



LAUREA
AMMATTIKORKEAKOULU

Uuden edellä

Tietojärjestelmän kehittämisprojekti

Case: Prime Solutions Oy

Husu, Ilari

2012 Laurea Kerava

Laurea-ammattikorkeakoulu
Laurea Kerava

Tietojärjestelmän kehittämisprojekti
Case: Prime Solutions Oy

Husu, Ilari
Tietojenkäsittelyn koulutusohjelma
Opinnäytetyö
Toukokuu, 2012

Ilari Husu

Tietojärjestelmän kehittämisprojekti - Case: Prime Solutions Oy

Vuosi 2012 Sivumäärä 38

Tietojärjestelmän kehittämisen vaiheita ovat esitutkimus, määrittely, suunnittelu, toteutus, testaus, käyttöönotto ja ylläpito. Näistä vaiheista jokainen on tärkeä osa onnistunutta tietojärjestelmän kehittämisprojektia. Tässä opinnäytetyössä syvennytään näistä kolmeen, eli suunnitteluun, toteutukseen ja testaukseen.

Työ jakautuu teoriaosuuteen ja käytännön työhön. Teoriaosuus koostuu kehittämisvaiheiden tutkimisesta, käytännön osuus keskittyy puolestaan enemmän tietojärjestelmän kehittämisprojektin suunnittelu-, toteuttamis- ja testausvaiheisiin. Tämän tutkimuksen päätavoitteena oli löytää uusia, entistä tehokkaampia toimintatapoja tietojärjestelmän kehittämisprojekteihin.

Opinnäytetyö toteutettiin käytännön työnä, jonka seurauksena opinnäytetyössä käsitellystä kehittämisprojektista valmistui terveydenhuoltoon suunnattu TUVATAR-tietojärjestelmä. Tietojärjestelmä suunniteltiin yhteistyössä asiakasyrityksen kanssa ja se toteutettiin Java-ohjelmointikielellä. Java on yksi yleisimmistä tietojärjestelmien toteutuksessa käytetyistä ohjelmointikielistä.

TUVATAR-tietojärjestelmä on kehitetty mielenterveysasiakkaiden toimintakyvyn arviointiin. Projektin tavoitteena oli luoda tietojärjestelmä, jonka avulla helpotetaan mielenterveysasiakkaiden arviointia sekä kerättyjen tietojen analysointia. Tietojärjestelmän avulla kerätään asiakasta koskevat tiedot yhteen valtakunnalliseen tietokantaan. Tietojärjestelmä otettiin käyttöön syksyllä 2011.

Tutkimuksen avulla selvisi, että tehokkaan tietojärjestelmän suunnitteluvaiheeseen pitäisi kuulua järjestelmän tekninen määrittely, jonka avulla seuraavat vaiheet, erityisesti toteutusvaihe, on tehokkaammin ja nopeammin suoritettavissa. Toteutuksen jälkeen suoritetaan testaus, jolla varmistetaan ja parannetaan vielä kehitettävän järjestelmän laatua.

Asiasanat: tietojärjestelmä, suunnittelu, toteutus, testaus, Java

Ilari Husu

Information System Development Project - Case: Prime Solutions Ltd

| Year | 2012 | Pages | 38 |
|------|------|-------|----|
|------|------|-------|----|

The phases of an information system development project include the feasible study, design, specification, implementation, testing, development and maintenance. Each of these phases is an important part of a successful information system development project. The objective of this Bachelor's study was to examine different phases of an information system development project, focusing mainly on designing, implementation and testing phases.

The study is divided into theoretical and in empirical parts. The theoretical part concentrates on examining the different stages of the designing phase, while the empirical part is more concentrated on the implementation, testing and occurring challenges of an information system development project. The main objective was to find new, more efficient ways of working in information system development projects.

The study was conducted as a case study, and as a result, an information system development project directed for the public health, TUVATAR, care was created. The project was executed in cooperation with a customer company and with the programming language of Java. Java is one of the most popular programming languages used in information system development projects.

The TUVATAR information system was developed for evaluating the condition of mental health patients. The objective was to create an information system that facilitates the evaluation and information analysis concerning the patients. The information system collects all information concerning one patient to a nationwide system and it was taken in use in autumn 2011.

The main finding of this study was that the designing phase should include the technical specifications, would enable easier and faster execution of the implementation phase. The testing phase followed by implementation would ensure functionality and would give ideas for the improvement project.

Key words: information system, designing, implementing, testing, Java

Sisällys

| | | |
|----|--|----|
| 1 | Johdanto..... | 6 |
| 2 | Tietojärjestelmät yleisesti | 7 |
| 3 | Tiedon merkitys terveydenhuollolle | 7 |
| 4 | Tietojärjestelmien kehittäminen | 7 |
| | 4.1 Kehittämisprojektin roolit | 9 |
| 5 | Tietojärjestelmän suunnittelu..... | 9 |
| | 5.1 Arkkitehtuurisuunnittelu..... | 11 |
| | 5.2 Moduulisuunnittelu..... | 12 |
| 6 | Tietojärjestelmän toteutus | 12 |
| | 6.1 Komponenttien uudelleenkäyttö | 13 |
| | 6.2 Toteutustekniikat, ohjelmistot ja välineet | 14 |
| | 6.2.1 Java..... | 14 |
| | 6.2.2 Apache Struts | 15 |
| | 6.2.3 JavaServer Pages | 15 |
| | 6.2.4 MySQL | 15 |
| | 6.2.5 Java Database Connectivity | 15 |
| | 6.2.6 NetBeans IDE | 15 |
| | 6.2.7 Apache Tomcat | 16 |
| | 6.2.8 JFreeReport | 16 |
| | 6.2.9 XML | 16 |
| | 6.3 Versionhallinta..... | 17 |
| 7 | Tietojärjestelmän testaus..... | 18 |
| | 7.1 Testitapaukset ja menetelmät..... | 20 |
| | 7.1.1 Lasilaatikkotestaus..... | 20 |
| | 7.1.2 Mustalaatikkotestaus | 21 |
| | 7.2 Bugzilla..... | 21 |
| 8 | TUVATAR-tietojärjestelmä | 22 |
| 9 | Tietojärjestelmän toimintaympäristö | 25 |
| 10 | Kehittämisprojektin ongelmat ja ratkaisut..... | 25 |
| 11 | Kehittämisehdotukset | 26 |
| 12 | Tulokset..... | 26 |
| 13 | Johtopäätökset | 27 |
| | Lähteet | 28 |
| | Kuvat | 29 |
| | Kuviot | 30 |
| | Taulukot | 31 |
| | Liitteet..... | 32 |

1 Johdanto

Opinnäytetyössäni tutkin tietojärjestelmän suunnittelua, toteutusta ja testausta. Työni tavoitteena oli kehittää tietojärjestelmän kehittämisprojektin ensimmäinen versio. Tavoitteena oli myös kuvata tarkasti tietojärjestelmän kehittämisen eri vaiheita ja menetelmiä. Pysin myös arvioimaan tekniikoiden, vaiheiden ja menetelmien pätevyyttä toimia onnistuneessa kehittämisprojektissa. Opinnäytetyöni on muodoltaan toimintakeskeinen opinnäytetyö.

Kehittämisprojektista syntyi TUVATAR-tietojärjestelmä. TUVATAR-tietojärjestelmä on mielen-terveyskuntoutujille suunnattu kirjaus ja seurantajärjestelmä. Tietojärjestelmä kehitettiin toimimaan vanhan menetelmän korvaajana ja sitä tullaan kehittämään asteittain. Tämä tarkoittaa, että kaikkia suunniteltuja ominaisuuksia ei tehty vielä ensimmäiseen versioon, vaan ensimmäisen version käyttöönoton jälkeen tietojärjestelmään tullaan kehittämään uusia ominaisuuksia. Kehittämisprojektista syntynyt tietojärjestelmä on suunniteltu laitoksien, yhdistyksien ja säätiöiden käyttöön. Käytännön projektia tehtiin vuoden 2011 aikana. Projektin työt aloitettiin keväällä ja ne saatiin päätökseen syksyllä.

Kehittämisprojektin tietojärjestelmä kehitettiin toisen järjestelmän pohjalta, joten tutkin myös toteutuksessa käytettyjen erilaisten komponenttien uudelleenkäyttöä ja näiden vaikutusta tietojärjestelmän kehittämisen eri vaiheisiin. Koska tietojärjestelmän kehittämisessä on erilaisia vaiheita ja tehtäviä, vaikuttaa jokaisen vaiheen lopputulos seuraaviin kehittämisvaiheisiin.

Opinnäytetyöni toimeksiantajana toimi Prime Solutions Oy. Yritys on erikoistunut IT-alan asiantuntija- ja projektipäällikkötehtäviin. Yritys on suomalainen ja sen tavoitteena on parantaa projektien ja IT-kehityshankkeiden tuottavuutta ja mahdollistaa uusien teknologioiden käyttö kytkemällä ne olemassa oleviin järjestelmiin. Opinnäytetyöni käytännön kehitysprojekti suoritettiin Prime Solutionsin ohjelmistoilla ja laitteilla. Yrityksen toimitilat sijaitsevat Helsingin Malmilla kauppakeskus Malmin Novassa.

2 Tietojärjestelmät yleisesti

Vuosien varrella tietojärjestelmät ovat tulleet osaksi monen yrityksen arkea. Tietojärjestelmiä hyödynnetään lähes jokaisella alalla ja moneen erilaiseen tarkoitukseen. Nykypäivän yrityksissä tietojärjestelmien vaikutukset näkyvät toiminnan tehokkuudessa ja paremmassa laadussa. Tietojärjestelmä on käsitteenä todella laaja ja se ei välttämättä ole edes tietokoneohjelma, jollaiseksi sitä usein kuvaillaan.

Tietojärjestelmien tarkoitus on käsitellä tietoja ja tämän avulla tehostaa tai helpottaa jotakin tiettyä toimintaa tai tehdä tämä toiminta mahdolliseksi. Tietojärjestelmä koostuu koulutetuista ihmisistä, tietojenkäsittelylaitteista, tiedostonsiirtolaitteista ja ohjelmistoista. Käsitteenä tietojärjestelmällä viitataan ohjelmistoja suurempaan kokonaisuuteen. Pelkkä ohjelmisto on käsitteenä suppea ja sillä tarkoitetaan tiettyyn koneeseen tai tehtäväalueeseen liittyviä ohjelmia, kuten tekstinkäsittely- ja taulukkolaskentaohjelmat. (Pohjoinen 2002, 5-10.)

3 Tiedon merkitys terveydenhuollolle

Terveydenhuollon alalla tarkan tiedon merkitys on suuri. Tämän vuoksi terveydenhuollon tietojärjestelmien on annettava luotettavaa ja tehokasta tietoa käyttäjänä toimiville hoitajille ja lääkäreille. Uudet teknologiat mahdollistavat ongelmien ratkaisun useisiin terveydenhuollon ongelmiin. Tietojärjestelmien hankinnassa on kuitenkin mietittävä tarkkaan kustannuksia ja järjestelmän palvelevuutta tietyn toiminnan tehostamista kohti. Tietojärjestelmien ominaisuuksia kannattaa kuitenkin käyttää aina, jos niistä on apua hoitotyöhön.

Terveydenhuollon tietojärjestelmien kehittämisessä on monenlaisia haasteita. Vaikka terveydenhuollon yrityksillä on suuri tarve useille erilaisille järjestelmille, on organisaatioilla käytössään usein vanhat paperiset menetelmät. Terveydenhuollon alan monimutkaisuus tuo haasteita niin ostajalle kuin toimittajillekin. Monille terveydenhuollon pienyrityksille on myös hankalaa hankkia tarpeisiinsa sopivaa monimutkaista järjestelmää, koska usein monimutkaisten järjestelmien kustannukset ovat suuret. (Glaser, Lee & Wager 2009, 22.)

