus. [45, s. 86]

Sopimuksen toteuttamisessa CMMI-ACQ korostaa asiakkaan ja toimittajan välistä kommunikointia ja raportointia. SÄHKE-projektissa toiminta oli jakaantunut käyttöön-otto- ja kehitystyöpääaktiviteetteihin. Lisäksi nämä jakaantuivat useisiin osa-alueisiin ja vaiheisiin. Eri alueilla puhtaasti käytännön aktiviteettien vuoksi yhteistyön ja seurannan tarve oli erilaista. Projektin alussa Effica-ohjelmiston käyttöönotto alkoi tietyiltä osin nopeasti (tekstinkäsittely, palautejärjestelmä, digitaalinen sanelu ja purku) ja sen myötä myös kommunikointi toimittajan kanssa oli tiivistä näillä alueilla. Myös laajat kehitysalueet, joilla osittaista käyttöönottoa aloitettiin jo projektin alkuvaiheessa (potilashallinto ja sairaskertomus) vaativat jatkuvaa kanssakäymistä asiakkaan ja toimittajan välillä. Laajojen kehitysprojektien seuranta ja ohjaus tapahtui TietoEnatorin koordinoimien asiakastyöryhmien avulla. Osa-alueilla, joilta tarvittava toiminnallisuus puuttui ja käyttöönoton aloittamiseen kului vuosia, oli suurin riski puutteelliseen seurantaan. Esimerkiksi hoitokertomuksen kehitystyö vaati laajaa määrittelytyötä, joka tapahtui projektin ulkopuolisissa asiakasryhmissä ja toimittajan toimesta. Ajoittain kanssakäyminen toimittajan suuntaan oli vähäistä [37].

Projektin organisaatorakenne ja kokouskäytännöt tukivat sopimuksen toteutumisen seurantaa. Yksityiskohtaisen ja kattavan projektisuunnitelman laatiminen, säännöllinen seuranta ja ylläpito olisivat antaneet paremman kokonaiskuvan projektin tilanteesta. Myös säännöllinen kokonaistilanteen raportointi toimittajan taholta esimerkiksi kuukausittain olisi saattanut parantaa näkyvyyttä.

Toimittajan toimintaprosessien seuraaminen on CMMI-ACQ:n mukaan tarpeellista kun asiakkaan ja toimittajan prosessit ovat tiiviisti kytköksissä keskenään. SÄHKE-projektin toiminta ei perustunut vakiintuneille toimintaprosesseille siinä määrin, että tällä olisi ollut merkitystä. Tässä tapauksessa riittävää parannusta toimittajan toiminnan seuraamisessa olisi saatavissa jo muiden toiminta-alueiden (projektisuunnittelu, raportointi ja ylläpito) kehittämisellä.

Toimitusten hyväksyminen tapahtui lukuisten käyttöönottojen kautta. Näiden tulokset ja havaitut virheet raportoitiin toimittajalle. Suurempien ohjelmistokokonaisuuksien ja versioiden käyttöönoton virallinen hyväksyntä tapahtui projektiryhmän kokouksissa. Laskutuskäytännöt oli sovittu hankintasopimuksessa [48]. Käytännössä laskutuksen ja käytännön työn kohdistaminen oli haastavaa [47].

6.1.3 Vaatimusten kehitystyö

Tämän prosessialueen tarkoitus on kehittää ja analysoida kahden pääluokituksen mukaisia vaatimuksia, jotka ovat asiakasvaatimukset ja sopimus pohjaiset vaatimukset. Asiakasvaatimukset määrittelevät sidosryhmien tarpeet tuotteille ja palveluille. Sopimus pohjaiset vaatimukset ovat asiakasvaatimusten muotoilu tarjouspyynnön ja hankintasopimuksen vaatimaan muotoon sekä asiakkaan ja toimittajan välistä toimintaa määrittävät ei-tekniset vaatimukset. Tämä prosessialue sisältää kolme erityistavoitetta: asiakasvaatimusten luominen, sopimusvaatimusten luominen ja vaatimusanalyysi. [45, s. 93]

Projektin sidosryhmien (esimerkiksi asiakkaat, loppukäyttäjät, toimittajat, testaajat, hallinto, tukioorganisaatiot) tarpeet ja odotukset analysoidaan ja yhdenmukaistetaan asiakasvaatimuksiksi. Tyypillisesti sidosryhmien tarpeet eivät ole täsmällisesti määriteltyjä ja ovat usein keskenään ristiriitaisia. Tämän vuoksi tarpeiden selkeyttäminen ja ymmärtäminen on tärkeää koko projektin elinkaaren ajan. Vaatimusten tarpeellisuus on syytä varmistaa säännöllisesti projektin edetessä. Olosuhteet muuttuvat ja ymmärrys lopputuotteen luonteesta jalostuu kehitystyön aikana. Näiden seurauksena alkuperäiset tarpeet, niiden tärkeys ja suhde muihin vaatimuksiin usein muuttuu.

Jo ennen varsinaisen SÄHKE-projektin käynnistämistä alkanut sähköisen sairaskertomuksen selvitys- ja tarvekartoitustyö jatkui projektin toimesta. Tarpeita ja toimittajien valmiuksia selvitettiin varsin kattavasti. Tätä käsiteltiin tarkemmin luvussa 4.1. Alkuperäiset vaatimukset olivat mukana tarjouspyynnössä [32]. Osa vaatimuksista oli sähköisen sairaskertomuksen suunnittelun myötä syntyneitä täysin uusia toiminnallisuuksia, mutta iso osa perustui olemassa olevien vanhojen tietojärjestelmien ja paperilomake pohjaisten toimintatapojen siirtämiseen uuteen sähköiseen järjestelmään.

Olemassa olevaan toiminnallisuuteen pohjautuvan vaatimuksen määrittely yksiselitteisesti on periaatteessa helppoa ja voidaan tehdä hyvinkin suppealla kuvauksella kun viitataan nykyiseen referenssitoiminnallisuuteen. Vaadittavan toiminnallisuuden toteuttaja pystyy epäselvissä tilanteissa löytämään oikean tulkinnan vaatimukselle referenssitoteutuksesta. Täysin uusista tarpeista syntyneiden vaatimusten kuvaukset pitää olla paljon yksityiskohtaisemmat. Usein kehitystyön alkuvaiheessa monista syistä johtuen

riittävän tarkan vaatimusmäärittelyn tekeminen uusille tarpeille on mahdotonta. Tämän vuoksi on tärkeätä, että vaatimusten rakenne ja hallintaprosessi tukee jatkuvaa päivytystä ja ylläpitoa. SÄHKE-projektin alkuperäisiä vaatimuksia ei päivitetty, mutta kehitystyöryhmissä tehtiin niiden pohjalta aluekohtaisia vaatimustarkennuksia. Ne olivat kuitenkin pääosin otsikkotasoisia listoja joiden tarkentaminen jäi toimittajan kanssa käytäviin palaverihin [50]. Koko projektin kattavan vaatimusmäärittelyn puute esti myös täsmällisen toimitusten sisällön seuraamisen siten, että kulloiseenkin ohjelmistoversioon toteutetut vaatimukset olisi pystytty yksilöimään.

CMMI-ACQ –malli määrittelee, että asiakasvaatimusten perusteella pitäisi jalostaa täsmälliset sopimusvaatimukset. Toiminnallisten vaatimusten lisäksi sopimusvaatimukset määrittelevät toiminnan ei-tekniisiä kriteereitä. Näitä ovat esimerkiksi toimintatavat, raportointi, kommunikointi, takuuasiat ja vaadittavat dokumentit. Jossain määrin edellä mainittuja asioita on käsitelty SÄHKE-projektin hankintasopimuksessa. Toimintatapoja kuten projektiraportointi, vaatimushallinta tai sopimusten ylläpito ei hankintasopimuksessa määritelty. Näiden mukanaolo alkuperäisessä sopimuksessa olisi saattanut suunnata projektikäytäntöjä varmempaan ja helpommin hallittavaan suuntaan.