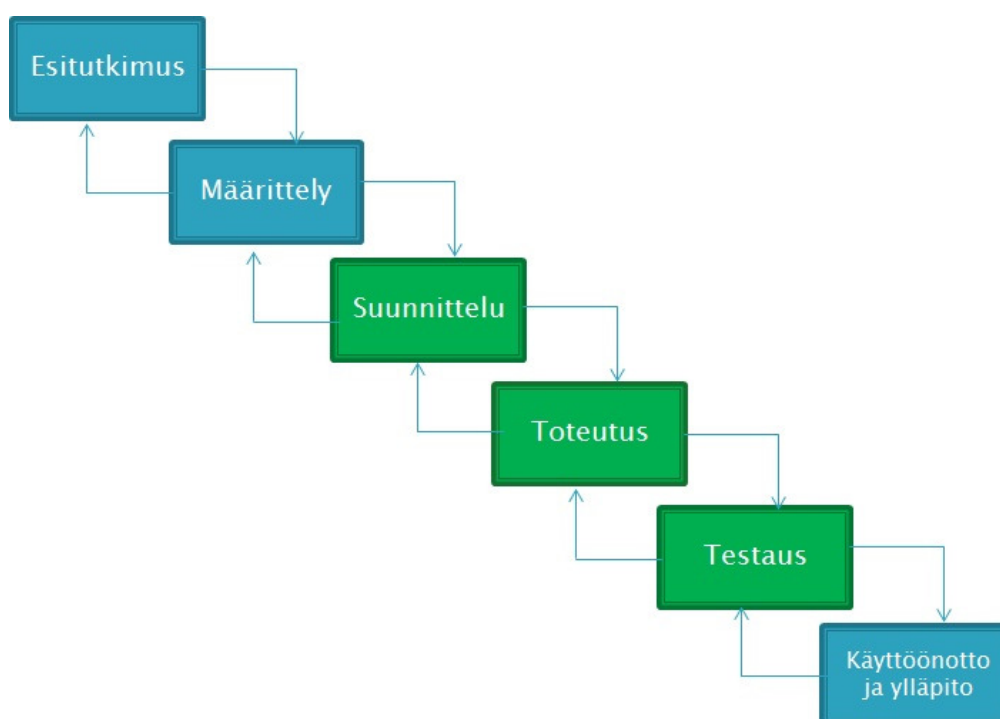
4 Tietojärjestelmien kehittäminen

Tietojärjestelmien kehittäminen aloitetaan usein yrityksen tai asiakkaan tarpeesta. Tarpeet syntyvät uusien teknisten mahdollisuuksien myötä, kehittämispaineiden vuoksi tai kartoitus-työn perusteella. Tietojärjestelmän kehitystyö on järjestelmällistä toimintaa, jossa tietyt kehittämisenvaiheet edeltävät toisia vaiheita. Nämä vaiheet yhdessä muodostavat tietojärjestelmän elinkaaren. Elinkaaren vaiheilla pyritään määrittelemään tietojärjestelmän kehittämisen tehtävät, niiden ajoitus ja riippuvuus toisistaan. Tietojärjestelmän elinkaareen kuuluvien

vaiheiden käsitykset vaihtelevat hieman, mutta yleisesti mielletään kuitenkin elinkaaren alkavan esitutkimuksella.

Tietojärjestelmän kehittämisen elinkaari sisältää erilaisia vaiheita. Elinkaari alkaa esitutkimuksella, jota seuraa vaatimusmäärittely. Tämän jälkeen suoritetaan tietojärjestelmän suunnittelu-, toteutus- ja testausvaihe. Hyväksytyyn testauksen jälkeen tietojärjestelmä käyttöön otetaan ja sen ylläpitovaihe aloitetaan. Ylläpitovaihe on usein kehittämisenvaiheista pitkäkestoisin, koska se jatkuu siihen saakka kunnes järjestelmästä luovutaan. (Pohjonen 2002, 27.)

Usein uskotaan, että tietojärjestelmän kehittämisprojektit kestävät niin kauan, kunnes tietojärjestelmän jokainen vaihe on suoritettu perusteellisesti ja tietojärjestelmä on laadultaan paras mahdollinen. Käytännössä tämä on kuitenkin mahdotonta. Useissa projekteissa joudutaan tuhoamaan projektin vaiheistus kokonaan ajan puutteen takia. Mitä pienempi ja yksinkertaisempi projekti on, sen vähemmän siinä on riskejä. Hyvin johdetun projektin ei kuitenkaan pitäisi viivästyä käyttöönoton ajankohdasta, sillä tietojärjestelmälle on asetettu tarkat laatuvaatimukset, joiden mukaan jokainen tietojärjestelmän kehittämisen vaihe suoritetaan. Projektin aikana ilmenee aina ongelmia, joten niiden ratkomiseen kuluva aika on huomioitava aikataulussa (Lehtimäki 2006, 155). Kuviossa 1 havainnollistan kehittämisprojektin työvaiheiden kulkua vesiputousmallin avulla.



Kuvio 1: Tietojärjestelmän kehittämisen vaiheet (Haikala & Mikkonen, 2011).

Vesiputousmallin avulla kuvataan usein tietojärjestelmän kehittämisen vaiheita. Suurin työ tietojärjestelmän elinkaarissa alkaa usein kuitenkin vasta tietojärjestelmän käyttöönoton

jälkeen, uusien kehittämiskohteiden ja teknologian kehityksen kautta. Monissa järjestelmissä huomataan käyttöönoton jälkeen puutteita sen palvelevuudessa, joten uusia ominaisuuksia kehitetään jatkuvasti.

4.1 Kehittämisprojektin roolit

Usein uskotaan, että tietojärjestelmien kehittämiseen osallistuu vain IT-alan ammattilaisia, mutta tämä ei pidä paikkaansa. Tietojärjestelmät ovat lähes joka alan organisaatioissa käytössä ja kyseisten ammattilaisten on toimittava osana kehittämisprojektin kehitysryhmää. Tietojärjestelmien on palveltava niille osoitettua käyttötarkoitustaan mahdollisimman hyvin, joten myös kehittäjien on tarkkaan tiedettävä tietojärjestelmän tarkoitus organisaation toiminnassa ja sen kehittämisessä. Myös tulevien käyttäjien on osallistuttava tietojärjestelmien kehittämiseen. Käyttäjät, jotka tulevat käyttämään tietojärjestelmää päivittäin, tietävät hyvin tietojärjestelmän tarkoituksen heidän organisaatiossaan. Paras informaation lähde tietojärjestelmää määriteltäessä onkin käyttäjien mielipiteet (Pohjonen 2002, 47).

Kehittäjät jaetaan yleensä kolmeen ryhmään: ohjelmoijiin, suunnittelijoihin ja määrittelijöihin. Suurissa projekteissa kyseinen jako usein toimii mutta pienissä yksittäinen kehittäjä voi toimia jokaisessa työvaiheessa. Tarkasteltavan kehittämisprojektin aikana toimin suunnittelijan ja ohjelmoijan tehtävissä. Vaikka kehitettävä tietojärjestelmä olikin laaja, komponenttien uudelleenkäyttö mahdollisti näiden kahden eri työtehtävän hoitamisen.

5 Tietojärjestelmän suunnittelu

Tietojärjestelmän määrittelyvaiheen jälkeen siirrytään suunnitteluvaiheeseen. Tietojärjestelmän suunnittelun tarkoituksena on suunnitella järjestelmän toteutus. Asiakkaan tarpeiden mukaisesti tehty toiminnallinen määrittely muunnetaan tekniseksi määrittelyksi. Järjestelmä toteutetaan suunnittelusta syntyneen teknisen määrittelyn mukaisesti. Suunnitteluvaihe jaetaan usein arkkitehtuuri- ja moduulisuunnitteluun.

Suunnitteluvaiheesta syntyneen teknisen määrittelyn tulisi sisältää kuvauksen seuraavista asioista:

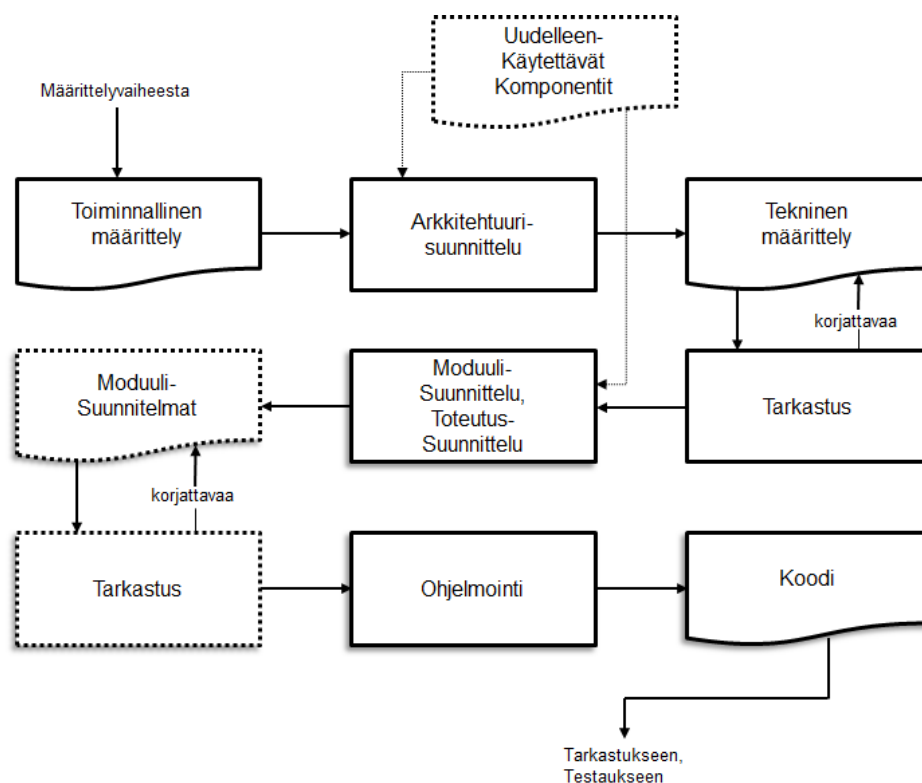
- järjestelmän tarkoitus,
- ohjelmisto- ja laiteympäristö,
- keskeiset reunaehdot toteutuksessa,
- ympäristön ja järjestelmän välinen vuorovaikutus,
- arkkitehtuuri (tietokanta- ja ohjelmistoarkkitehtuuri, uudelleen käytettävät komponentit),

- moduulit ja alijärjestelmät,
- erityiset tekniset ratkaisut,
- hylätyt ja vaihtoehtoiset ratkaisut ja
- muut toteutukseen vaikuttavat seikat.

(Pohjonen 2002, 32.)

Tietojärjestelmien toteutuksessa on usein ongelmana ero sille asetettujen vaatimusten ja käytettävissä olevien teknologioiden välillä. Suunnittelun avulla pyritään löytämään parhaat teknologiset ratkaisut kyseiseen ongelmaan. Suunnitteluvaiheessa saadaan tietojärjestelmään liittyvät tekniset ongelmat helposti selville ja ne pystytään ratkaisemaan ennen toteutusvaiheeseen siirtymistä. Hyvin suoritettuna suunnitteluvaiheen perusteella toteutus on suoraviivaista ja helppoa.

Suunnitteluvaiheessa kommunikoinnilla ja oppimisella on myös suuri merkitys. Suunnittelijan on pystyttävä kommunikoimaan projektin muiden suunnittelijoiden kanssa hyvin, jotta yhden suunnittelijan nerokkaana pidetyt ratkaisut eivät rasita suunnitteluvaihetta. Kommunikointi ei ole onnistunut, jos projektin muilla suunnittelijoilla menee rutkasti aikaa yhden suunnittelijan ratkaisun ymmärtämiseen. Suunnittelun luonnollisia tavoitteita ovat ymmärrettävyys, selkeys, luotettavuus, tehokkuus, ylläpidettävyys ja siirrettävyys. (Haikala & Mikkonen 2011, 177-178.) Kuviossa 2 havainnollistetaan suunnittelun työtehtävien kulkua.



Kuvio 2: Suunnitteluprosessi (Haikala & Märijärvi, 1995).

Suunnittelu on pitkälle kehittynyt prosessi. Järjestelmän elinkaaren tutkiminen helpottaa koko suunnitteluprosessin ymmärtämistä. Elinkaari pitää sisällään suunnitteluprosessin kaikki vaiheet tarpeesta tai ongelman havaitsemisesta valmiiseen tietojärjestelmään. Suunnittelussa on otettava huomioon tekniset ongelmat, strategiset vaikutuksen, kustannukset ja työn aiheuttamat muutokset. Valittaessa menetelmää tietojärjestelmien suunnitteluun on muistettava, että erilaisia menetelmiä käytetään erilaisissa tilanteissa. Rakennettaessa suuria, rakenteeltaan tarkasti määrättyjä tietojärjestelmiä käytetään muodollisempia suunnittelumenetelmiä. Suunnittelumenetelmästä riippumatta on aina muistettava, että tietojärjestelmä rakennetaan ratkaisemaan oikeaa ongelmaa. Järjestelmän täytyy sopia olemassa olevaan ympäristöön ja tulevien käyttäjien työtehtäviin (Pohjonen 2002, 32).

Suunnitteluvaiheessa huomasin joidenkin toimintojen määrittelyiden olleen hieman suppeita. Pystyimme ratkaisemaan nämä ongelmat jo suunnitteluvaiheessa, joten ongelmat eivät seuranneet mukana toteutusvaiheeseen, jolloin ne olisi ollut vaikeampi korjata. Kehitimme tietojärjestelmän toisen järjestelmän pohjalta, ja uudelleen käytettäviä komponentteja löytyi paljon. Muokkasin niitä kuitenkin hieman uusien ominaisuuksien mukaan.

Suunnittelin myös tietojärjestelmän toimintaympäristöt. Tietojärjestelmälle suunnitellut ympäristöt olivat testi-, koulutus- ja tuotantoympäristöt. Testiympäristön avulla jokainen kehittämisprojektin jäsen voi testata tuotantoon menevää versiota. Koulutusympäristöä käytetään nimensä mukaisesti käyttäjien koulutukseen. Asiakasyritys kouluttaa käyttäjät tarvittaessa. Ehtona ympäristöille oli, että koulutus- ja tuotantoympäristöt sisältävät vain huolellisesti testatut ja hyväksytyt tietojärjestelmän versiot.

5.1 Arkkitehtuurisuunnittelu

Järjestelmän yleinen rakenne määritellään arkkitehtuurisuunnittelun avulla. Arkkitehtuurisuunnittelun tarkoituksena on jakaa järjestelmä pieniin osiin, jotka voidaan jakaa yksittäisien kehittäjien suunniteltaviksi ja toteutettaviksi. Nämä pienet osat ovat moduuleita, joiden tehtävät määritellään erikseen modulisunnittelun avulla (Pohjonen 2002, 32).

Ihmisen on hankala hallita koko järjestelmää sen monimutkaisuuden vuoksi. Arkkitehtuurisuunnittelun avulla pyritään helpottamaan järjestelmän monimutkaisuuden hallitsemista. Tuloksena arkkitehtuurisuunnittelusta syntyy tekninen määrittely. Toteutuksen pohjana toimiva tekninen määrittely on oltava hyvin palveleva. Arkkitehtuurisuunnitelman ohjelmistorakenteen määrittelyssä esiintyvät virheet ovat kriittisiä, koska ne voivat tehdä tietojärjestelmästä käyttökelvottoman (Haikala & Märijärvi 1995. 61-63).

Arkkitehtuurisuunnittelu on tärkeä osa tietojärjestelmän kehittämistä, mutta se voi olla riehana kehittämisprojektille. Koko kehittämisprojekti voi halvaantua, jos suunnitteluvaiheesta ei edetä. Organisaatioilla ja niiden henkilöstöllä on erilaisia näkökulmia suunnitteluvaiheen kestosta. Jokaiselle suunnittelun vaiheelle onkin hyvä asettaa kehittämisprojektin alkaessa laadulliset tavoitteet. Kun tavoitteet on laadittu, voidaan suunnittelun saavuttaessa tavoite-laadun siirtyä toteutusvaiheeseen. Näin estetään projektin suunnitteluvaiheen ja etenkin arkkitehtuurisuunnittelun aikana mahdollisesti tapahtuvan halvaantumisen (Lehtimäki 2006, 97).

5.2 Moduulisuunnittelu

Kun järjestelmä on arkkitehtuurisuunnittelun avulla jaettu pieniin osiin, siirrytään moduulisuunnitteluun. Moduulisuunnittelussa suunnitellaan jokaisen komponentin sisäinen rakenne. Suunnittelussa on pyrittävä mahdollisimman pieneen moduulikokoon, koska suuret moduulit ovat monimutkaisempia ja hankalampia ylläpitää (Pohjonen 2002, 33).

Usein moduulisuunnittelua ei pidetä erillisenä suunnitteluvaiheena, vaan yksittäisen moduulin suunnitteleminen, toteuttaminen ja testaaminen käsitetään yhdeksi tehtäväksi (Haikala & Märijärvi 1995. 61-63).

6 Tietojärjestelmän toteutus

Suunnitteluvaiheesta seuraa toteutus. Toteutusvaiheessa tietojärjestelmä tai sen komponentti toteutetaan esimerkiksi jollakin ohjelmointikielillä tai sovelluskehittimellä. Toteutuksen kulkuun vaikuttaa suunnittelusta valmistunut tekninen määrittely. Mikäli tekninen määrittely on tehty hyvin, pitäisi toteutuksen olla melko suoraviivainen toimenpide, koska kaikki tietojärjestelmän toiminnallisuutta ja rakennetta koskevat ratkaisut on jo tehty.

Toteutusteknologian ja ohjelmointikielen onnistunut valinta ei vielä takaa onnistumista. Tietojärjestelmän toteutuksen on myös vastattava järjestelmälle asetettuja vaatimuksia ja sen on oltava teknisen ja toiminnallisen määrittelyn mukainen. Toteutuksen onnistumiseen vaikuttavat paljon aiemmissa vaiheissa tehdyt ratkaisut ja niiden kelpoisuus. Jos aikaisemmissa vaiheissa on tehty huonoja ratkaisuja, ei toteutus ole kuitenkaan vielä epäonnistunut. Toteutuksen onnistumisen takaamiseksi on kyettävä hallitsemaan nämä uudet esille tulleet vaatimukset. (Pohjonen 2002, 33.)

Toteutusvaihe sujui kehittämisprojektissa hyvin, ilman suurempia ongelmia. Hyvän suunnitteluvaiheen tuloksena valmistui tekninen määrittely, jonka mukaan järjestelmä toteutettiin. Tietojärjestelmä on toiminnoiltaan samanlaisen tietojärjestelmän pohjalta suunniteltu, joten ohjelmointikielen valinta sopi mainiosti kehitettävään tietojärjestelmään. Testausvaiheesta

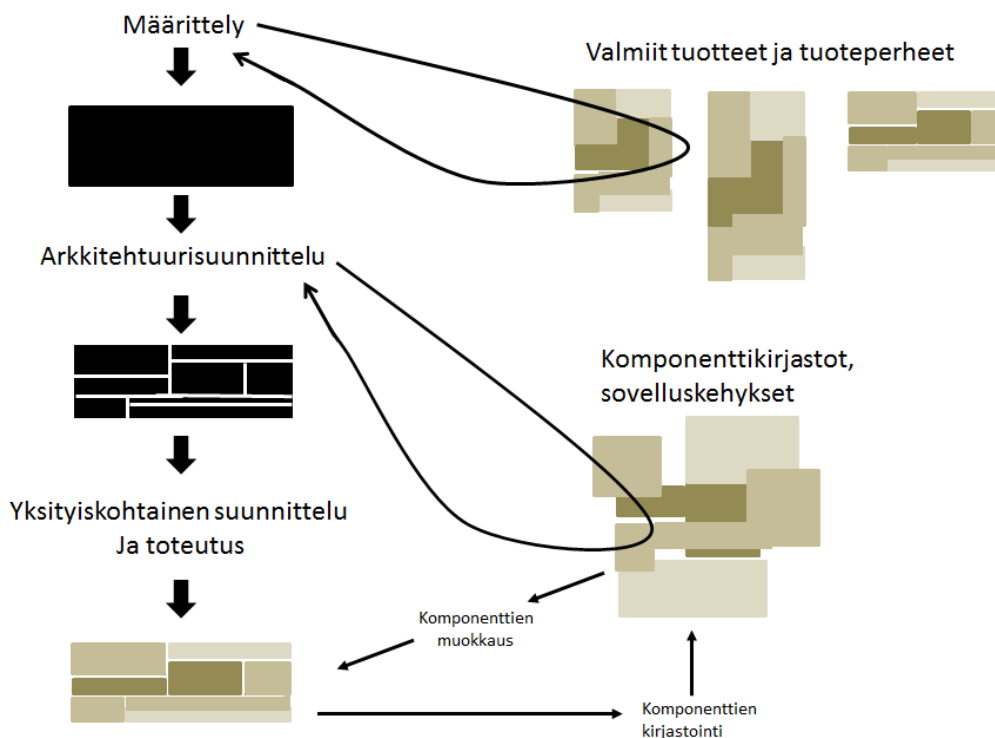
palattiin toteutusvaiheeseen korjaamaan löydettyjä virheitä. Virheitä ei kuitenkaan löytynyt tietojärjestelmän rakenteesta, joten korjattavat virheet olivat pieniä, yksittäisen komponentin korjauksia.

6.1 Komponenttien uudelleenkäyttö

Tietojärjestelmien kehittämissuhteissa käytetään usein komponentteja uudelleen. Komponentit ovat usein samassa organisaatiossa kehitettyjä. Uskotaan, että maailmalla tehtävistä ohjelmistoista 60-80 % on tehty jo aikaisemmin. Uudelleenkäyttö ei koske pelkästään komponentteja, vaan kehittämissuhteiden eri vaiheita voidaan käyttää uudestaan. Esimerkiksi tietojärjestelmän määrittelyn pohjaksi voidaan ottaa vanhasta samankaltaisesta projektista syntynyt dokumentti. Yleisin muoto on kuitenkin komponenttitason uudelleenkäyttö.

Uudelleenkäytettävä komponentti muokataan sopivaksi käyttötarkoitukseensa. Yleisimpiä menetelmiä on ottaa toteutuksen pohjaksi vanha versio ja muokata tämä uusiin käyttötarkoituksiin sopivaksi. On väitetty, että komponenttien uudelleenkäyttö lisäävät tuottavuutta ja parantavat kehitettävän tietojärjestelmän laatua. Tuottavuus kasvaa uutta tietojärjestelmää tehdessä, kun kaikkia sen komponentteja ei tarvitse tehdä kokonaan uudelleen. Vanhat uudelleen käytettävät komponentit ovat yleensä virheettömiä, joten testaukseen ja virheiden korjaukseen ei tarvitse tuhata paljon aikaa. (Haikala & Mikkonen 2011, 190-191.)

Tietojärjestelmä kehitettiin toisen samankaltaisen tietojärjestelmän pohjalta, joten toteutuksessa uudelleenkäytettiin toisen järjestelmän erilaisia komponentteja. Tämän ansiosta saimme kehittämissuhteiden valmiiksi huomattavasti nopeammin. Komponenttien uudelleenkäyttö näkyi selvästi testauksessa. Testausprosessi helpottui tämän ansiosta huomattavasti. Kuvio 3 osoittaa uudelleenkäytön mahdollisuuksia erilaisissa työvaiheissa.



Kuvio 3: Uudelleenkäyttö tuotantoprosessissa (Haikala & Mikkonen, 2011).

6.2 Toteutustekniikat, ohjelmistot ja välineet

Tietojärjestelmän suunnitteluvaiheessa on valittu ohjelmointikieliet, ohjelmistokehykset ja tekniikat, joiden avulla järjestelmä kehitetään. Valitsimme toteutuskieleksi Javan, koska se on yksi yleisimmistä tietojärjestelmien toteutuksissa käytetyistä ohjelmointikielistä. Järjestelmä hyödyntää Struts-ohjelmistokehystä ja kommunikoi MySQL-tietokannan kanssa. MySQL on yksi yleisimmistä tietokantaohjelmistoista tietojärjestelmien datan varastointiin. Projektin aikana käytimme useita avoimeen lähdekoodiin perustuvia ohjelmistoja.

6.2.1 Java

Java on järjestelmäriippumaton oliopohjainen ohjelmointikieli. Syntaksiltaan Java muistuttaa paljon C++-kieltä, mutta niissä on myös paljon eroja. Javan parhaimpia ominaisuuksia ovat sen järjestelmäriippumattomuus, jonka ansiosta Java-ohjelmia voidaan kehittää ja testata eri käyttöjärjestelmissä. Esimerkiksi sama koodi voidaan suorittaa Linuxissa ja Windowsissa. Käyttöympäristöjä on paljon ja niiden määrä kasvaa jatkuvasti. Javalla voi ohjelmoida esimerkiksi puhelimia ja digitelevisioita (Kosonen, Peltomäki & Silander 2008, 2).