Vaatimusanalyysin tarkoitus on varmistaa, että lopullinen toteutus täyttää tarpeet asiakkaan käyttöympäristössä. Tämä on myös luonteeltaan koko projektin elinkaaren ajan jatkuva toiminto jolla pyritään reagoimaan toimintaympäristössä ja vaatimuksissa tapahtuviin muutoksiin. SÄHKE-projektissa ei ollut muodollista vaatimushallintaprosessia, joten tätäkään aktiviteettia ei pystytty järjestelmällisesti toteuttamaan. Työryhmä tasolla käytiin kuitenkin projektin edetessä keskusteluja toteutuksen sisällöstä toimittajan kanssa ja sisäisesti. Toiminnallisuuksia priorisoitiin ja poistettiin sopimuksen piiristä. Vaatimusanalyysialueen kehittäminen käynnistyisi automaattisesti, jos vaatimushallintaprosessia ylläpidettäisiin.

6.1.4 Vaatimushallinta

Vaatimushallinnan tarkoitus on ylläpitää vaatimuksia ja tunnistaa poikkeavuuksia vaatimusten, projektisuunnitelmien ja toimitettavien tuotteiden välillä [45, s. 341]. Vaatimusten hallinta ja vaatimusten kehitys prosessialueet liittyvät toisiinsa läheisesti ja toimivat projektissa samanaikaisesti.

Ennen vaatimusten toteutuksen suunnittelun aloitusta varmistetaan vaatimuksen alkuperäisestä lähteestä, että analysoitu vaatimus on vielä yhdenmukainen alkuperäisen tarpeen kanssa. Vaatimukseen sitoutumisella tarkoitetaan projektiin osallistujien ja toimittajan hyväksymistä ja uuden vaatimuksen aiheuttamien muutosten tekemistä projektisuunnitelmiin, aktiviteetteihin ja tuotetoimituksiin.

Projektin edetessä vaatimukset muuttuvat monista syistä johtuen. On olennaista saada nämä muutokset ja lisäykset tehtyä tehokkaasti. Muutoksen vaikutuspiirissä olevien vaatimusten alkuperäinen lähde pitää tuntea, jotta muutoksen vaikutukset tulevat oikein ymmärretyksi. Myös muutoksen perustelut on hyvä dokumentoida. Vaatimusmuutos saattaa aiheuttaa tarvetta myös hankintasopimuksen muuttamiseen. SÄHKE-projekti

kesti pitkään ja sen kuluessa tapahtui paljon muutoksia. Projektin toimintaympäristössä tapahtui merkittäviä muutoksia erilaisten kansallisten hankkeiden myötä. Muut sairaanhoitopiirit aktivoituivat ja perustettiin asiakasryhmiä ajamaan tiettyjen Efficahojelmiston osa-alueiden kehitystyötä. Proxit-projekti perustettiin vuonna 2005 ajamaan keskitetysti kansallisen potilasarkiston vaatimien muutosten saamista Efficiaan. Terveystuennon lainsäädännön muutokset heijastuivat usein myös ohjelmistoon. Esimerkiksi hoitotakuu ja potilastietojen käsittelyyn liittyvät lain muutokset vaativat muutoksia ohjelmistoon [47]. SÄHKE-projektin vaatimusten muutosten hallinta tapahtui hajanaisesti työryhmätasolla koko projektin kattavan vaatimushallinnan puuttuessa.

Tämän prosessialueen käytäntönä on kirjattu myös vaatimusten kaksisuuntainen jäljitettävyyden. Se tarkoittaa, että tiedetään mitkä vaatimukset on toteutettu ja toimitettu asiakkaalle ja toisaalta toimitetut kokonaisuudet pystytään yhdistämään tiettyihin vaatimuksiin. Tämä on erittäin tärkeää, jotta asiakkaalle on selkeä ymmärrys toimituksen ja tehdyn sopimuksen vastaavuudesta. Kaksisuuntainen jäljitettävyyden edellyttää selkeää vaatimushallintaa, jossa yksilöidyt vaatimukset on linkitetty toimituksiin ja tätä yhteyttä ylläpidetään jatkuvasti projektin edetessä. SÄHKE-projektissa vaatimushallinta ei ollut sillä tasolla, että tämä olisi ollut mahdollista. Vaatimusten keskinäinen riippuvuus oli vaikea hallita. Tästä esimerkkinä on potilaan riskitietojen esittäminen, joka sisältyi tarjouspyynnön vaatimukseen [32]. Toteutusvaikeuksien vuoksi tämä päätettiin toteuttaa vain tiettyihin ohjelmiston näkymiin ja muihin se lisättäisiin vasta myöhemmässä vaiheessa [47]. Toimivassa vaatimushallinnassa tilanne olisi hoidettu vaatimusten keskinäisen riippuvuuden kuvaamisella. Sovitun muutoksen jälkeen alkuperäistä vaatimusta olisi päivitetty, riippuvuuksia muutettu ja mahdollisesti luotu uusi vaatimus puuttuvaksi jääneestä toiminnallisuudesta.

6.1.5 Projektisuunnittelu

Projektisuunnitteluprosessin tarkoitus on projektin aktiviteettien suunnittelu ja tehdyn suunnitelman ylläpito [45, s. 281]. Prosessin osa-alueita ovat varsinaisen projektisuunnitelman tekeminen, vuorovaikutus sidosryhmien kanssa, suunnitelmaan sitoutumisen aikaansaaminen ja suunnitelman ylläpito. Projektisuunnitelman lähtökohtana ovat vaatimukset, jotka määrittelevät tuotteen ja projektin. Prosessialueen erityistavoitteita ovat arvioiden laatiminen, projektisuunnitelman tekeminen ja suunnitelmaan sitoutumisen aikaansaaminen.

Projektisuunnitteluparametreille tehdään arviot ja niitä ylläpidetään. Nämä parametrit sisältävät kaiken tiedon projektin suunnitteluun, organisointiin, henkilöresursointiin, johtamiseen, raportointiin ja budjetointiin. Arviot perustuvat hankintastrategiaan ja sisältävät myös toimittajan alustavat arviot omasta osuudestaan.

Hankintastrategia muodostaa suunnittelun taloudellisen ja teknisen kehyksen. Strategiaan liittyy hankinnan tavoitteet, rajoitteet, henkilö- ja teknologiaresurssien saatavuus, hankintamenetelmien analysointi, vaihtoehtoiset sopimustyytit, sopimusehdot, taloudelliset näkökulmat, riskit ja hankittavan tuotteen tuki sen koko elinkaaren ajan. Se heijas-

telee koko projektin toimintakenttää. SÄHKE-projektissa näitä asioita kartoitettiin ennen tarjouspyyntöä ja määriteltiin lopullisesti hankintasopimuksessa [48]. Vuoden 2000 projektiraportissa [31] todettiin, että sähköisen sairaskertomusohjelmiston hankinta ei ole minkään olemassa olevan ohjelmiston osto vaan tulee vaatimaan räätälöintiä ja uusien toiminnallisuuksien kehittämistä. Tämän seurauksena projekti käynnistyessään jakautui kahteen pääaktiviteettiin: käyttöönottoon ja uusien ominaisuuksien kehittämiseen.

Projektin toimintakenttä määritellään ylimmän tason tehtäväjaottelun (*WBS, Work Breakdown Structure*) avulla. Tämä tehdään alustaviin vaatimusmäärittelyihin pohjautuen. Tehtäväjaottelu kehittyy ja tarkentuu projektin edetessä. Tehtäväjaottelu tarjoaa myös pohjarakenteen tehtävien työmääräarvioille. Se antaa mekanismin työmäärien, aikataulujen ja vastuiden kohdentamiseen yksittäisiin tehtäviin. SÄHKE-projektin hankintasopimuksessa tehtiin työmääräarvio kuudelle kehittämisalueelle. Mikäli näihin alueisiin kohdistuvat vaatimukset olisi identifioitu, ylimmän tason tehtäväjaottelu luotu ja työmääräarviot kohdennettu tehtäviin, olisi saatu hyvä pohja projektisuunnitelmalle. Varsinaisen vaatimusjaottelun jälkeen tehtävien pilkkominen ja jatkosuunnittelu olisi pitänyt tapahtua toimittajan toimesta ja SÄHKE-projektin tehtävänä olisi ollut vain sen seuranta.