6.2.2 Apache Struts

Struts on web-sovelluksien luontiin tarkoitettu ohjelmistokehys. Strutsin avulla eri osa-alueiden osajat, esimerkiksi tietokantasuunnittelijat, web-suunnittelijat ja ohjelmistosuunnittelijat, voivat työskennellä saman projektin parissa yhtäaikaisesti. Tämän ansiosta tietojärjestelmän toteuttaminen tehostuu ja toimiva järjestelmä saadaan käyttöönotettua lyhyessä ajassa. Struts on hyvin tunnettu ohjelmistokehys maailmassa joten siitä löytyy ongelmatilanteissa myös helposti tietoa (Doray 2006, 5).

6.2.3 JavaServer Pages

JavaServer Pages (JSP) -tekniikan avulla voidaan luoda dynaamisia web-sivuja. JSP hyödyntää Servlet-rajapintaa ja sen avulla voidaan helposti upottaa Java-koodia web-sovelluksiin. JSP on riippuvainen servleteistä, koska ne kääntävät JSP-koodin lähdekoodiksi. JSP-sivut muutetaan servleteiksi palvelimesta riippuen asennettaessa tai ensimmäisellä käyttökerralla (Kosonen ym. 2008, 478).

6.2.4 MySQL

MySQL on yksi maailman suosituimmista avoimen lähdekoodin tietokantaohjelmistoista. MySQL on saatavilla monelle eri käyttöjärjestelmälle ja se on tunnettu toimivuudestaan, nopeudestaan, pienistä kustannuksista ja siirrettävyydestään. MySQL:n nykyinen omistaja on Oracle, joka tunnetaan maailman suurimpana tietokantaratkaisujen toimittajana (Meloni 2003, 11-15).

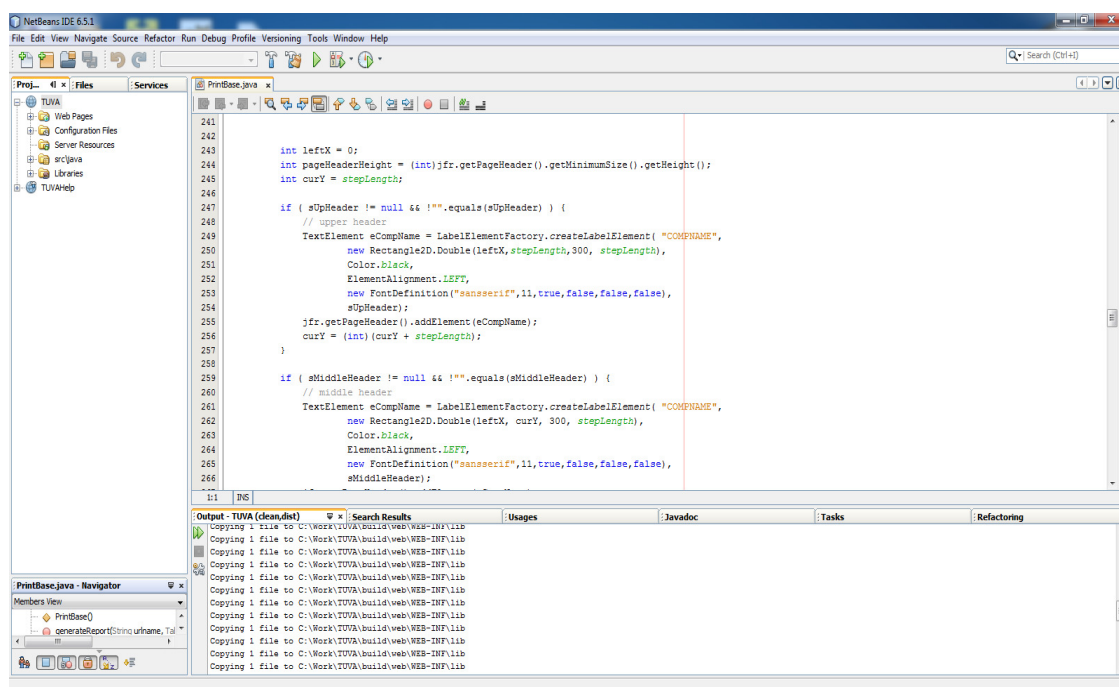
6.2.5 Java Database Connectivity

Java Database Connectivity (JDBC) on suosittu järjestelmä- ja tietokantariippumaton rajapinta. JDBC:n avulla ohjelmoija pystyy käyttämään tietokantoja SQL-kielellä. Rajapinta toimii erilaisilla SQL-kielillä, jos niihin on kehitetty ajurit. Verrattuna tietokantarajapintoihin, jotka toimivat vain yhden valmistajan tuotteiden välillä on JDBC hyvin yleiskäyttöinen rajapinta sen tietokantariippumattomuuden vuoksi. (Kosonen ym. 2008, 530-531).

6.2.6 NetBeans IDE

NetBeans kehitysympäristö on Java-ohjelmointikielellä toteutettu ilmainen ohjelmointityökalu. Kehitysympäristöllä voidaan kirjoittaa, kääntää, debugata ja ottaa käyttöön sovelluksia.

Netbeans tukee mitä tahansa ohjelmointikieltä ja se on laajennettavissa siihen ladattavilla useilla eri moduuleilla. Kuvassa 1 havainnollistan Netbeans 6.5.1 kehitysympäristön ulkoasua.



Kuva 1: Kuvakaappaus Netbeans 6.5.1 kehitysympäristöstä.

6.2.7 Apache Tomcat

Tomcat on yksi käytetyimmistä Java web-sovellusten sovelluspalvelimista, mutta se ei kuitenkaan ole täysimittainen Java EE-palvelin. Tomcat toteuttaa ainoastaan Servlet säiliön, joten siinä on mahdollista ajaa Servlettejä sekä JSP-sivuja. Tomcatin ja erillisen EJB-säiliön kanssa siinä voidaan suorittaa myös EJB-komponentteja. Näin siitä saadaan täysimittainen Java EE-palvelin (Kuha 2008, 376).

6.2.8 JFreeReport

JFreeReport on ilmainen Java-luokkakirjasto raporttien luontiin. JFreeReportin avulla voidaan raportit helposti muotoilla halutun näköiseksi. Raporttien muotoilu ja tiedot haetaan erillisistä XML-pohjaisista määrittelytiedoista. TUVATAR-tietojärjestelmässä käyttäjä voi tallentaa haetun raportin omalle tietokoneelleen PDF-muodossa tai tulostaa sen suoraan.

6.2.9 XML

XML (Extensible Markup Language) on tekstipohjainen standardi tiedon esittämiseen ja siitä on myös muodostunut tärkeä tiedonsiirron standardi. Alustariippumaton ja laajennettava XML on tärkeä osa Web Services -palveluiden tuottamista. Lukuiset yritykset käyttävät XML:ää jär-