Projektin jaksottaminen päävaiheisiin tarjoaa seurantapisteitä projektin hallintaan ja päätöksentekoon. SÄHKE-projektissa jaksotus tapahtui eri käyttöönottovaiheiden mukaan. Käyttöönotot jaksottuivat melko tasaisesti koko projektin ajalle.

Projektisuunnitelma on muodollinen ja hyväksytty dokumentti, jota käytetään projektin ohjaamiseen ja hallintaan. Se pohjautuu projektin vaatimuksiin ja tehtyihin kustannus- ja työmääräarvioihin.

Projektin budjettia ja aikataulua tulee seurata ja ylläpitää koko projektin ajan. Aikataulutuksen perusteena olevat oletukset on hyvä tuoda esille. Ne antavat käsityksen kokonaisuikataulun luotettavuudesta. Rajoitteet ja tehtävien väliset riippuvuudet tulisi tunnistaa. SÄHKE-projektissa käyttöönottovaiheiden suunnittelu oli huomattavasti kehityshankkeita tarkempaa. Tämä johtui todennäköisesti siitä, että ne olivat suurelta osalta myös asiakkaan puolella tehtävää työtä.

Kehityshankkeiden suunnitelmat jäivät lähes hankesopimuksen tasolle, jota tapauskohtaisesti tarkennettiin projektin edetessä projektiryhmän ja työryhmien palaverissa sekä yhteistyössä toimittajan kanssa. Systemaattista vaatimuksiin pohjautuvaa suunnittelua ei tehty. Eri osa-alueiden väliset riippuvuudet paljastuivat työn edetessä ja aiheuttivat paljon viiveitä, joita ei osattu ennakoida. Esimerkiksi hoitokertomus ja hoitotaulukon kehitystyötä viivästytti potilashallinnon ja muiden osa-alueiden puuttuva toiminnallisuus [37; 38; 40].

Riskit tulee tunnistaa ja analysoida projektisuunnitelman tueksi. Niiden mahdolliset vaikutukset tulee selvittää ja ne olisi priorisoitava. SÄHKE-projektissa ei ollut järjestelmällistä riskianalyysiä, mutta yksittäisiä riskejä nostettiin esiin keskusteluissa. Näitä olivat esimerkki käyttäjien tietoteknisen osaamisen riittämättömyys, Effica-järjestelmän keskeneräisyys, aikataulujen viivästyminen ja versiopäivitysten aiheuttamat riskit. [47].

Projektin aikana kertyy paljon tietoa (esimerkiksi analyyskejä, raportteja, pöytäkirjoja, sopimuksia). Projektitiedon hallinnan suunnittelu auttaa varautumaan oleellisen tiedon asianmukaiseen käyttöön, arkistointiin ja päivitykseen. SÄHKE-projektin aikana syntyi paljon materiaalia, mutta koska dokumentointia ei tehty minkään ennalta määritellyn prosessin mukaan ei ollut mahdollisuutta laatia etukäteen dokumentointisuunnitelmaakaan.

SÄHKE-projektin resurssitarpeita suunniteltiin erityisesti projektiryhmässä ja käynnistämistyöryhmissä. Käynnistämistyöryhmien työ liittyi käyttöönottojen valmisteluihin, jotka vaativat pääosan koko projektin resursseista. Tekninen työryhmä suunnitteli käyttöönottoon liittyvät tekniset yksityiskohdat ja tarvittavat materiaaliressurit. Koulutus-työryhmä suunnitteli käyttökoulutuksen vaatimat henkilöresurssit. Myös laajan koulutuksen vaatimiin tiloihin, laitteisiin ja ohjelmistoihin varauduttiin.

Dokumentoitu projektisuunnitelma varmistaa, että projektin työhönsä tarvitsema tuki saadaan kommunikoidua ja sovittua sidosryhmien kanssa. SÄHKE-projektissa ei ollut erillistä koko projektin kattavaa projektisuunnitelmadokumenttia. Suunnitelmat kommunikointiin projektin sisäisissä ja ulkoisissa palaverissa. Vuosittaisiin ohjausryhmän kokouksiin valmisteltu edeltävän vuoden tapahtumat kokoava väliraportti sisälsi myös jatkosuunnitelmaosuuden. Näissä kirjattiin päälinjaukset lähitulevaisuuden suunnitelluista projektiaktiiviteeteista. Tarkempi projektisuunnitelma olisi edellyttänyt systemaattisen vaatimushallinnan ja vaatimusdokumentoinnin olemassaoloa.

Projektin loppuvaiheessa tapahtuneesta Effic version 3.5.01 käyttöönotto 2007 alkaen tehtiin omana aliprojektina, jossa tehtiin erillinen projektisuunnitelma tarkistuspiteen. Myös muut projektinhallinnan osa-alueet kuten riskien hallinta ja raportointi olivat paremmin huomioitu tässä aliprojektissa. Effic versio 3.5.01 käyttöönottoprojektin tavoitteet toteutuivatkin pääosin aikataulussa. [51].

Projektin sidosryhmien sitoutuminen projektisuunnitelmaan takaa, että projektin edellyttämä työ saadaan tehtyä. SÄHKE-projektin suunnitelmien hyväksyminen tapahtui ohjausryhmän kokousten kautta. Toimittajan sitoutuminen suunnitelmiin tapahtui hankintasopimuksen ja käytyjen sopimustarkastelukokousten välityksellä.

6.1.6 Projektin seuranta ja ohjaus

Tämän prosessialueen tarkoitus on tuottaa ymmärrys projektin etenemisestä, jotta tarvittavat korjaavat toimenpiteet voidaan tehdä, mikäli toteutuma poikkeaa merkittävästi suunnitelmista [45, s. 287]. Seuranta perustuu dokumentoituun projektisuunnitelmaan. Eteneminen määritellään vertaamalla tehtävien suunniteltuja työmääriä, kustannuksia ja aikataulua toteutuneisiin. Kun havaittu projektistatus poikkeaa merkittävästi oletetusta, tehdään korjaavia toimenpiteitä. Nämä voivat vaatia alkuperäisen suunnitelman tarkistamista ja lisäyksiä tai lievennyksiä sopimukseen. Mikäli korjaavia toimenpiteitä tarvitaan, niiden määrittely ja toteuttamisen seuranta kuuluu myös tähän prosessialueeseen. Toimittajan osuuden seuraaminen edellyttää säännöllistä raportointia toimittajalta asiakkaalle. Näiden käytäntöjen vaatimukset tulisi olla kirjattu hankintasopimukseen. Tä-

män prosessialueen erityistavoitteita ovat projektisuunnitelman etenemisen seuraaminen ja korjaavien toimenpiteiden hallinta.

Projektin etenemistä seurataan tuotteiden ja toimitusten ominaisuuksien, kustannusten, työmäärän ja aikataulun suhteen. Tuotteiden ja toimitusten ominaisuuksia ovat esimerkiksi koko, kompleksisuus, paino, muoto, sopivuus ja toiminnallisuus. Seuranta tapahtuu tyypillisesti mittaamalla näiden projektin suunnitteluparametrien toteutuneita arvoja ja vertaamalla niitä projektisuunnitelmassa tehtyihin arvioihin. Vertailussa havaitut merkittävät poikkeamat tunnistetaan. Tunnistettujen poikkeamien analysointi ja niiden vaatimien korjaavien toimenpiteiden määrittely kuuluu ”korjaavien toimenpiteiden hallinta” -erityistavoitteeseen.

SÄHKE-projektissa toimittajan kokonaistoimituksen toteutumista seurattiin hankintasopimuksen tarkastelukokouksissa [48; 49]. Työryhmissä seurattiin eri osa-alueiden tarkempaa edistymistä. Seuranta ei ollut järjestelmällistä, koska seuranta- ja raportointikäytäntöjä ei ollut määritelty. Suurempien toimituskokonaisuuksien valmistuttua järjestettiin toimittajakokousten yhteydessä osatoimitusten päätösraporttien katselmointi ja hyväksyntä.

Muodollisia etappi- tai vaihekatselmoiteja ei järjestetty. Hankintasopimuksen liitteessä 6 toimittaja kuvaa tuotekehitysprojektinsa vaiheet ja niistä syntyvät tulokset (dokumentit) [52]. Näiden pohjalta olisi ollut mahdollista järjestää etappiseurantaa ja katselmoiteja. Hankintasopimuksessa kuvattuja etappidokumenttien toimitusten toteutumista ei seurattu.

Kun seurannan kautta havaitaan, että toimittajan toiminta ei vastaa riittävässä määrin sopimuksessa määriteltyä tasoa, asiakas määrittelee toimittajan kanssa korjaavat toimenpiteet. Osa korjaavista toimenpiteistä voi olla pelkästään asiakkaan puolella tapahtuvia. Äärimmäisenä esimerkkinä tästä on vaihtoehtoisen toimittajan etsiminen. Kun korjaavat toimenpiteet on määritelty, täytyy myös varmistaa ja seurata, että ne myös saatetaan loppuun.

SÄHKE-projektissa tapahtui paljon poikkeamia alkuperäiseen suunnitelmaan nähden. Merkittävänä syynä tähän oli alkuperäisten suunnitelmien määrittely liian korkealla tasolla ja ettei niitä järjestelmällisesti tarkennettu ja seurattu projektin aikana. Uudet vaatimukset ja suuret muutokset toimintaympäristössä aiheuttivat muutostarpeita suunnitelmiin. Poikkeamia tapahtui toimitusten aikatauluissa ja niiden sisällöissä. Myös alussa tehdyt työmääräarviot ylittyivät runsaasti [40]. Useiden kehitysalueiden käyttöönottoja jouduttiin viivästäämään jopa vuosia. Strukturoitu kertomuksen käyttöönotto aloitettiin lokakuussa 2004, mutta jouduttiin keskeyttämään puutteellisen toiminnan vuoksi [38]. Käyttäjät oli ehditty jo kouluttaa käyttöönottoa varten. Strukturoidun kertomuksen käyttöönotto tapahtui keväällä 2005. Poikkeamiin piti reagoida korjaavilla toimenpiteillä. Merkittävään osaan poikkeamista piti reagoida aikataulua viivästäämällä ja tekemällä uudelleenbudjetointia. Korjaavien toimenpiteiden hallinnassa ei ollut käytössä muodollista prosessia. Poikkeamat olivat kuitenkin usein niin kriittisiä koko projektin kannalta, että ne saivat automaattisesti riittävän ison prioriteetin ja tulivat käsitteilyyn seurantapalaverissa.

6.1.7 Prosessin ja tuotelaadun varmistus

Tämän prosessialueen tarkoitus on tuottaa henkilöstölle ja organisaation johdolle objektiivinen näkemys toimintaprosesseihin ja niihin liittyviin tuotoksiin [45, s. 312]. Prosessialue keskittyy määriteltyjen toimintaprosessien arviointiin, arviointitulosten kommunikointiin ja tallentamiseen. SÄHKE-projektissa toimintatapoja ei ollut muodollisesti määritelty, joten niiden arviointi ja kehittämistäkään ei tehty suunnitelmallisesti.

6.1.8 Konfiguraatioiden hallinta

Konfiguraatioiden hallinnan tarkoitus on muodostaa ja ylläpitää tuotteiden rakenteellista yhdenmukaisuutta hyödyntäen konfiguraatioiden tunnistusta, hallintaa, tilanneseurantaa ja audiointeja [45, s. 146]. SÄHKE-projektin luonteeseen ei kuulunut tarve ylläpitää erilaisia versioita ja konfiguraatioita perusohjelmistosta. Toimittajalla tämänkaltainen tarve saattaisi ollakin, mutta se ei näkynyt asiakasprojektille. Tällä prosessialueella ei siten ollut merkitystä SÄHKE-projektille.

6.1.9 Mittaus ja analyysi

Tämän prosessialueen tarkoitus on kehittää ja ylläpitää mittausmenetelmiä, joilla tuetaan projektihallinnan tiedonkeräystarpeita [45, s. 183]. Mittauksen kohde on projektin toimintaprosessit. Mittaustiedon keräys parantaa ymmärrystä toiminnan tehokkuudesta ja laadusta. Se antaa menetelmän seurata prosessien kehitystoimenpiteiden tuloksellisuutta. Tämän tyyppistä toimintaa ei tehty SÄHKE-projektissa, eikä se olisi ollut tarkoituksenmukaistakaan projektin toimintaprosessien määrittelytaso huomioiden.

6.2 Toimittajan näkemys SÄHKE-projektista

Tieto Healthcare & Welfare Oy on antanut kirjallisessa muodossa palautetta SÄHKE-projektin osana tehdystä Effica v.3.5.01 käyttöönottoprojektista, joka aloitettiin 5.9.2007 [51; 53]. Projektin tarkoituksena oli Effica versio 3.5.01 käyttöönotto, SAPO-potilashallinnon järjestelmän korvaaminen Effica potilashallinnolla somaattisilla erikoisaloilla ja Effica Leikkaushoidon käyttöönotto. Kirjallisia palautteita täydentäviä arviota projektista saatiin puhelinhaastattelussa Tiedon edustajilta [54].

Projektihallinnan tarpeellisuudesta toimittaja antaa selkeää palautetta: ”Projektin aikana tuli selkeästi esille, että hyvällä suunnittelulla ja projektinhallinnalla pystytään viemään suuriakin projekteja hallitusti päätökseen ja toisaalta pienemmätkin käyttöönotot edellyttävät onnistuakseen projektinhallintaa. Projektinhallinta olisi pitänyt olla keskeisemmässä roolissa jo vuodesta 2002 alkaen.” [53]. Toisaalta keskustelussa [54] tuli esille toimittajan edustajien epäily, että vaikka projektihallintapalvelua olisi sai-

raanhoitopiirille projektin alkuvaiheessa tarjottu, asiakkaalla ei olisi ollut halukkuutta maksaa sen aiheuttamia kustannuksia.

Effica ja SAPO -järjestelmät ovat erilaisia perusteiltaan ja Effica-potilashallinnon kaikkia ominaisuuksia ei ole enää toteutettu samalla tavalla kuin vanhassa SAPO-järjestelmässä. Selkeämpi vaatimushallinta olisi toimittajan mukaan tuonut etua projektin ohjaamiseen ja kehittyneempien projektihallintamenetelmien käyttö SÄHKE-projektissa olisi ollut hyödyllistä myös toimittajan näkökulmasta. Monien avoimien asioiden olemassaolo projektin alussa oli tiedossa ja molemmat osapuolet ottivat tietoisesti riskin SÄHKE-projektissa. Toiminta perustui suuressa määrin keskinäiseen luottamukseen. [54]. Tätä edesauttoi, että yhteistyö asiakkaan kanssa sujui hyvin. Asiakkaan hyvä ammattitaito ja kyky informoida oikeista asioista oikealla tarkkuudella helpottivat ongelmatilanteiden ratkaisua toimittajan organisaatiossa [53].

SÄHKE-projektin osaprojektina ollut Effica v.3.5.01 käyttöönoton tavoitteet saavutettiin aikataulussa leikkaushoitoa lukuun ottamatta, jonka käyttöönotto siirrettiin asiakkaan pyynnöstä. Projekti alkoi 5.9.2007 ja leikkaushoito viimeisenä osana otettiin käyttöön 1.12.2008. Vaikka sekä toimittaja että asiakas olivat tehneet ennakkoon paljon testaustyötä, tuotantokäytössä tuli esille uusia virheitä. Reagointiaika Tiedon tuotekehitykselle virheiden korjaamiseksi jäi todella lyhyeksi. Resurssien rajallisuus ja aikataulu aiheuttivat sen, että kaikkia virheitä ei pystytty korjaamaan asiakkaan haluamassa aikataulussa, vaan niiden keskinäistä korjausjärjestystä jouduttiin priorisoimaan. Projektin päättyessä avoinna olevien virheiden korjausaikataulun tekeminen osoittautui haasteelliseksi, eikä siihen varattu aika riittänyt. Käyttöönottoprojektin lopullinen hyväksyntä ohjausryhmältä on edelleen saamatta.[53].