jestelmien välisessä tiedonsiirrossa, koska se on suunniteltu vastaamaan suuren mittaluokan vaatimuksia. Java sisältää monia kirjastoja XML-dokumenttien käsittelyyn (Kosonen ym. 2008, 572). Kuvassa 2 havainnollistan TUVATAR-tietojärjestelmästä tulostettavan potilastulosten XML-määrittelytiedostoa.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<report name="LivingTypeReport" pageformat="A4" orientation="portrait" topmargin="28" bottommargin="28" leftmargin="28" rightmargin="28" >
  <reportheader height="48" fontname="sansserif" fontstyle="bold" fontsize="20">
    <label x="0" y="0" width="100%" height="24" alignment="left" >TUVATAR-lomake</label>
  </reportheader>
  <pageheader height="21" fontname="serif" fontstyle="plain" fontsize="10" onfirstpage="true">
    <date-field x="0" y="0" width="100%" height="14" alignment="right" format="dd.MM.yyyy" fieldname="report.date"/>
  </pageheader>
  <pagefooter height="18" fontname="sansserif" fontstyle="plain" fontsize="10" >
    <string-field name="pgn" fieldname="PageNoM" x="0" y="0" width="100%" height="12" alignment="right" />
  </pagefooter>
  <groups >
    <group name="GRP_LABELS" >
      <groupheader height="3" fontname="Helvetica" fontsize="10" pagebreak="false">
        <line name="line2" x1="0" y1="3" x2="0" y2="3" weight="0.3"/>
      </groupheader>
      <groupfooter>
        <line name="line2" x1="0" y1="0" x2="0" y2="0" weight="0.7"/>
      </groupfooter>
      <fields>
        <field>GROUP</field>
      </fields>
    </group>
  </groups>
  <items pagebreak="true">
    <band name="GareSummaryHeader" layout-cachable="raise" dynamic="true" height="15" fontname="sansserif" fontstyle="plain" fontsize="12" >
      <rectangle name="background" x="0" y="0" width="100%" height="100%" color="#F5F5F5" weight="0"/>
      <rectangle name="r0" x="0" y="0" width="0.5" height="100%" color="#000000" weight="0.5"/>
      <rectangle name="r1" x="538.5" y="0" width="0.5" height="100%" color="#000000" weight="0.5"/>
      <string-field name="hdr" x="5" y="2" width="305" height="15" dynamic="true" fieldname="HEADER" />
      <string-field name="txt" x="313.5" y="2" width="220" height="15" dynamic="true" nullstring="" fieldname="TEXT" />
    </band>
  </items>
  <functions>
    <property-ref name="report.date" />
    <function name="backgroundTrigger" class="org.jfree.report.function.ElementVisibilitySwitchFunction">
      <properties>
        <property name="element">background</property>
      </properties>
    </function>
    <function name="MyPageNumber" class="org.jfree.report.function.PageFunction"/>
    <function name="MyPageCount" class="org.jfree.report.function.PageTotalFunction" />
    <expression name="PageNoM" class="org.jfree.report.function.TextFormatExpression">
      <properties>
        <property name="pattern">Sivu {0,number,integer} / {1,number,integer}</property>
        <property name="field[0]">MyPageNumber</property>
        <property name="field[1]">MyPageCount</property>
      </properties>
    </expression>
  </functions>
</report>
```

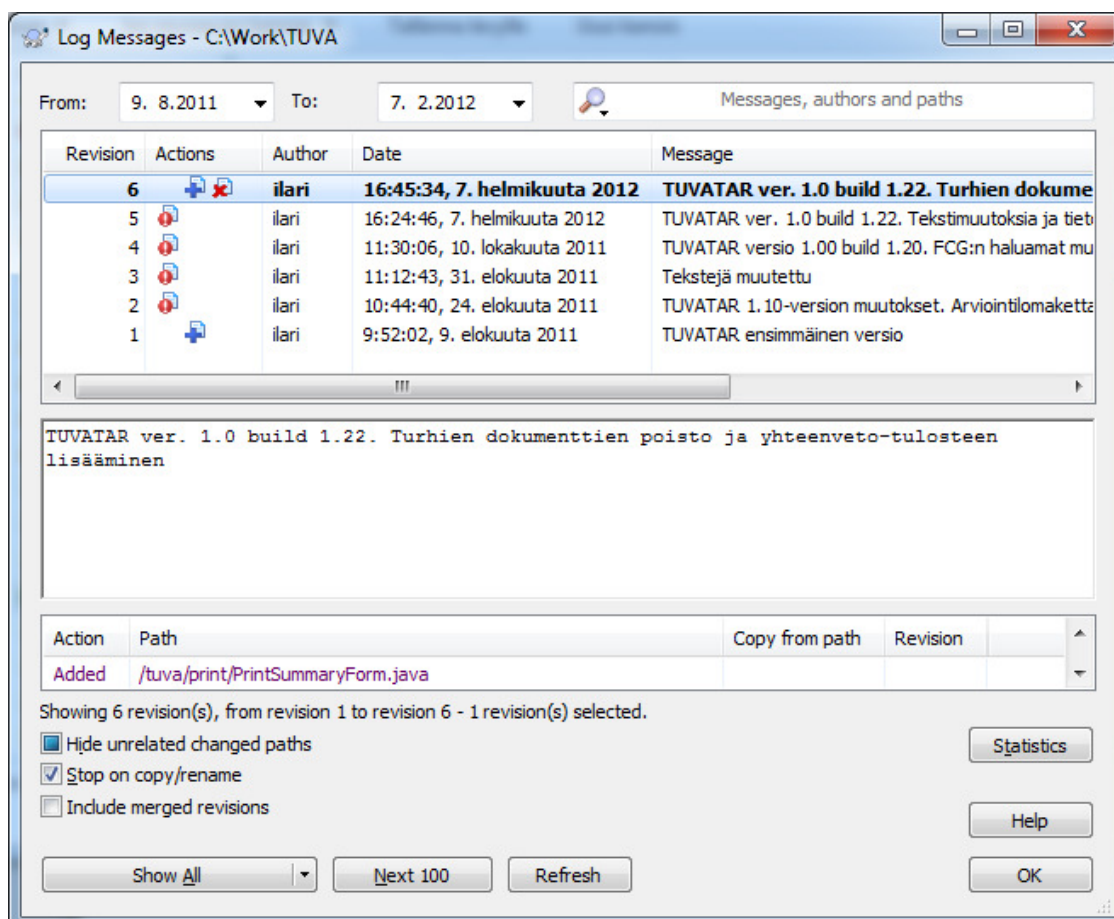
Kuva 2: Tulostettavan TUVATAR-lomakkeen XML-määrittelytiedosto.

6.3 Versionhallinta

Ohjelmistoprojekteissa on tärkeää välillä katsoa menneisyyteen. Versionhallinta on tärkeä työkalu ohjelmoijille, jotka tekevät jatkuvasti pieniä muutoksia ohjelmien lähdekoodiin ja tiedostoihin. Ohjelmoijat peruvat tai tarkastavat tekemiänsä muutoksia usein seuraavana työpäivänä. Versionhallinta on myös tarpeellinen, kun kehitysprojektin jäsenet työskentelevät yhtäaikaaisesti samojen tiedostojen parissa tehden niihin muutoksia. Lyhyesti kuvailtuna versionhallinta on informaation muutosten hallintaa.

TortoiseSVN on avoimen lähdekoodin versionhallintaohjelma, joka toimii Windows-käyttöliittymänä Apache Subversion -versionhallintaan. Ohjelman avulla tiedostot ja hakemis-

tot tallennetaan keskitettyyn arkistoon ja niistä kirjataan muistiin tehdyt muutokset. Arkisto toimii kuin tiedostopalvelin, mutta sen avulla voidaan palauttaa kehittämissprojektin vanhoja versioita ja tutkia minkälaisia muutoksia on tehty. Ohjelman avulla voidaan tarkastella, milloin tiedostot ovat muuttuneet ja kuka niitä on muuttanut. (Küng, Large & Onken, 2011.) Kuva 3 on kuvankaappaus lokitietojen tarkastusikkunasta TortoiseSVN-versionhallintaohjelmassa.

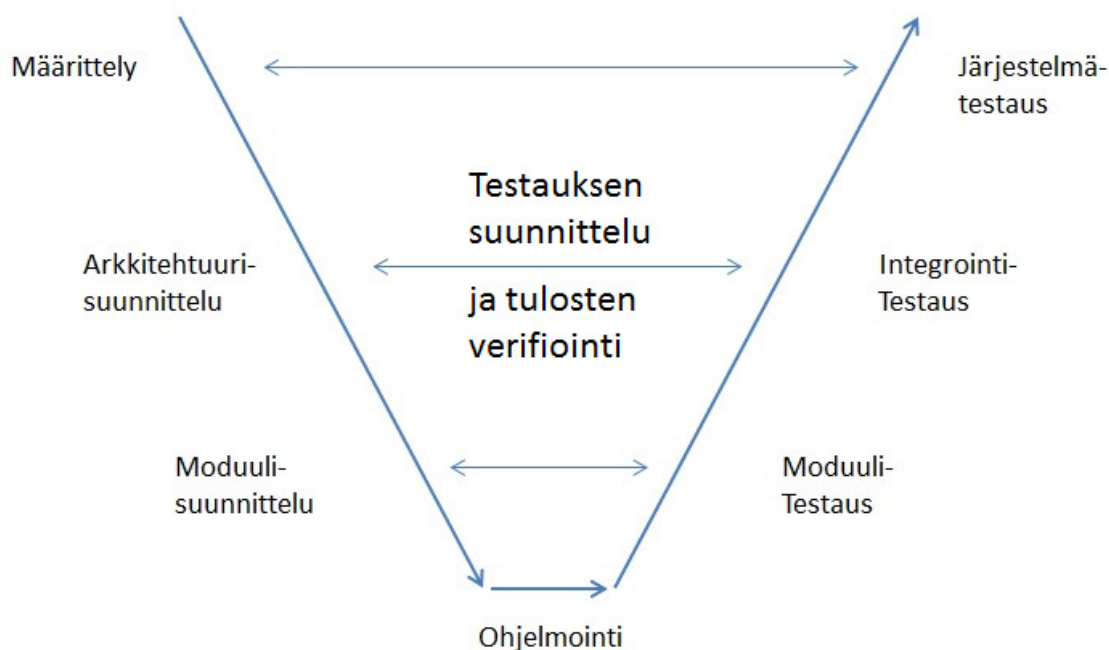


Kuva 3: TortoiseSVN-lokitietojen tarkastelu kehittämissprojektista.

7 Tietojärjestelmän testaus

Tietojärjestelmän toteutuksen jälkeen suoritetaan testaus. Testauksen tarkoituksena on parantaa tietojärjestelmän laatua ja varmistaa sen toimintojen olevan toiminnallisen määrittelyn mukaisia. Testauksella tarkoitetaan lähes kaikkea kokeilemistä. Täydellisen kattava testaus pitäisi tuoda esiin kaikki tietojärjestelmän virheet ja puutteet, mutta käytännössä tämä on kuitenkin mahdotonta. Tietojärjestelmän toimintaan vaikuttavien tekijöiden määrä on usein keskikokoisessa järjestelmässä liian suuri, joten kaikkia virheitä on kustannussyistä mahdotonta etsiä. Hyvän testauksen tavoitteena tulisi voida todeta virheiden olemassaolo, ei niiden täydellistä puuttumista (Pohjonen 2002, 33).

Tietojärjestelmälle asetettavaa testauksen määrää on usein vaikeaa arvioida. Usein kehittämisprojekteissa järjestelmätestausta voidaan jatkaa ”kunnes rahat ja/tai aika loppuvat”. Testausvaiheen lopettaminen ja käyttöönottoon siirtyminen tarkoittaa usein, että on tehty kompromissi tietojärjestelmässä olevien vikojen aiheuttamien kustannuksen ja markkinoilta myöhästymisen aiheuttaman tuoton menetyksen välillä (Haikala & Märijärvi 1995, 232). Kuviossa 4 havainnollistetaan testauksen suoritusta monella tasolla V-mallin mukaisesti.



Kuvio 4: V-mallitestausta (Haikala & Märijärvi, 1995).

V-mallin mukaisesti testattaessa tietojärjestelmän testaus tehdään vertaamalla valmista tietojärjestelmää sen toiminnalliseen määrittelyyn. Testaustasot osoittavat virheiden kriittisyyden; mitä korkeammalla testausasolla virhe havaitaan, sen kalliimmaksi virheen korjaus tulee. On myös mahdollista, että virheen korjaustyö aiheuttaa uusia virheitä. Tällainen tilanne voi esiintyä esimerkiksi, jos järjestelmätestauksessa havaittu virhe aiheuttaa muutoksia useisiin komponentteihin. Suositeltavaa on, että kaikki moduulit ja lopuksi vielä koko järjestelmä testataan uudestaan (Haikala & Mikkonen 2011, 207).

Kuten muihinkin testausmenetelmiin, myös käyttöliittymän testaukseen on valmistauduttava. Testauksen luonteesta riippuen on hyvä hankkia koehenkilöt testausta varten. Koehenkilöille on selvitettävä, mitä heidän toimenkuvaansa kuuluu ja kuinka laajan testauksen he tekevät. Usein on parempi, että testaajana toimii projektin ulkopuolinen henkilö. Ulkopuolinen testaaja etsii tietojärjestelmästä heikkouksia tunteettomammin kuin kehittämisprojektissa työskentelevä henkilö. Näin tietojärjestelmän heikkoudet ja virheet tulevat paremmin esille ja niitä

päästään heti ratkomaan. Testiympäristön lähtötilanteen on oltava jokaiselle testaajalle samanlainen, minkä ansiosta virheen sattuessa on sen aiheuttaja helpompi löytää (Wiio 2004, 223-228).

Komponenttitestauksessa testattavana kohteena on yksittäinen komponentti. Testauksessa tarkastellaan komponentin toimintaa ja verrataan sitä arkkitehtuurisuunnittelusta syntyneeseen tekniseen määrittelyyn. Testaajana toimii usein kyseisen komponentin toteuttaja (Haikala & Märijärvi 1995, 228).

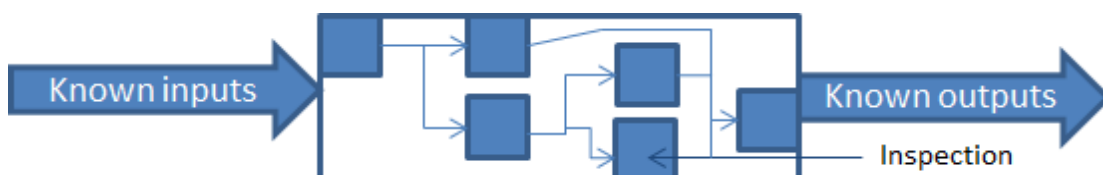
Testauksessa on erilaisia työvaiheita. Näitä ovat testauksen suunnittelu, testiympäristön luonti, testin suorittaminen ja tulosten tarkastelu. Kuten toteutuksen suunnittelu, myös testauksen suunnittelu on tärkeä vaihe onnistunutta testausta. Testauksen suunnittelussa määritellään testitapaukset ja menettelytavat erilaisissa tilanteissa. Kehittämiprojektissa loin jokaiselle oman testiympäristön alkuvaiheen testausta varten. Kun testaus oli edennyt pitkälle, virallinen testiympäristö jaettiin kaikille kehitysprojektin jäsenille. Tämän testiympäristön tarkoituksena on olla viimeinen testi tuotantoon menevälle ohjelmistokorjaukselle tai muulle muutokselle.

7.1 Testitapaukset ja menetelmät

Testitapauksen valinnassa on kaksi eri lähestymistapaa: lasilaatikkotestaus ja mustalaatikkotestaus. Lasilaatikkotestausta käytetään usein V-mallin alimmilla tasoilla, kun taas mustalaatikkotestausta käytetään sen ylimmillä tasoilla. Toteuttajat tietävät tietojärjestelmän toiminnallisuuden sisällön ja näin ollen tietävät, miten tieto liikkuu järjestelmän sisällä. Lisäksi on vielä harmaalaatikkotestaus, joka on näiden kahden testausstrategioiden yhdistelmä (Haikala & Mikkonen 2011, 209).