7 Johtopäätökset

Luvussa 3 käsiteltiin Suomen terveydenhuollon tietojärjestelmien kehitystä ja siihen liittyviä ongelmia. Monissa kriittisissä kommentteissa viitataan näiden kehityshankkeiden huonoon vaatimushallintaan, venyneisiin aikatauluihin ja muihin projektihallintaan läheisesti liittyviin alueisiin. Kuinka paljon terveydenhuollon tietojärjestelmähankkeiden onnistumista voitaisiin parantaa projektihallinnan menetelmien tehokkaammalla käytöllä?

Tässä työssä arvioitiin Etelä-Pohjanmaan sairaanhoitopiirin sähköisen potilaskertomusjärjestelmäprojektin (SÄHKE) hallintaa vertaamalla sitä CMMI-ACQ –mallin [45] kypsyystason 2 prosessialueisiin. Työssä ei pyritty tekemään virallisen CMMI-ACQ -mallin mukaista arviota. CMMI-ACQ –mallin prosessialueita käytettiin vertailuun, jotta arvioinnista tulisi riittävän monipuolinen ja kattava.

SÄHKE-projekti on ollut keskeisessä valtakunnallisessa roolissa Tieto Healthcare & Welfare Oy:n Effica-potilastietojärjestelmien kehitys- ja käyttöönotto-työssä vuosina 2002-2009. Tämän vuoksi SÄHKE-projektista tehdyt johtopäätökset saattavat olla hyödynnettävissä myös muissa vastaavissa hankkeissa. Vuonna 2005 käynnistynyt kymmenen Efficaa käyttävän sairaanhoitopiirin Proxit-hanke otti vähitellen vastattavakseen järjestelmän kehityksen ohjaamisen. Proxit-hankkeessa tapahtunut projektihallintamenetelmien kehitys ei näkynyt tässä arvioissa.

CMMI-ACQ –malli [45] antaa menetelmän organisaation toiminnan arvioimiseen mallissa määriteltyjen prosessialueiden osalta. Prosessialueet on liitetty niiden tärkeyden ja keskinäisen riippuvuussuhteen perusteella neljälle eri kypsyystasolle (alkaen tasolta 2). Lähtökohtana on, että erilaiset projektit erilaisissa ympäristöissä voivat itse määrittellä omaan tilanteeseensa sopivan toimintatapojen kypsyystason. Projektin tarkoitus on kuitenkin tuottaa tavoitteidensa mukainen lopputuote, eikä rakentaa tarkoitukseensa nähden ylimitoitettu resursseja kuluttava prosessibyrokraatia. Toisaalta projektin toimintaan nähden oikein mitoitettut prosessimäärittelyt varmistavat projektin tehokkaan työskentelyn laadukkaana lopputuloksen aikaansaamiseksi. Oikean prosessitason määrittelyyn vaikuttaa projektin tavoitteiden ja toimintaympäristön lisäksi se lähtötaso, jolla projektin toiminta on määritelty. Prosessialueilla on keskinäisiä riippuvuuksia, joten ensin täytyy käynnistää ne alueet jotka tarjoavat perustan muiden prosessialueiden kehittämiseksi.

Valittu arviointitapa SÄHKE-projektin projektinhallintamenetelmien arviointiin oli onnistunut. CMMI-ACQ –mallin [45] perusajatus eri kypsyystasoille kohdennettavista prosessialueista rajasi arvioitavat prosessialueet tarkoituksenmukaisesti arvioitavan pro-

jektin käytäntöihin nähden. Rajauksesta huolimatta mallin käyttö arvioitavien prosessi-alueiden valinnassa takasi riittävän kattavan tarkastelun.

Selkeä arviointia haittaava tekijä oli kirjattujen toimintaprosessien puuttuminen. SÄHKE-projektissa käytetyt menetelmät piti tulkita projektiraporteista ja selvittää suoraan projektiin osallistuneilta. Tämä oli huomattavasti hitaampaa kuin kirjattujen toimintaprosessien arviointi.

SÄHKE-projektin toimintaa arvioitaessa löytyi paljon parannuskohteita CMMI-ACQ –mallin prosessialuemäärittelyihin verrattuna. Vaikka kyseessä oli ensimmäisen määrittelyn tason prosessialueet, eivät kaikki prosessialueiden yksityiskohdat olleet perusteltuja kehityskohteita SÄHKE-projektin tapaiselle projektille. Selkeästi tärkeimmiksi kehitysalueiksi nousi järjestelmällisen vaatimushallinnan määrittäminen (luvut 6.1.3 ja 6.1.4) ja sopimusten ja suunnitelmien säännöllinen ylläpito. Suunnitelmien ylläpitoon kuuluu hankintasopimuksen ylläpito (luvut 6.1.1 ja 6.1.2) ja varsinaisten projektisuunnitelmien laatiminen ja ylläpito (luvut 6.1.5 ja 6.1.6). Näistä alueista huolehtiminen parantaa samalla toiminnan tasoa muillakin kuin omalla pääprosessialueellaan. CMMI-ACQ –mallin perusteella esille nostetut alueet ovat yhdenmukaisia luvussa 6.2 esitetyn toimittajapalautteen kanssa. Myös luvussa 3.3 käsitellyissä yleisissä terveydenhuollon tietojärjestelmiin liittyvissä ongelmissa on paljon yhteistä SÄHKE-projektin arvioissa havaittuihin kehityskohteisiin. Näin ollen työn tulokset voivat olla hyvinkin sovellettavissa muihin vastaaviin projekteihin.

Prosessinkehityshankkeet pitää tehdä organisaation sisällä eikä niitä pidä yrittää tuoda valmiina ulkopuolelta. Jokaisessa toimintaympäristössä on omat erityispiirteensä, jotka on syytä huomioida prosessikehityksessä. Näiden ominaispiirteiden tunnistaminen ei onnistu niin hyvin organisaation ulkopuoliselta taholta kuin organisaation sisällä työskentelevältä. Myös koko organisaation sitoutuminen prosessikehitykseen on parempaa silloin kun se lähtee vahvasti sisäisellä ohjauksella. Osaamista, näkemystä ja apua kehitystyöhön voidaan hankkia organisaation ulkopuolelta, mutta pääasiallinen vastuu ja ymmärrys menetelmien soveltamisesta pitää löytyä organisaation sisältä.

Suomen terveydenhuollon paikallisten organisaatioiden resurssit eivät välttämättä riitä itsenäisesti tarvittavan projektihallintaosaamisen kasvattamiseen tarvittavalle tasolle. Olisi varmasti perusteltua, että jokin terveydenhuollon tietojärjestelmien kansallinen toimija ottaisi vastuulleen ohjelmistoprojektien hallintaan liittyvän osaamisen ja tuen tarjoamisen sairaanhoitopiireille ja muille alueellisille organisaatioille. Tämän toiminnon kehittäminen voisi tapahtua esimerkiksi Kuntaliiton KunTo-toimiston puitteissa. KunTo tukee paikallisia organisaatioita KanTa-palveluun liittymisessä muiltakin osin.

Jokaisen hankkeen erityispiirteiden vuoksi ei kannata uskoa uuden menetelmän käänteentekevään vaikutukseen – ohjelmistotuotantomenetelmien ”hopealuotia” ei ole olemassa. Parhaaseen lopputulokseen päästään huomioimalla organisaation lähtötaso, poimimalla kaikkein oleellimmat kehitysalueet ja kehittämällä niitä vaiheittain. Prosessikehityksen vaikutuksia jatkuvasti seuraamalla pystytään tarkentamaan seuraavien vaiheiden sisältöä ja kehityskohteiden prioriteettia. Tästä johtuen tämän arvionkaan tu-

loksia ei pidä lähteä toteuttamaan suoraan ilman organisaation sisäistä kritiikkiä ja kehitystulosten seuranta.

Seuraavaksi esitellään toteutusjärjestyksessä SÄHKE-projektin lähtötilanteen (2001-2002) kaltaisessa vaiheessa olevalle projektille suositeltavat projektihallinnan kehitystoimet.

7.1 Elinkaarimallin määrittäminen

Projektin suoritusjärjestyksessä ensimmäinen merkittävä kehityskohde SÄHKE-projektissa olisi ollut projektin elinkaarimallin hahmottaminen. Projektin olisi ollut hyvä aloittaa elinkaarimallin miettiminen viimeistään tarjouspyynnön tekemisen yhteydessä, ja jatkaa sitä toimittajavalinnan jälkeen mahdollisesti toimittajan kanssa yhteistyössä. Luvuissa 2.3 ja 2.4.1 käsiteltiin ohjelmistoprojektin toimintaprosessin valintaa ja elinkaarimallin muokkaamista hankintaprojektin tarpeisiin sopivaksi. Luku 2.4.2 antaa erityisesti hankintaprojektiin sovellettuja käytäntöjä IEEE standardista IEEE std 1062 [12].

Periaatteessa organisaation kannattaisi määrittää projektiansa elinkaarimalli toimittajariippumattomaksi, jotta se ei vaikuttaisi toimittajavalintaan. Toisaalta erityisesti eiteknisen organisaation kannattaa pyrkiä siirtämään teknistä osaamista vaativat prosessialueet mahdollisuuksien mukaan toimittajan tehtäväksi. Kuitenkin huolehtien siitä, että selkeästi asiakkaan vastuulle kuuluvien prosessialueiden (esimerkiksi vaatimusmäärittely) kontrolli säilyy asiakkaalla. Tällaisissa tilanteissa tulee tarve linjata asiakkaan ohjelmistohankintaprojektin elinkaarimalli yhdenmukaiseksi toimittajan ohjelmistotuotannon elinkaarimallin kanssa. SÄHKE-projektin kaltaisessa pitkässä saman toimittajan kanssa tehtävässä projektissa molempien osapuolten prosessien keskinäinen synkronointi on perusteltua.

SÄHKE-projektin ei olisi kannattanut tarjouspyyntö- ja toimittajavalintavaiheessa laittaa elinkaarimallin määrittelyyn liikaa työpanosta. Hankintasopimuksen tekohetkellä tavoite projektin päättymisestä oli vuoden 2004 puolivälissä. Todellisuudessa projekti päättyi vuoden 2009 lopussa. Näinkin merkittävä projektin muuttuminen olisi näkynyt todennäköisesti myös tarpeena muuttaa elinkaarimallia. Olennaisinta SÄHKE-projektin alussa tehdyssä elinkaarimallissa olisi ollut, että se olisi sisältänyt iteratiivisen ja inkrementaalisen komponentin. Näin malliin olisi ollut sisäänrakennettuna itse elinkaarimallin mukauttaminen ympäristön muutoksiin.

7.2 Vaatimushallinta

SÄHKE-projektin tilanteessa tärkein kehityskohde olisi ollut jatkuvan vaatimushallinnan määrittäminen ja käyttöönotto. Tarkempia tietoja vaatimushallinta-alueesta on arviointituloksien luvuissa 6.1.3 ja 6.1.4. Vaatimusmäärittelydokumentin sisältö voidaan sovittaa projektin tarpeiden mukaan. Valmiita ohjeita sisällöstä ja esimerkkimalleja löytyy kirjallisuudesta ja standardeista [55]. SÄHKE-projektin tilanteessa järjestelmän kehitystyö tehtiin tiiviissä yhteistyössä toimittajan kanssa. Tässä tilanteessa luotaessa asi-

akkaan omaa vaatimushallintajärjestelmää on hyvä tutustua toimittajan omaan vaatimushallintaan ja varmistaa, että asiakkaalle luotava järjestelmä ei olisi rakenteellisesti ristiriidassa toimittajan järjestelmän kanssa. Saattaa olla mahdollista hyödyntää myös suoraan toimittajan omaa vaatimushallintajärjestelmää tuottamaan asiakkaan vaatimaa vaatimushallintamateriaalia.

SÄHKE-projektin toiminnallisten vaatimusten lukumäärä olisi noussut mahdollisesti satoihin erillisiin vaatimuksiin. Vaatimusten hallintaan on saatavissa erillisiä ohjelmistoja, mutta se onnistuu myös normaaleilla toimisto-ohjelmilla, kuten tekstinkäsittely-, taulukkolaskenta-, tai tietokantaohjelmistoilla. Suurta vaatimusmäärää hallittaessa on syytä varmistaa, että käytettävä vaatimusmäärittelyn rakenne tukee riittävän hyvin niiden seuranta. Vaatimuksissa tulee olla yksiselitteinen tunniste ja tilatieto, jotta niistä voidaan helposti koostaa kokonaistilanteen kertova yhteenveto. Vaatimuksen tila kuvaa sen elinkaaren vaihetta. Nämä voisivat olla esimerkiksi: uusi, toimittajahyväksyntä, suunniteltu, toimitettu, asiakashyväksyntä, suljettu. Lisäksi vaatimuksessa täytyy olla prioriteetti, jotta vaatimusten toteutusjärjestys voidaan suunnitella halutuksi.

Kuva 17 on esimerkki yksittäisen vaatimuksen suuntaa antavasta sisällöstä. Sen ei ole tarkoitus olla esimerkillinen toteutus Hoitotaulukko-vaatimuksen sisällöstä, vaan antaa konkreettinen mallin minkä tyyppistä tietoa toteuttaja tarvitsee vaatimukselta. Yleensä vaatimustekstiä joudutaan vielä täydentämään yhdessä toteuttajan kanssa, jotta yhteinen ymmärrys saavutettaisiin.

Vaatusmäärittely				Tunniste:	R.12
Prioriteetti:	Pakollinen	Luotu:	11.11.2001	Tila:	Toimitettu
				Lähde:	Osasto X/NN
Vaatumuksen nimi					
Hoitotaulukko					
Kuvaus:					
<p>Hoitotaulukko on yhden näytön kokoinen kooste potilaan tutkimuksista, tilasta ja lääkityksestä. Esitystavan pitää vastata nykyisin käytössä olevaan 'kuumekurva' (ks. liite "kuumekurve.gif"). Yläosassa näkyvät pakolliset potilaan tunnistetiedot, diagnoosi(t) sekä tutkimukset ja toimenpiteet, joita varten potilas on osastolla. Em. Tiedot siirtyvät hoitosuunnitelmalta tai läheteeltä. Näiden tietojen alapuolella on päivämääräsarakkeet ja mittaustulokset. Taulukon alaosassa ovat lääkitystiedot. Lisäksi pitää olla vapaasti kirjattavissa oleva "Huomioitavaa"-osuus. Mikäli minkä tahansa osion teksti on pidempi kuin näytölle sopii se pitää avautua kokonaisuudessaan kun hiirisoittimen siirtää alueen päälle.</p> <p>Mittaustaulukon ylimmässä ruudussa kunkin päivän kohdalla näkyy kyseisen päivän määräykset/tutkimukset/toimenpiteet. Ne kirjataan max. 4 kirjaimisilla lyhenteillä. Kirjaamisdialogi avautuu ruudusta hiiritoiminnolla.</p> <p>Päivämäärärivillä näkyy kerralla 10-14 päivän tiedot ja piilossa olevat päivät saadaan liutettua näytölle. Kun päivämääräpainiketta painaa avautuu näytölle avautuu täydennettävissä oleva hoitosuunnitelmalomake kyseisen päivän kohdalta.</p> <p>Näytön keskiosassa on varsinainen hoitotaulukko, jossa voidaan seurata päivittäin eri parametreja potilaasta graafisena esityksenä. Vaaditut parametrit ovat: lämpö, pulssi, RR, paino, happisaturaatio, nestetasapaino, sokeritasapaino, ravintoenergia ja psyykinen tila. Samasta potilaasta voidaan seurata useita eri muuttujia, mutta taulukossa näkyy yhden muuttujan arvot kerrallaan. Muut muuttujat löytyvät välilehtien takaa. Päivittäiset mittausarvot on syötettävissä hiiritoiminnolla avattavan dialogin kautta.</p> <p>Hoitotaulukon alaosaan tulee potilaan nykyinen lääkitys automaattisesti Lääkityslomakkeelta. Osastojakson aikana lääkitystietoja voidaan päivittää.</p>					
Vaatusriippuvuudet:				R.5,R.89	
Muutoshistoria:					
<p>Uusi. 11.11.2001</p> <p>Uusi. Kuvausta täydennetty toimittajan kanssa, 21.8.2002</p> <p>Toimittajahyväksyntä. 2.9.2002</p> <p>Suunniteltu. 14.12.2002</p> <p>Toimitettu. 4.10.2006</p> <p>Toimittajahyväksyntä. Edellinen toimitus ei täyttänyt vaatimuksia 18.10.2006</p> <p>Suunniteltu. Uusi aikataulu. 18.11.2006</p> <p>Toimitettu. 15.3.2007</p>					