7.1.1 Lasilaatikkotestaus

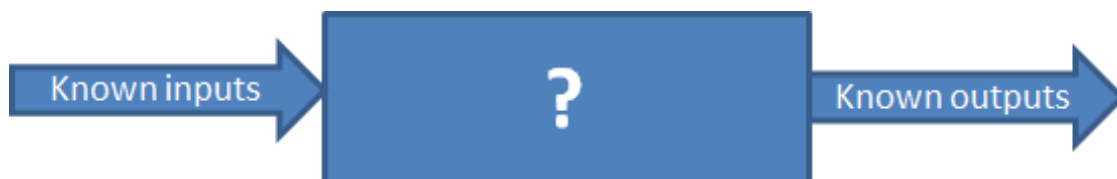
Lasilaatikkotestauksessa testitapausten valinnassa käytetään järjestelmän teknisiä tietoja. Lasilaatikkotestauksessa testaajalla on tarvittavat tiedot testata kaikkia mahdollisia asioita järjestelmästä. Kuviossa 5 havainnollistan testaajan taustatietoa järjestelmän rakenteesta testausvaiheessa.



Kuvio 5: Lasilaatikkotestaus (Clifton 2003).

7.1.2 Mustalaatikkotestaus

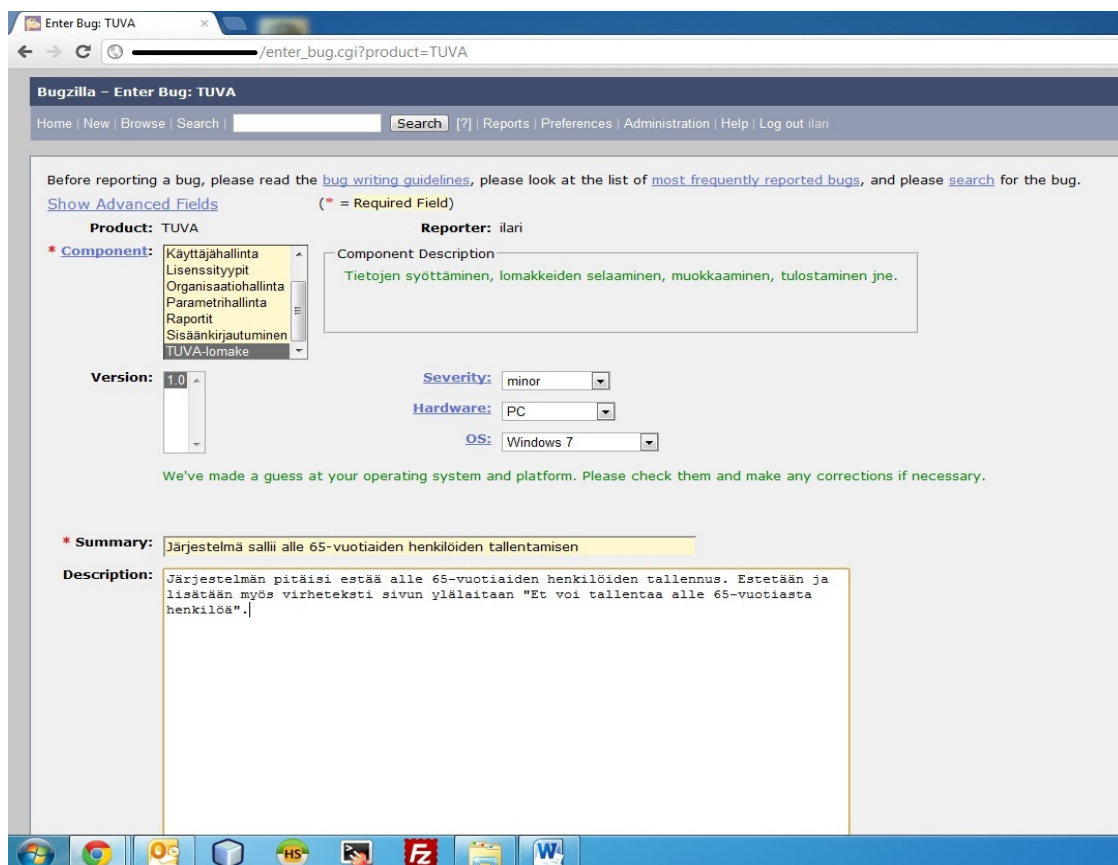
Mustalaatikkotestauksessa testaaja ei tiedä tai välitä järjestelmän rakenteesta, vaan hän tutkii syötteitä ja järjestelmän palauttamien arvojen oikeellisuutta. Testauksessa ei edes tiedetä, miten järjestelmä käsittelee virheitä, jos niitä ilmaantuu. Mustalaatikkotestauksessa testauksen kohde on tuntematon ja tästä nimitys ”mustalaatikko” (Clifton 2003). Kuvio 6 osoittaa testaajan tietämättömyyttä järjestelmän rakenteesta.



Kuvio 6: Mustalaatikkotestaus (Clifton 2003).

7.2 Bugzilla

Ohjelmistoprojektien virheiden raportoimisessa hyödynnetään usein tähän tarkoitukseen toteutettuja työkaluja. Bugzilla on suosittu web-pohjainen virheiden raportointityökalu. Bugzillan käyttäminen vaatii web- ja tietokantapalvelimen. Projektille luodaan oma alue Bugzillan, jonka kautta testaajat ja käyttäjät voivat ilmoittaa esille tulleita ohjelmistovirheitä eli buggeja. Raportoinnin jälkeen bugit annetaan jollekin kehittäjälle korjattavaksi. Ohjelmistovirheitä voidaan lisätä huomautuksia jälkeensä. Kun kehittäjä on ratkaissut ongelman, muutetaan kyseisen virheen tilaksi ”selvitetty”. Kuvassa 4 havainnollistan ohjelmistovirheen tallentamista Bugzillan.



Kuva 4: Kuvakaappaus virheensyötöstä Bugzillaan.

8 TUVATAR-tietojärjestelmä

TUVATAR-tietojärjestelmän mielenterveyskuntoutujan tietojen keruun tarkoituksena on antaa hoitavalle yksiköille mahdollisuus tutkia omaa toimintaansa ja verrata tuloksia oman organisaation muihin tietoihin. Jokaisella yksiköllä on yksinoikeus omiin tietoihinsa, eikä laitoksen tietoja ja tuloksia luovuteta muille osapuolille ilman kyseisen laitoksen suostumusta. TUVATAR-tietojärjestelmän tarkoituksena on kerätä ja raportoida tietoa.

TUVATAR rekisteri syntyy, kun asiakkaasta tehdään ja tallennetaan mielenterveyskuntoutukseen liittyviä arviointitietoja valtakunnalliseen tietokantaan. TUVATAR käyttöoikeus myönnetään aina rekisterinpitäjittäin. Rekisterinpitäjä voi olla laitos, yhteisö tai säätiö, jonka käyttöä varten henkilörekisteri perustetaan ja jolla on oikeus määrätä henkilörekisterin käytöstä.

TUVATAR-tietojärjestelmä sisältää mielenterveyskuntoutujan perustietoja sekä arviointien hallinnan. Hoitoisuus tarkoittaa hoitajan omaa tuntemusta siitä, kuinka raskaana tai helppona hän kokee asiakkaan hoidon riippumatta tämän toimintakyvystä. TUVATAR - mielenterveysmittari luokittelee oireet, jotka häiritsevät päivittäisistä toiminnoista suoriutumisesta sekä niissä tarvittavien taitojen arvioinnin. Mittari mittaa päivittäisiä toimintoja ja niis-

sä tarvittavia taitoja. Käyttäjä saa tallennetuista tiedoista asiakaskohtaisten tulosteiden ja raporttien lisäksi laajemmat vertailuraportit. Kuvassa 5 esitän esimerkin TUVAR-arvioinnin tekemisestä.

The screenshot shows the TUVATAR web application interface. The browser address bar shows the URL `/DoInputFormAction.do`. The page title is "TUVATAR" and the logo "FGG Finnish Consulting Group" is visible. The navigation menu on the left includes items like "Etusivu", "Arvioinnin tekeminen", "Lomakkeiden haku", "Henkilön arvioinnit", "Dokumentit", "Tietojen siirto", "Kaikki arvioinnit", "Kaikki tiedot", "Salasanan vaihto", "Parametristot", "Käyttäjät", "Lisenssityypit", "Organisaatiotiedot", "Etusivun hallinta", and "Kirjautu ulos".

The main content area is titled "TUVAR-arvioinnin tekeminen" and includes a warning: "Muista tallentaa muutoksesi ennen tämän sivun sulkemista!". There are two search fields: "Asiakkaan henkilötunnus" (160988) and "Asiakkaan nimi" (Testaja Enni). A "Hae" button is next to the first field. Below the search fields, it says "Asiakkaan tiedot löytyivät!". There is a "Tee asiakkaalle uusi arviointi" button.

The "Rekisterinpitäjä" is "Prime Solutions Oy". The "Kotikunta" is "Helsinki". There is a checkbox for "Salli kunnan vaihtaminen".

The table below shows the assessment criteria:

| | Yksityinen | Kunnallinen |
|---|----------------------------------|-----------------------|
| Kotihoito (Koti) | <input type="radio"/> | <input type="radio"/> |
| Perusterveydenhuollon avovastaanotto (PthAvo) | <input checked="" type="radio"/> | <input type="radio"/> |
| Erikoissairaanhoidon avovastaanotto (EshAvo) | <input type="radio"/> | <input type="radio"/> |
| Tukiasuminen (TukiAs) | <input type="radio"/> | <input type="radio"/> |
| Päivätoiminta (Päivät) | <input type="radio"/> | <input type="radio"/> |
| Työtoiminta (TyöToi) | <input type="radio"/> | <input type="radio"/> |
| Palveluasuminen - ei ympärivuorokautinen (PaTa) | <input type="radio"/> | <input type="radio"/> |
| Tehostettu palveluasuminen - ympärivuorokautinen (PaTe) | <input type="radio"/> | <input type="radio"/> |
| Sosiaalitoimen laitoshoido (SosLai) | <input type="radio"/> | <input type="radio"/> |
| Perusterveydenhuollon vuodeosasto (PthVos) | <input type="radio"/> | <input type="radio"/> |

Kuva 5: TUVAR-arvioinnin tekeminen.

TUVATAR-tietojärjestelmä sisältää erilaisia osioita. Käyttäjät liikkuvat näiden osioiden sisällä ja suorittavat toimintoja. Osa osioista on ainoastaan tietojen tallennusta varten ja osa tietojen lukemista. Osioiden välillä liikutaan tietojärjestelmän osiovalikon avulla. Osiot ja toiminnot päätettiin kehittämissuunnitelman määrittelyvaiheessa. Taulukossa 1 esittelen näitä osioita ja niiden ominaisuuksia.

| Osio | Kuvaus |
|---------------------------|---|
| Etusivu | Etusivun tehtävä on toimia käyttäjille ilmoitustauluna. Etusivulla ilmoitetaan tietojärjestelmän päivityksistä ja muista ajankohtaisista asioista. |
| Arvioinnin tekeminen | Tietojärjestelmään tallennetaan asiakaskohtaiset tiedot tämän osion kautta. Käyttäjä tallentaa asiakkaan hoitoisuuden arviointiin tarvittavat tiedot tietokantaan. |
| Lomakkeiden haku | Käyttäjä voi käyttöoikeuksien mukaan hakea rekisteripitäjän tai hänen itsensä tallentamat valmiit ja keskeneräiset lomakkeet. Lomakkeita voidaan muokata, poistaa tai tulostaa. |
| Henkilön arvioinnit | Yksittäisen henkilön kaikki arvioinnit haetaan tämän osion kautta. Haun avulla saadaan myös lomakkeet, jotka on poistettu tai korvattu. Lomakkeiden tiedot voidaan myös tulostaa. |
| Dokumentit | Dokumentit-osio sisältää käyttäjille tärkeitä dokumentteja. Dokumentit on tallennettu PDF-muotoon, joten käyttäjä voi tarvittaessa tulostaa haluamansa tiedot. |
| Tietojen siirto | Käyttäjän on mahdollista ladata csv-tiedosto, joka pitää sisällään rekisteripitäjän kaikki lomakkeet ja niihin tallennetut henkilötiedot. |
| Kaikki arvioinnit | Pitää sisällään käyttäjän rekisteripitäjään tallennettuja arviointitietoja. Tämä on kuitenkin suppeampi raportti tietojen siirto-osioista saataviin tietoihin verrattuna. |
| Kaikki tiedot | Mahdollistaa käyttäjälle kaikkiin rekisteripitäjiin tallennettujen tietojen lataamisen csv-muodossa. Tämä osio on tarkoitettu vain ylläpitäjää varten. |
| Salasanan vaihto | Käyttäjät voivat vaihtaa uuden salasanan tunnukselleen. |
| Parametrien hallinta | Lisätään tai poistetaan organisaation parametreja. Parametreille voidaan myös liittää ylätasoja, joihin ne ovat riippuvaisia. |
| Käyttäjähallinta | Käyttäjähallinnassa voidaan lisätä, poistaa tai muokata käyttäjien tietoja. Myös käyttöoikeuksien asettaminen käyttäjälle voidaan hoitaa tämän osion kautta. |
| Lisenssityyppien hallinta | Mahdollistaa lisenssityyppien lisäyksen ja poiston. Lisenssityypeille voidaan määritellä erilaisia käyttöoikeusyhdistelmiä. |
| Organisaatiohallinta | Organisaatiohallinnassa lisätään, muokataan tai poistetaan organisaatioita. Organisaatioille asetetaan myös lisenssityyppi, joka määrää organisaatiolle asetetut käyttöoikeudet. |

Taulukko 1: TUVATAR-tietojärjestelmän osiot.

Määrittelyvaiheessa tarkoituksena oli ottaa mukaan myös raportointiosuus, mutta se pudotettiin pois asiakkaan pyynnöstä suunnitteluvaiheessa. Raportointiosuus toteutetaan mahdollisesti tietojärjestelmän seuraavaan versioon. Tietojärjestelmän ensimmäisestä versiosta jätettiin