Kuva 17. Esimerkki toiminnallisesta vaatimuksesta

Kaikki järjestelmältä vaadittava toiminnallisuus pitää kuvata vastaavalla tasolla. Tarvittavan dokumentaation määrän ja sen ylläpitämiseen tarvittavan työmäärän rajoittamiseksi kannattaa erillisiä vaatimuksia pyrkiä yhdistämään, jos se lopputulosta heikentämättä on mahdollista. Esimerkiksi jo olemassa olevan järjestelmän toiminnallisuksia ei välttämättä kannata eritellä omiin vaatimuksiin, mikäli ne vastaavat asiakkaan tarpeita. Tällöin voidaan tehdä vain yksi vaatimus, jolla viitataan tunnetun ohjelmistoversion kokonaissisältöön. Tämän jälkeen voidaan keskittyä vain uusien toiminnallisuuksien ja muutosvaatimusten erillisiin määrittelyihin.

Vaatimuksella on kaksi ensisijaista tehtävää: sen tulee määrittellä toimittajalle yksiselitteisesti mitä asiakas haluaa, ja sen pitää mahdollistaa toteutuksen seurannan vaati-

muksia vasten. Tämän lisäksi vaatimukseen voidaan lisätä valittua toimintaprosessia tukevaa tietoa, kuten suunniteltu toteutuspäivämäärä tai toteutuksen vaiheistustietoja. Kaikki lisätieto lisää kuitenkin vaatimusten ylläpitoon vaadittavaa työmäärää. Tässä piilee sama riski kuin kaikessa prosessihallinnan parannuksissa: otetaan käyttöön liian kunnianhimoinen ja työmäärältään ylimitoitettu menetelmä. Lopputuloksena on usein, että aika ei riitä valitun menetelmän käyttämiseen ja siitä täytyy luopua. Menetelmän muutama ensisijainen tehtävä uhrataan liiallisten toissijaisten työmäärän lisääjien vuoksi.

7.3 Hankintasopimuksen ylläpito

Hankintasopimuksen päivittäminen pitkän projektin aikana olisi ollut tarpeellista. Tätä osuutta on arvioitu luvuissa 6.1.1 ja 6.1.2.

SÄHKE-projektin hankintasopimukseseen olisi ollut syytä kirjata sopimuksen muutoskäytännöt. Sopimusta olisi pitänyt päivittää toimittajan kanssa järjestetyissä vuosittaisissa sopimuksen tarkastelutilaisuuksissa, joissa päivitettiin projektin aikataulua ja hinnoittelua [48; 49]. Alkuperäiset kehityshankkeiden työmääräarviot osoittautuivat alimitoitetuiksi, mikä oikeutti toimittajan sopimuksen mukaisiin hinnankorotuksiin. Myös alkuperäisen sopimuksen piiriin kuulumattomien vaatimusten toteutukset vaikuttivat projektin etenemiseen ja kustannuksiin. Muutokset kirjattiin kokousten pöytäkirjoihin, mutta alkuperäisestä sopimuksesta ei tehty päivitysversioita.

7.4 Projektisuunnittelu

Kun SÄHKE-projektilla olisi ollut elinkaarimalli, toimiva vaatimushallinta ja ajantasainen hankintasopimus, olisivat projektisuunnitelman laatimisen ja ylläpidon perusedellytykset olleet olemassa. Projektisuunnittelua on arvioitu luvuissa 6.1.5 ja 6.1.6.

Projektisuunnitelman lähtökohtana ovat vaatimukset, jotka määrittelevät tuotteen ja projektin. SÄHKE-projektin hankintasopimuksessa tehtiin työmääräarvio kuudelle kehittämisalueelle. Mikäli näihin alueisiin kohdistuvat vaatimukset olisi identifioitu, ylimmän tason tehtäväjaottelu luotu ja työmääräarviot kohdennettu tehtäviin, olisi saatu hyvä pohja projektisuunnitelmalle. Varsinaisen vaatimusjaottelun jälkeen tehtävien pilkkominen ja jatkosuunnittelu olisi pitänyt tapahtua toimittajan toimesta ja SÄHKE-projektin tehtävänä olisi ollut toimittajan tekemän suunnitelman etenemisen seuranta.

Lähteet

- [1] Archibald R., State of the Art of Project Management: 2003 [WWW]. Project Management Conference, Escuela Colombiana de Ingeniera, Bogota, Colombia, December 5–6, 2003. PMForum. [viitattu: 5.2.2010]. Saatavissa: <http://www.pmforum.org/library/papers>.
- [2] Boehm, B., A view of 20th and 21st century software engineering, Proceedings of the 28th international conference on Software engineering, May 20-28, 2006, Shanghai, China. New York 2006, ACM Press. pp. 12–29
- [3] McConnell, S., Ohjelmistotuotannon hallinta, Helsinki 2002, IT Press, 638 s.
- [4] Floyd, C., A Systematic Look at Prototyping. In: Budde, R., Kuhlenkamp, K., Mathiassen, L., Ziillighoven, H. Approaches to Prototyping, Springer-Verlag, Berlin, 1984, pp. 1-18
- [5] Martin, J. Rapid Application Development. Indianapolis 1991, Macmillan Publishing Co.. 788 p.
- [6] Coleman, G., Verbruggen, R.. A quality software process for rapid application development. Software Quality Journal 7(1998)2, pp. 107-122
- [7] Abrahamsson, P., Warsta, J., Siponen, M., and Ronkainen, J. New directions in agile methods: Comparative analysis. Proceedings of the 25th International Conference on Software Engineering, May 3–10, 2003, Portland, Oregon. Washington, DC 2003, IEEE Computer Society. pp. 244–254.
- [8] Haikala, I., Märijärvi, J. Ohjelmistotuotanto. 11. painos. Helsinki 2006, Talentum. 440 s.
- [9] Boehm, B. A Spiral Model of Software Development and Enhancement, Computer 21(1988)5, pp. 61-72
- [10] Manifesto for Agile Software Development [WWW], [Viitattu: 23.1.2010]. Saatavissa: <http://www.agilemanifesto.org/>
- [11] Dybå, T., Dingsøy, T. Empirical studies of agile software development: A systematic review. Inform. Softw. Technol. 50(2008)9-10. pp. 833–859
- [12] IEEE Std 1062a-1998. IEEE Recommended Practice for Software Acquisition. New York 1998. IEEE-SA Standards Board. 49 p.
- [13] Hartikainen, K., Kuusisto-Niemi, S., Lehtonen, E. Sosiaali- ja terveydenhuollon tietojärjestelmäkartoitus 2001. Helsinki 2002, Stakes, Osaavien keskusten verkoston julkaisu 1/2002. 101 s.
- [14] Kiviahho, K., Winblad, I., Reponen, J. Terveydenhuollon toimintaprosesseja ja asiointia tukevat atk-sovellukset Suomessa. Helsinki 2004, Stakes, Osaavien keskusten verkoston julkaisu 8/2004. 55 s.