pois myös tulosteiden grafiikat. Tulevaisuudessa tulosteille tullaan toteuttamaan kaaviot, jotka havainnollistavat kerättyjen tietojen muutoksia.

9 Tietojärjestelmän toimintaympäristö

Tietojärjestelmän eri ympäristöt sijaitsevat Nebulan palvelintiloissa. Palvelin on luotettava ja vakaa sekä perustuu yleisiin ja käytössä toimiviksi havaittuihin ratkaisuihin. Palvelimelle on turvattu katkoton sähkönsyöttö sekä kohdennettu verkkoyhteys. TUVATAR-tietojärjestelmä on käytettävissä periaatteessa ympäri vuorokauden, kuitenkin niin, että mahdolliset virhe- ja vikatilanteet käsitellään virka-aikana. Tietojärjestelmään kohdistuneista käyttökatkoksista ilmoitetaan käyttäjille hyvissä ajoin tietojärjestelmässä olevien tiedotustoimintojen avulla.

Tietojärjestelmä kerää lokitietoja siitä, kuka järjestelmää on käyttänyt, miten ja milloin järjestelmää on käytetty. Lokitietojen avulla voidaan selvittää tietojärjestelmässä mahdollisesti tapahtuneita väärinkäyttöjä. Lokitiedot toimivat hyvin ongelmatilanteiden ratkaisussa.

10 Kehittämiprojektin ongelmat ja ratkaisut

Jokaisessa projektissa ilmenee ongelmia. Ongelmat voivat johtua esimerkiksi projektissa työskentelevien henkilöiden erilaisista toimintatavoista ja tavoitteista tai ohjelmistoprojektin teknisistä syistä. Ongelmia pitäisi lähteä ratkomaan heti huomaamisen jälkeen. Jos ongelmien ratkomista lykätään myöhemmäksi, ne suurenevat ja niiden ratkaisu vaikeutuu.

Tässä projektissa ilmeni erilaisia ongelmia. Ongelmia ilmeni suunnitteluvaiheessa, kun osa toiminnoista oli määritelty liian suppeasti. Ongelma ilmeni vasta suunnitteluvaiheessa, mutta ongelma ratkaistiin nopeasti. Ongelma ratkaistiin asiakkaan kanssa sähköpostikeskustelussa, jossa määriteltiin joitain toimintoja tarkemmin. Ongelmasta ei kuitenkaan syntynyt suurempia vahinkoja projektille.

Arviointilomake määriteltiin hieman suppeasti. Nämä ongelmat ilmenivät kuitenkin vasta testausvaiheessa, jolloin jouduttiin palaamaan takaisin määrittelyvaiheeseen. Ongelmana oli käytettävyyteen liittyviä seikkoja. Tietojärjestelmässä olevan arviointilomakkeen koko oli niin iso, joten sen käyttämisen raskaus tuli vasta testausvaiheessa esille. Arviointilomakkeen ulkoasu määriteltiin uudestaan ja tehtiin tarvittavat muutokset. Tämän ansiosta tietojärjestelmän käytettävyys parani huomattavasti.

11 Kehittämisehdotukset

Kehittämisprojektissa keskityttiin mielestäni liian vähän suunnitteluun ja testaukseen. Toteutukselle annettiin suurin osa työpanoksesta. Tarkemman suunnittelun avulla olisimme voineet pienentää toteutusprosessin raskautta ja yksinkertaistaa koko työvaihetta. Suunnitteluvaihe sujui kuitenkin niin hyvin, että sen pohjalta saatiin toteutettua tietojärjestelmä.

Järjestelmän tuotantojulkaisu saatiin asiakkaille hieman myöhässä. Myöhästyminen johtui testi-, koulutus- ja tuotantoympäristöjen käyttöönoton viivästyisestä. Projektisuunnitelmassa oli määritelty milloin palvelimesta lähetetään pyyntö, mutta tämä osui juuri loma-aikoihin, joten palvelin saatiin käyttöön odotettua myöhemmin.

Järjestelmän ominaisuudet olisi hyvä määritellä mahdollisimman tarkasti projektin määrittelyvaiheessa. Järjestelmän kehitysprosessissa palattiin pari kertaa taaksepäin toteutusvaiheessa tulleiden ongelmallisuuksien vuoksi.

12 Tulokset

Opinnäytetyön käytännön työnä toimineesta kehittämisprojektista syntyi TUVATAR-tietojärjestelmä. Uskon järjestelmän palvelevan hyvin sen käyttötarkoitustaan. Tietojärjestelmä nopeuttaa hoitajien työtä ja helpottaa kerätyn tiedon analysoimista. Tietojärjestelmän kehittämisen ohessa loimme myös tietojärjestelmän rinnalle ohjeen toiminnoista. Ohjeen tarkoituksena on opastaa käyttäjiä eri toimintojen suorittamisessa.

Tietojärjestelmän aikataulutuksessa oli hieman ongelmia, koska asiakas sai valmiin tietojärjestelmän käyttöön hieman myöhässä. Uskon asiakkaan olevan kuitenkin tyytyväinen tietojärjestelmän lopputulokseen. Oppimiskokemuksena kehittämisprojekti toimi todella hyvin. Eri laisten kehitysvaiheiden, toimintatapojen ja menetelmien tutkiminen toi minulle uusia näkökulmia tietojärjestelmän kehittämiseen. Tarkan suunnittelun ansiosta tietojärjestelmää on helppoa lähteä kehittämään tulevaisuudessa. Tietojärjestelmä suunniteltiin kehitettäväksi asteittain, joten tulemme tekemään jatkossa uusia toiminnallisuuksia ja raportteja tietojärjestelmään.

Järjestelmä on ollut asiakkaan käytössä yli viisi kuukautta ja se on toiminut hyvin. Mitään ongelmia tietokannassa ei ole ilmennyt sen käyttöönoton jälkeen. Ohjelmasta ei ole myöskään löytynyt suurempia virheitä. Asiakkaan pyynnöstä on kuitenkin tehty pieniä muutoksia tietojärjestelmän teksteihin.

13 Johtopäätökset

Kehittämisprojekti onnistui ja tietojärjestelmä saatiin valmiiksi. Työvaiheissa pystyin hyödyntämään paljon jo ennalta opittuja taitoja ja tietoja, mutta opin paljon uutta, esimerkiksi tietojärjestelmän suunnittelumenetelmistä. Tietojärjestelmän kehittämisessä huomasin jokaisen kehittämisvaiheen tärkeyden kehittämisprojektin valmiiksi saattamisessa. Jokainen huonosti suoritettu kehitysvaihe vaikuttaa tuleviin kehitysvaiheisiin todella paljon, esimerkiksi huonon suunnittelun pohjalta on todella vaikeaa saada suoritettua toteutus onnistuneesti. Kuitenkin nykyisten suunnittelumallien ja menetelmien ansiosta projektin kehitys saadaan pidettyä hyvin kasassa.

Uutena asiana huomasin tietojärjestelmän testaamisen raskauden. Testausvaiheeseen tulisi kuluttaa paljon aikaa, jotta jokainen järjestelmän osio voidaan testata huolellisesti. Testaamiseen on luotu helpottavia menetelmiä ja työkaluja, esimerkiksi yksikkötestaus- ja raportointityökalut.

Kehittämisprojektin tietojärjestelmä on kehitetty toisen järjestelmän pohjalta, joten useita komponentteja uudelleenkäytettiin. Komponenttien uudelleenkäyttö on hyvin yleistä ja myös suositeltavaa. Monet organisaatiot käyttävät komponentteja uudelleen kehittämisprojekteissaan. Kehittämisprojektin aikana huomasin uudelleenkäytön vaikutukset hyvin. Uudelleenkäyttö nopeuttaa huomattavasti tietojärjestelmän kehittämisen vaiheita. Esimerkiksi testaus sujui nopeammin, koska osa komponenteista oli jo huolellisesti testattu aiemmassa kehittämisprojektissa.

Projektille asetettujen aikataulujen ja laadullisten vaatimuksien avulla kehittämisprojektin kulku on suoraviivaista. Ennen kuin siirrytään suunnitteluvaiheeseen, olisi hyvä suunnitella jokaiselle vaiheelle oma aikataulu, jonka mukaan vaiheiden tehtävät suoritetaan. Tietojärjestelmien kehittäminen on kehittynyt paljon, tämän johdosta yhä harvempi kehittämisprojekti epäonnistuu.

Lähteet

Doray A. 2006. Beginning Apache Struts: From Novice To Professional. USA: Apress.

Glaser, P. Lee, W. & Wager, A. 2009. Health Care Information Systems. San Francisco: Jossey-Bass.

Haikala, I. & Mikkonen T. 2011. Ohjelmistotuotannon käytännöt. Hämeenlinna: Kariston Kirjapaino Oy.

Haikala, I. & Märijärvi J. 1995. Ohjelmistotuotanto. Espoo: Suomen ATK-kustannus Oy.

Kosonen, P. Peltomäki, J & Silander, S. 2008, Java 2 - ohjelmoinnin peruskirja. Jyväskylä: Docendo Finland Oy.

Kuha, J. 2008. Tehokas Java EE -sovellustuotanto. Jyväskylä: WSOYpro/Docendo-tuotteet.

Lehtimäki, T. 2006. Ohjelmistoprojektit käytännössä. Jyväskylä: Gummerus Kirjapaino Oy.

Meloni J. 2003. MySQL Trainer Kit. Helsinki: Edita Prima Oy.

Pohjonen, R. 2002. Tietojärjestelmien kehittäminen. Jyväskylä: Docendo Finland Oy.

Wiiio, A. 2004. Käyttäjystävällisen sovelluksen suunnittelu. Helsinki: Edita Prima Oy.

Internet lähteet

Clifton, M. 2003. Advanced Unit Testing, Part I - Overview. Viitattu 23.3.2012.
<http://www.codeproject.com/Articles/5019/Advanced-Unit-Testing-Part-I-Overview#Black%20Box%20vs.%20White%20Box%20Test>

Küng, S, Large, S & Onken L. 2011. TortoiseSVN. Suomentaja Grano, K. Viitattu 23.3.2012.
http://tortoisesvn.net/docs/release/TortoiseSVN_fi/index.html

Kuvat

| | |
|---|----|
| Kuva 1: Kuvakaappaus Netbeans 6.5.1 kehitysympäristöstä. | 16 |
| Kuva 2: Tulostettavan TUVATAR-lomakkeen XML-määrittelytiedosto. | 17 |
| Kuva 3: TortoiseSVN-lokitietojen tarkastelu kehittämisprojektista. | 18 |
| Kuva 4: Kuvakaappaus virheensyötöstä Bugzillaan..... | 22 |
| Kuva 5: TUVA-arvioinnin tekeminen. | 23 |
| Kuva 6: Lomakkeiden haku. | 33 |
| Kuva 7: Henkilön arvioinnit. | 33 |
| Kuva 8: Dokumentit. | 34 |
| Kuva 9: Tietojen siirto. | 34 |
| Kuva 10: Kaikki tiedot..... | 35 |
| Kuva 11: Salasanan vaihto..... | 35 |
| Kuva 12: Parametristojen ylläpito. | 36 |
| Kuva 13: Käyttäjähallinta. | 36 |
| Kuva 14: Lisenssityyppien ylläpito. | 37 |
| Kuva 15: Organisaatietietojen ylläpito..... | 37 |