- [15] Winblad, I., Reponen, J., Hämäläinen, P., Kangas, M. Informaatio- ja kommunikaatioteknologian käyttö. Helsinki 2006, Stakes, Stakesin raportteja 7/2006. 99 s.
- [16] Winblad, I., Reponen, J., Hämäläinen, P., Kangas, M. Informaatio- ja kommunikaatioteknologian käyttö Suomen terveydenhuollossa vuonna 2007. Helsinki 2008, Stakes, Stakesin raportteja 37/2008. 140 s.
- [17] Kuusi, O., Ryytänen, O-P., Kinnunen, J., Lammintakanen, J. Terveydenhuollon tulevaisuus. Tulevaisuusvaliokunnan kannanotto vuoden 2015 terveydenhuoltoon. Helsinki 2006, Eduskunnan kanslian julkaisu 3/2006.
- [18] L 9.2.2007/159. Laki sosiaali- ja terveydenhuollon asiakastietojen sähköisestä käsittelystä
- [19] L 2.2.2007/61. Laki sähköisestä lääkemääräyksestä
- [20] Ahlblad, J. Mammuttijärjestelmä ottaa ensiaskeleitaan. Suomen Lääkärilehti 63(2008)24, s. 2206-2208
- [21] Sosiaali- ja terveystieteiden ministeriö 2009. Ministeri Risikko: Terveydenhuollon sähköistä tiedonhallintaa vaiheistettava [WWW]. STM Tiedote 385/2009. [viitattu 21.3.2010]. Saatavissa: <http://www.stm.fi/tiedotteet/tiedote/view/1438004#fi>
- [22] Himmelstein, D., Wright, A., Woolhandler, S. Hospital Computing and the Costs and Quality of Care: A National Study. The American Journal of Medicine 123(2010)1, pp. 40-46
- [23] Haukilahti, R., Virjo, I., Halila, H., Hyppölä, H., Isokoski, M., Kujala, S., Vänskä, J., Mattila, K. Enemmistö lääkäreistä pitää tietoteknologian muutosta myönteisenä. Suomen Lääkärilehti 63(2008)48, s. 4223-4229
- [24] Kortteisto, T., Mäntyranta, T., Komulainen, J., Kaila, M. Lääkäreillä vielä paljon sanottavaa sähköisistä potilaskertomusjärjestelmistä. Suomen Lääkärilehti 63(2008)14, s. 1297-1300
- [25] Kekomäki, M. Tietojärjestelmät ja niiden integroitavuus arvioitava ennen käyttöönottoa. Suomen Lääkärilehti 64(2009)25, s. 2325
- [26] Lahti, H. Kansallinen potilastietojärjestelmä tulee. Premissi (2008)1, s.15-16
- [27] Lääveri, T. Valvooko kukaan? Suomen Lääkärilehti 64(2009)18, s.1643
- [28] Lääveri, T. Ovatko lääkärit tyytyväisiä sähköisiin tietojärjestelmiinsä? Suomen Lääkärilehti 65(2010)5, s.357
- [29] Nykänen, P., Häyrinen, K., Jalonen, M., Lehtinen, A., Lehtonen, J., Miettinen, M., Porrasmäki, J., Virtanen, T. Potilastietojärjestelmäklusterien katselmointi 2010, yhteenvetoraportti 9.2.2010, [viitattu 22.3.2010]. Saatavissa: https://www.kanta.fi/c/document_library/get_file?uuid=145dc302-39fe-4409-b451-c5859f3774fd&groupId=10206
- [30] STM. KANTA – Kokonaisarkkitehtuuri arkkitehtuurimäärittely. 28.2.2006, [viitattu 23.3.2010]. Saatavissa: https://www.kanta.fi/c/document_library/get_file?uuid=0ae54bdd-b3f0-4c95-9fa9-68e8275195c3&groupId=10206
- [31] Sähke-projekti, Väiliraportti 2000. Seinäjoki 2001, EPSHP. Sisäinen raportti. s. 32

- [32] Tarjouspyyntö. Seinäjoki 29.1.2001, EPSHP. Sähköisen potilaskertomusjärjestelmän tarjouspyyntö. s.25
- [33] Sähke-projekti, vuosi 2001. Seinäjoki 2002, EPSHP. Sisäinen raportti. s.42
- [34] ATK-järjestelmän sopimusehdot, sopimuskohteet ja veloitukset. Seinäjoki 10.6.2002, EPSHP. Sähköisen potilaskertomusjärjestelmän hankintasopimuksen liite1, sopimus nro 100211. s.8
- [35] Sähke-projekti, vuosi 2002. Seinäjoki 2003, EPSHP. Sisäinen raportti. s. 16
- [36] Sähke-projekti, vuosi 2003. Seinäjoki 2004, EPSHP. Sisäinen raportti. s. 20
- [37] Sähke-projekti, vuosi 2006. Seinäjoki 2007, EPSHP. Sisäinen raportti. s. 26
- [38] Sähke-projekti, vuosi 2004. Seinäjoki 2005, EPSHP. Sisäinen raportti. s. 16
- [39] Sähke-projekti, vuosi 2007. Seinäjoki 2008, EPSHP. Sisäinen raportti. s. 18
- [40] Sähke-projekti, vuosi 2005. Seinäjoki 2006, EPSHP. Sisäinen raportti. s. 17
- [41] Sähke-projekti, vuosi 2008. Seinäjoki 2009, EPSHP. Sisäinen raportti. s. 22
- [42] Sähköinen sairauskertomus (SÄHKE)-projektin kokous 9/2009. Seinäjoki 25.11.2009, EPSHP. Sisäinen muistio. s. 3
- [43] Sheard, S. Evolution of the Framework's Quagmire. Computer 34(2001)7, pp. 96-98
- [44] Capability Maturity Model for Software, Version 1.1 [WWW]. Software Engineering Institute, Carnegie Mellon University. February 1993. [viitattu 10.2.2010]. Saatavissa: <http://www.sei.cmu.edu/reports/93tr024.pdf>
- [45] CMMI for Acquisition, Version 1.2 [WWW]. Software Engineering Institute, Carnegie Mellon University, November 2007. [viitattu 10.2.2010]. Saatavissa: <http://www.sei.cmu.edu/publications/documents/07.reports/07tr017.html>
- [46] Duncan, S. Making sense of ISO 15504 (and Spice) [WWW], 29.1. 2003, [viitattu 11.3.2010]. Saatavissa: http://www.westfallteam.com/Papers/Making_Sense_of_15504.pdf
- [47] Stenman, Markku. Hankejohtaja, EPSHP. Seinäjoki. Haastattelu 9.-10.12.2009
- [48] Hankintasopimuksen tarkasteluneuvottelu pöytäkirja. Seinäjoki 17.9.2004, EPSHP, SÄHKE-projekti. Sisäinen muistio. s. 2
- [49] Sopimuksen toteutumisen tarkastelutilaisuus, kokousmuistio. Seinäjoki 7.10.2005. EPSHP, SÄHKE-projekti. Sisäinen muistio. s. 4
- [50] Sairauskertomuksen jatkotyöstö 15022005, kokousmuistio. Seinäjoki 1.2.2005. EPSHP, SÄHKE-projekti. Sisäinen muistio. s. 15
- [51] Juopperi, K. Effica v.3.5.01 käyttöönotto, loppuraportti. 14.1.2009, Tieto Oyj. Sisäinen raportti. s.17
- [52] Effica järjestelmien tarjousten yhteinen osuus. Oulu 10.6.2002, Tieto Enator. Sähköisen potilaskertomusjärjestelmän hankintasopimuksen liite 6, sopimus nro 100211. s.23
- [53] Juopperi, K. Näkökulmia sähköisen potilaskertomuksen käyttöönotosta EPSHP:ssä. 22.3.2010, Tieto Oyj. Sähköposti. s.3
- [54] Räätäri, Kati. Johtaja, Tieto Oyj. Juopperi Kimmo. Projektipäällikkö, Tieto Oyj. Puhelinhaastattelu 22.3.2010

- [55] IEEE Std 830-1998. IEEE Recommended Practice for Software Requirements Specifications. New York 1998. IEEE-SA Standards Board. 39 p.