| Kuva 16: Etusivun ylläpito. | 38 |

Kuviot

| | |
|---|----|
| Kuvio 1: Tietojärjestelmän kehittämisen vaiheet (Haikala & Mikkonen, 2011)..... | 8 |
| Kuvio 2: Suunnitteluprosessi (Haikala & Märijärvi, 1995)..... | 10 |
| Kuvio 3: Uudelleenkäyttö tuotantoprosessissa (Haikala & Mikkonen, 2011). | 14 |
| Kuvio 4: V-mallitestausta (Haikala & Märijärvi, 1995). | 19 |
| Kuvio 5: Lasilaatikkotestausta (Clifton 2003). | 20 |
| Kuvio 6: Mustalaatikkotestausta (Clifton 2003). | 21 |

Taulukot

| | |
|---|----|
| Taulukko 1: TUVATAR-tietojärjestelmän osiot. | 24 |
|---|----|

Liitteet

| | |
|---|----|
| Liite 1: Kuvakaappaukset TUVATAR-tietojärjestelmän osioista. | 33 |
|---|----|

Liite 1: Kuvakaappaukset TUVATAR-tietojärjestelmän osioista.

TUVATAR Ohje

TUVA-lomakkeiden haku

Lomakkeita voit hakea henkilötunnuksen, asumismuodon tai aikavälin perusteella. Kaikkia hakeutoja ei tarvitse käyttää. Voit jättää halutessasi kaikki hakeudet tyhjiksi, jolloin näet viimeiset itse tekemäsi arvioinnit painamalla hae nappia. Saman asiakkaan muiden työntekijöiden tekemät arvioinnit saavat näkyviin "Henkilön arvioinnit" kautta.

Asiakkaan henkilötunnus:

Taso 1:

Aikaväli -

Vain valmiit TUVA-lomakkeet
 Vain keskeneräiset TUVA-lomakkeet
 Ota hakuun mukaan kaikki rekisterinpitäjän lomakkeet

Tulokset:

| Nimi | Henkilötunnus | Taso 1 | Tekohetki | Valmis? | TUVA |
|-------|---------------|------------|-----------|------------|-------|
| Testi | Testaaja Enni | 160988-███ | PthAvo | 02.01.2012 | Kyllä |

FCG Finnish Consulting Group Oy
 PL 950, 00601 Helsinki, käytösosoite: Osmontie 34, 00601 HELSINKI
 Vaihde 010 4090 Faksi 010 409 2395 etunimi.sukunimi@fcg.fi
 Ohjelman toimittaja Prime Solutions Oy, www.primesolutions.fi

Kuva 6: Lomakkeiden haku.

TUVATAR Ohje

Raportti yksittäisen asiakkaan toimintakyvyn muutoksista

Henkilötunnus

Aikaväli -

Tulokset:

Voit valita tulosteelle 1-6 arviointia. Listalla näkyy vain valmiit arvioinnit. Poistetut lomakkeet näkyvät punaisella tekstillä, korjatut vihreällä.

| Tekohetki | Asumismuoto | TUVA-indeksi | Hoitoisuus | Tekijä |
|-------------------------------------|-------------|--------------|------------|------------|
| <input checked="" type="checkbox"/> | 02.01.2012 | PthAvo | 4 | Ilari Husu |

FCG Finnish Consulting Group Oy
 PL 950, 00601 Helsinki, käytösosoite: Osmontie 34, 00601 HELSINKI
 Vaihde 010 4090 Faksi 010 409 2395 etunimi.sukunimi@fcg.fi
 Ohjelman toimittaja Prime Solutions Oy, www.primesolutions.fi

Kuva 7: Henkilön arvioinnit.

The screenshot shows a web browser window with the address bar at /TUVA/Documents.do. The page header includes the Finnish Consulting Group (FCG) logo and the TUVATAR title. A navigation menu on the left lists various functions such as 'Etusivu', 'Arvioinnin tekeminen', 'Lomakkeiden haku', 'Henkilön arvioinnit', 'Dokumentit', 'Tietojen siirto', 'Kaikki arvioinnit', 'Kaikki tiedot', 'Salasanan vaihto', 'Parametristot', 'Käyttäjät', 'Lisenssityypit', 'Organisaatiotiedot', 'Etusivun hallinta', and 'Kirjaudu ulos'. The main content area is titled 'Ohje' and 'Dokumentit', displaying a list of documents: 'TUVATAR-ohjelman käyttöehdot' and 'TUVA lomake'. A 'Testi' link is visible at the bottom left of the main area. The footer contains contact information for FCG and Prime Solutions, along with the version number 'TUVATAR ver. 1.0 build 1.22'.

Kuva 8: Dokumentit.

The screenshot shows a web browser window with the address bar at /TUVA/FileTransfer.do. The page header includes the Finnish Consulting Group (FCG) logo and the TUVATAR title. A navigation menu on the left is identical to the previous screenshot. The main content area is titled 'Ohje' and 'Tietojen siirto', with the sub-header 'Luo CSV-tiedosto (erotinmerkki ;)'. Below this, there is a form with the label 'Aikaväli jolta tiedot haetaan:' and two input fields. The first field contains '1.1.2011'. A 'Luo CSV-tiedosto' button is located below the form. A 'Testi' link is visible at the bottom left of the main area. The footer contains contact information for FCG and Prime Solutions, along with the version number 'TUVATAR ver. 1.0 build 1.22'.

Kuva 9: Tietojen siirto.

TUVATAR Finnish Consulting Group

TUVATAR

Ohje

Hae kaikki asiakastiedot TUVATTARESTA

Luo CSV-tiedosto (erotinmerkki ;)

Aikaväli jolta tiedot haetaan:
 -

Asiakkaan kotikunta

Rekisterinpitäjä

Ota mukaan kaikki voimassa olevat rekisterinpitäjät

Testi

FCG Finnish Consulting Group Oy
 PL 950, 00601 Helsinki, käytössoite: Osmontie 34, 00601 HELSINKI
 Vaihde 010 4090 Faksi 010 409 2395 etunimi.sukunimi@fcg.fi
 Ohjelman toimittaja Prime Solutions Oy, www.primesolutions.fi

Rekisteriseloste
 TUVATAR ver. 1.0 build 1.22



Kuva 10: Kaikki tiedot.

TUVATAR Finnish Consulting Group

TUVATAR

Ohje

Salasanan vaihto

Vanha salasana

Uusi salasana

Uusi salasana uudestaan

HUOM: Salasana pitää olla minimissään kuusi merkkiä ja maksimissaan 20 merkkiä.

Testi

FCG Finnish Consulting Group Oy
 PL 950, 00601 Helsinki, käytössoite: Osmontie 34, 00601 HELSINKI
 Vaihde 010 4090 Faksi 010 409 2395 etunimi.sukunimi@fcg.fi
 Ohjelman toimittaja Prime Solutions Oy, www.primesolutions.fi

Rekisteriseloste
 TUVATAR ver. 1.0 build 1.22



Kuva 11: Salasanan vaihto.

Parametristojen ylläpito

Muista tallentaa tiedot toiminnon jälkeen!

Valitse parametrieriymä: Taso 1

| Nimi | Lukitu? |
|---|---------|
| <input type="checkbox"/> Kotihoito | Ei |
| <input type="checkbox"/> Perusterveydenhuollon avovastaanotto | Ei |
| <input type="checkbox"/> Erikoissairaanhoidon avovastaanotto | Ei |
| <input type="checkbox"/> Tukiasuminen | Ei |
| <input type="checkbox"/> Päivätoiminta | Ei |
| <input type="checkbox"/> Työtoiminta | Ei |
| <input type="checkbox"/> Palveluasuminen - ei ympärivuorokautinen | Ei |
| <input type="checkbox"/> Tehostettu palveluasuminen - ympärivuorokautinen | Ei |
| <input type="checkbox"/> Sosiaalitoimen laitoshoido | Ei |
| <input type="checkbox"/> Perusterveydenhuollon vuodeosasto | Ei |
| <input type="checkbox"/> Erikoissairaanhoidon osasto / yksikkö | Ei |

Valitse toiminto. Ennen parametrien muokkausta, tutustu huolella ohjekirjaan.

Lisää uusi parametri

Poista valitut parametrit

Yhdistä valitut parametrit

Lukitse / vapauta lukituksesta valitut parametrit

Valitse parametri / Tallenna yllätyt

Tallenna tiedot

Kuva 12: Parametristojen ylläpito.

Käyttäjähallinta

Valitse organisaatio: Lilliputti

| Pääkäyttäjä? | Nimi | Käyttäjätunnus | Organisaatio | Rekisterinpitäjä |
|--------------|-------------------|----------------|--------------|------------------|
| Kyllä | Lilliputti, Leena | lilliputti | Lilliputti | Lilliputti |

Sukunimi: Lilliputti

Etunimet: Leena

Puhelin:

Sähköpostiosoite:

Käyttäjätunnus: lilliputti

Salasana:

Salasana uudestaan:

Tietojen syöttö

Lomakkeiden haku

Henkilön arvioinnit

Oikeus nähdä organisaation kaikki lomakkeet

Oikeus raportointiin

Luo CSV-tiedosto

Oikeus parametristojen ylläpitoon

Oikeus käyttäjähallintaan

Uusi Poista Tallenna tiedot Peruuta

Kuva 13: Käyttäjähallinta.

Lisenssityyppien ylläpito

| | Tyyppi | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
|--------------------------|-----------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| <input type="checkbox"/> | Kaikki oikeudet | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| <input type="checkbox"/> | Pääkäyttäjä | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Käyttöoikeusnumeroiden selitykset:

- 1 Tietojen syöttö
- 2 Lomakkeiden haku
- 3 Henkilön arvioinnit
- 4 Oikeus nähdä organisaation kaikki lomakkeet
- 5 Oikeus raportointiin
- 6 Luo CSV-tiedosto
- 7 Oikeus parametristojen ylläpitoon
- 8 Oikeus käyttäjähallintaan
- 9 Tarkastele kaikkien yksiköiden käyttäjätietoja
- 10 Oikeus lisenssityyppien ylläpitoon
- 11 Oikeus organisaatioiden ylläpitoon
- 12 Oikeus etusivun ylläpitoon
- 13 Oikeus hakea kaikki TUVATTAREN asiakastiedot

Kuva 14: Lisenssityyppien ylläpito.

Organisaatiotietojen ylläpito

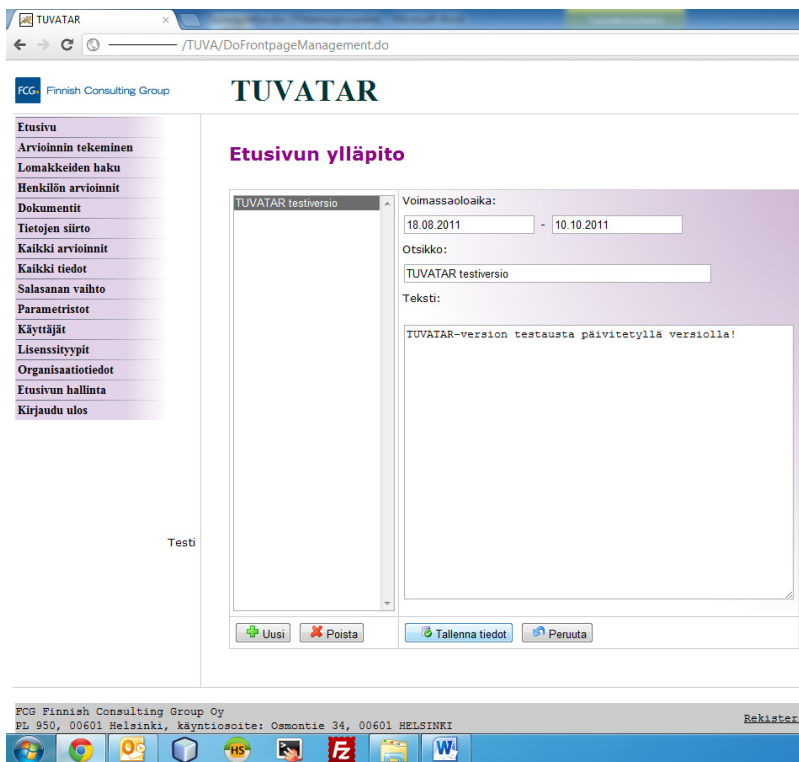
Yksityinen
 Julkinen
 Voimassa
 Ei Voimassa

Kompositio Oy

Yksityinen
 Julkinen

Lisenssityyppi: Pääkäyttäjä
 Nimi: Nokkonen
 Rekisterinpitäjä: Nokkonen
 Kunta: Joensuu
 Läni: Itä-Suomen lääni
 Voimassaolo alkaa: 18.08.2011
 Voimassaolo loppuu:
 Pääkäyttäjä / yhteyshenkilö:
 Vaihda pääkäyttäjä: Nokkonen, Helmi
 Sukunimi: Nokkonen
 Etunimet: Helmi
 Puhelin:
 Sähköpostiosoite:
 Käyttäjätunnus: nokkonen
 Salasana:
 Salasana uudestaan:

Kuva 15: Organisaatiotietojen ylläpito.



Kuva 16: Etusivun ylläpito.